KnowLA: Enhancing Parameter-efficient Finetuning with Knowledgeable Adaptation

Xindi Luo[†] **Zequn Sun**[†,*] **Jing Zhao**[§,*] **Zhe Zhao**[§,*] **Wei Hu**[†,‡,*]
[†] State Key Laboratory for Novel Software Technology, Nanjing University, China
[‡] National Institute of Healthcare Data Science, Nanjing University, China
[§] Tencent AI Lab, China
xdluo.nju@gmail.com, sunzq@nju.edu.cn
{janinezhao,nlpzhezhao}@tencent.com, whu@nju.edu.cn

Abstract

Parameter-efficient finetuning (PEFT) is a key technique for adapting large language models (LLMs) to downstream tasks. In this paper, we study leveraging knowledge graph embeddings to improve the effectiveness of PEFT. We propose a knowledgeable adaptation method called KnowLA. It inserts an adaptation layer into an LLM to integrate the embeddings of entities appearing in the input text. The adaptation layer is trained in combination with LoRA on instruction data. Experiments on six benchmarks with two popular LLMs and three knowledge graphs demonstrate the effectiveness and robustness of KnowLA. We show that KnowLA can help activate the relevant parameterized knowledge in an LLM to answer a question without changing its parameters or input prompts.

1 Introduction

In the era of large language models (LLMs) with billions and possibly trillions of parameters (Du et al., 2022; OpenAI, 2023; Touvron et al., 2023a), parameter-efficient finetuning (PEFT) stands out as a crucial technique enabling the necessary adaptation of LLMs to downstream tasks. It freezes most or even all parameters of LLMs and only finetunes a small number of parameters using limited instruction data. LoRA (Hu et al., 2022) is a widely-used PEFT method that trains small low-rank adapters to approximate the large layers in LLMs. Follow-up work improves the efficiency of LoRA by using quantized weights (Dettmers et al., 2023). In this work, we seek to improve the effectiveness of LoRA while preserving comparable efficiency.

Inspired by knowledge-injected pre-trained language models (PLMs), e.g., ERNIE (Zhang et al., 2019), we explore knowledge graphs (KGs) to enhance the PEFT of LLMs with LoRA. A KG is a large-scale structured knowledge base containing a massive amount of trustworthy knowledge. The typical way of injecting KGs into PLMs in the past few years is incorporating pre-trained entity embeddings at the input layer of a PLM and finetuning the full model on NLP tasks (Lauscher et al., 2019; Peters et al., 2019; Yang et al., 2019; Zhang et al., 2019; Levine et al., 2020; Liu et al., 2021; Lu et al., 2021; Wang et al., 2022). Knowledge injection has improved many PLMs, e.g., BERT (Devlin et al., 2019) and RoBERTa (Zhuang et al., 2021). However, previous knowledge injection methods require fully tuning PLMs, which is inapplicable to LLMs. Furthermore, these methods are founded on the encoder-based architecture of PLMs, and their effectiveness for recent decoder-based LLMs remains unknown. The following questions thereby arise: *Can knowledge injection still enhance the PEFT of LLMs? Also, how can knowledge injection be used to enhance PEFT?*

To answer these questions, in this paper, we propose a knowledgeable adaptation method for PEFT, particularly for LoRA, called KnowLA. It inserts an adaptation layer into a pre-trained LLM. The layer integrates external KG embeddings of entities appearing in the input text of the LLM. Entity embeddings and parameters of the LLM are frozen in PEFT. The proposed adaptation layer is trained combined with LoRA on instruction data. The parameters in our adaptation layer are significantly fewer than those in the LLM and even fewer than those in LoRA. Thus, our KnowLA is also a parameter-efficient method without changing the original parameters of the LLM.

We evaluate KnowLA on six datasets, including commonsense reasoning on CommonsenseQA (Talmor et al., 2019), social interaction reasoning on SIQA (Sap et al., 2019) and BIG-Bench Hard (Suzgun et al., 2023), single-hop reasoning of KBQA on WebQuestionSP (Yih et al., 2016), and close-book QA on TriviaQA (Joshi et al., 2017) and TruthfulQA (Lin et al., 2022). Experimental results show that KnowLA can enhance the effectiveness

* Corresponding authors

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7153–7166

of LoRA at the expense of a limited number of additional parameters. Even when compared to Alpaca2 (Taori et al., 2023), which has a larger LoRA with a similar number of parameters, KnowLA with a smaller LoRA achieves better results.

We assess the robustness of KnowLA with two popular foundation models (i.e., LLaMA 1 (Touvron et al., 2023a) and Llama 2 (Touvron et al., 2023b)), different instruction data (i.e., instruction-following demonstrations in Alpaca2 and Vicuna2 (Chiang et al., 2023)), various KGs (i.e., WordNet (Miller, 1995), ConceptNet (Speer et al., 2017), and Wikidata (Vrandecic and Krötzsch, 2014)), and typical embedding learning models (i.e., RESCAL (Nickel et al., 2011), TransE (Bordes et al., 2013), and RotatE (Sun et al., 2019)), combined with two PEFT methods (i.e., LoRA (Hu et al., 2022) and AdaLoRA (Zhang et al., 2023)). Experiments show that KnowLA can offer stable improvements.

To understand how KnowLA changes the output of an LLM, we analyze the results from two perspectives, which show several interesting findings: (i) KnowLA with LoRA can align the space of the LLM with the space of KG embeddings, and (ii) KnowLA can activate the parameterized potential knowledge that originally exists in the LLM, even though the used KG does not contain such knowledge. According to our findings, in some cases, the LLM outputs incorrect answers not because it does not know the answers, but because its relevant knowledge is not activated by the input prompts. KnowLA can help activate its relevant knowledge without changing its parameters or input prompts.

2 Related Work

2.1 Knowledge Injection

There are three typical knowledge injection methods for PLMs. The first method involves KG embeddings at the input layer of PLMs for joint learning (Zhang et al., 2019; Lu et al., 2021; Wang et al., 2021b). Existing works incorporate entity embeddings for classification tasks, and their knowledge injection modules are independent of PLMs. This poses challenges to aligning the semantic spaces of entity embeddings and PLMs. These knowledge injection methods also necessitate updating the entire model of PLMs. The second method converts relevant triples in KGs into natural language sentences used for pre-training PLMs (Liu et al., 2020; Sun et al., 2020, 2021). The third method introduces adapters into PLMs to enable them to learn KGs

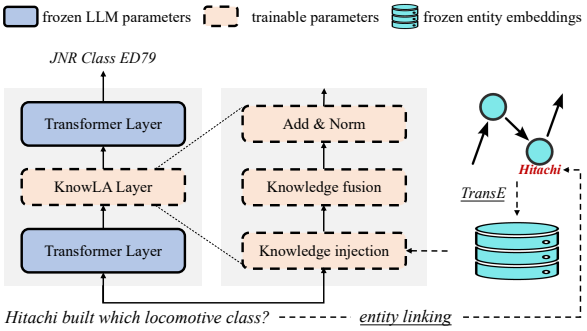

Figure 1: Illustration of knowledgeable adaptation. The KnowLA layer is inserted between two decoder layers of an LLM. It consists of knowledge injection and fusion.

(Wang et al., 2021a). Our KnowLA relates to the first type of methods. It is also a variant of the third method. However, previous methods are built on PLMs while our method is the first attempt to LLMs. KnowLA does not update the parameters of LLMs. It employs a knowledge adapter during PEFT to enhance the LLM's capabilities. The injected entity knowledge can also be deeply integrated with the LLM's knowledge in subsequent decoding steps.

Apart from the above work injecting knowledge inside the model, there are also methods retrieving and augmenting relevant knowledge on the input side of the model (Shwartz et al., 2020; Izacard et al., 2022; Liu et al., 2022; Baek et al., 2023). For example, given an input, Contriever (Izacard et al., 2022) extracts relevant passages from Wikipedia. GKP (Liu et al., 2022) generates relevant prompt text using a sophisticated LLM. KAPING (Baek et al., 2023) retrieves relevant triples in KGs.

2.2 Parameter-efficient Finetuning

PEFT methods aim to optimize LLMs while minimizing the computational resources and data required. Adapter Tuning (Houlsby et al., 2019) is a lightweight alternative that inserts a small neural module called adapter in each layer of a PLM while keeping the majority of the pre-trained parameters frozen. Inspired by the prompt engineering methods, Prefix Tuning (Li and Liang, 2021) sets trainable prefix tokens in the input or hidden layers, and only these soft prompts are trained. LoRA (Hu et al., 2022) is a low-rank adaptive method that allows training dense layers indirectly by optimizing low-rank factorized matrices that capture changes in dense layers during the adaptation process while keeping the pre-trained weights unchanged. QLoRA (Dettmers et al., 2023) im-

proves LoRA by using NF4 quantization and double quantization techniques. Adalora (Zhang et al., 2023) is an improvement on LoRA, addressing the limitation of the fixed incremental matrix rank in LoRA. Adalora introduces a method that dynamically allocates ranks for downstream tasks, yielding promising results. Our KnowLA follows the mainstream research of LLMs and achieves PEFT with fewer parameters combined with LoRA. During the finetuning process, the parameters of LLMs and entity embeddings are fixed, allowing only gradient backpropagation through the parameters of adapters. This enables the use of external knowledge to unleash the potential of LLMs.

3 KnowLA

Considering that the hidden states in Transformer layers encapsulate the parameterized knowledge of an LLM (Li et al., 2023), we propose fusing entity embeddings in a KG with the hidden states of an LLM during PEFT. KnowLA inserts an adaptation layer into an LLM, as shown in Figure 1.

Given a KG, we adopt a representation learning model, e.g., TransE (Bordes et al., 2013), to train its entity embeddings. The pre-trained embedding of entity e is denoted by $\mathbf{e}$. For an input question $Q = \{t_i\}_{i=1}^{n}$ to an LLM, each token t_i may be linked to a set of entities $E(t_i)$ in the KG. Our key idea is to enhance PEFT by injecting the embedding $\mathbf{e}_i$ for each $e_i \in E(t_i)$ into the representation in the LLM. This method can be divided into three modules: (i) *Entity linking*, which links the tokens in a question to entities in the KG. (ii) *Knowledge mapping and injection*, which maps the KG embedding space to the LLM's representation space and infuses the entity embeddings corresponding to a specific token in the question. (iii) *Knowledge fusion*, which integrates each token representation with its entity embedding. Given the powerful abilities, popularity, and open-source nature of the LLaMA family (Touvron et al., 2023a,b), we consider it the foundation to build our KnowLA.

3.1 Entity Linking

Given an input text, we return its synsets as candidate entities in a KG. We use the text-rank algorithm to recognize important tokens and link the recognized tokens to the KG by string matching. We also collect a set of synonyms for each related entity. Based on the byte pair encoding (BPE) algorithm (Sennrich et al., 2016), each token is divided into multiple subwords sharing the same entity candidate. After this step, we obtain relevant entities in the KG for the important tokens in the text. Each entity is associated with a pre-trained embedding.

3.2 LLM Encoding

Given an LLM, e.g., Llama 2, it first encodes the input text to get embeddings for prompts and questions. Specifically, for a prompt p, the LLM first converts it into $Q = ([s], p, [/s])$. The decoder of the LLM tokenizes Q with the BPE algorithm. After tokenization, Q turns into $\{\mathbf{h}_i\}_{i=1}^{m} \in \mathbb{R}^{d_1}$, which is taken as input to the LLM.

3.3 Knowledge Mapping and Injection

The text representation of the l-th decoder layer in the LLM is denoted by $\mathbf{h}^l$. In the knowledge mapping module, to align with the pre-norm mode adopted by the decoder and mitigate the issues of gradient vanishing or exploding, we apply RMSNorm (Zhang and Sennrich, 2019) to the input $\mathbf{h}^l$ received by the decoder. We also map the semantic space of entity embeddings to the semantic space of the LLM for transformation, aiming to improve knowledge injection and fusion.

The BPE encoding method employed by many LLMs would let each token have multiple sub-tokens after encoding. Let $\{\mathbf{h}_i^l\}_{i=1}^{k}$ denote the sub-token embeddings, where k is the number. To better calculate the relevance between different entities and the given word, we unify the representations of the k sub-tokens as $\mathbf{u}_i$ using mean pooling:

$$\mathbf{u}_i = \mathrm{AvgPooling}(\mathbf{h}_1^l, \ldots, \mathbf{h}_k^l). \quad (1)$$

As LLMs are employed for handling complex natural language tasks, it is essential to have input dimensions sufficiently large to accommodate the intricacies. To enhance the expressive ability of entity representation $\mathbf{e}_i$ and align with the semantic space of the LLM, we expand its dimension to enrich the representation of $\mathbf{e}_i$:

$$\mathbf{e}_i = \mathbf{W}_d\big(\mathrm{SwiGLU}(\mathbf{W}_u\,\mathbf{e}_i + \mathbf{b}_u)\big), \quad (2)$$

where $\mathbf{W}_d \in \mathbb{R}^{d_1 \times d_3}$, $\mathbf{W}_u \in \mathbb{R}^{d_3 \times d_2}$, and $\mathbf{b}_u \in \mathbb{R}^{d_3}$ are trainable weights. SwiGLU (Shazeer, 2020) is an activation function.

3.4 Knowledge Fusion

To mitigate the risk of the LLM encountering unfamiliar entities during finetuning in downstream tasks, as well as to ensure the extracted entities are

relevant to the input tokens, we follow (Yang et al., 2019) and introduce a knowledge sentinel $\overline{\mathbf{e}}$. First, we calculate the similarities of each token with its relevant entities and the knowledge sentinel:

$$\alpha_{ij} = \frac{\exp(\mathbf{e}_j \cdot \mathbf{u}_i)}{\sum_j \exp(\mathbf{e}_j \cdot \mathbf{u}_i) + \exp(\overline{\mathbf{e}} \cdot \mathbf{u}_i)}, \quad (3)$$

$$\beta_i = \frac{\exp(\overline{\mathbf{e}} \cdot \mathbf{u}_i)}{\sum_j \exp(\mathbf{e}_j \cdot \mathbf{u}_i) + \exp(\overline{\mathbf{e}} \cdot \mathbf{u}_i)}, \quad (4)$$

where α_{ij} represents the relevance between the i-th token and the j-th entity. β_i represents the relevance between the i-th token and the knowledge sentinel. Here, we constrain that $\sum_j \alpha_{ij} + \beta_i = 1$. Then, we fuse $\mathbf{u}_i$ with its relevant entities:

$$\overline{\mathbf{u}}_i = \sum_j \alpha_{ij}\, \mathbf{e}_j + \beta_i\, \overline{\mathbf{e}}, \quad (5)$$

$$\overline{\mathbf{h}}_i = \theta\, \mathrm{SwiGLU}\big(\mathbf{W}_m[\overline{\mathbf{u}}_i; \mathbf{u}_i] + \mathbf{b}_m\big) + \mathbf{h}_i, \quad (6)$$

where θ serves as a trainable balancing factor to equalize the impact of KG and text. $\mathbf{W}_m \in \mathbb{R}^{2d_1 \times d_1}$ and $\mathbf{b}_m \in \mathbb{R}^{d_1}$ are trainable weights. During knowledge fusion, all the k sub-token embeddings $\{\mathbf{h}_i\}_{i=1}^k$ share the same $\overline{\mathbf{u}}_i$. $\overline{\mathbf{h}}_i$ denotes the final representation of knowledge injection and serves as the output of the current adapter, which is passed as input to the next layer of the decoder.

Similar to other parameter-efficient modules like LoRA (Hu et al., 2022), KnowLA achieves the alignment between KG knowledge and textual semantics by freezing the LLM during finetuning. It can also be used in conjunction with LoRA to achieve efficient learning of the LLM with a limited number of parameters. The effectiveness of this module is shortly assessed in the experiments.

4 Experiments

4.1 Baselines and Implementation

We consider the following LLMs with 7B parameters as foundation models in our main experiments:

- **Llama 2** is a collection of open-source LLMs trained on public datasets with trillions of tokens. We use the Llama 2-7B model.
- **Alpaca2** (Taori et al., 2023) is a Llama 2 variant finetuned with 52,000 instruction-following demonstrations using LoRA.

Given that there are currently no knowledge injection methods for PEFT, we choose retrieval augmented generation (RAG) methods as baselines:

- **Contriever** (Izacard et al., 2022) is pretrained using English Wikipedia. We use it to retrieve triples from KGs and passages from Wikipedia to augment the input of the LLM.
- **KAPING** (Baek et al., 2023) retrieves relevant triples from KGs to improve the KBQA task. We use KAPING to enhance LLMs on knowledge-relevant tasks.

In our main experiments, we use the official hyperparameters and instruction data of Alpaca2 to finetune Llama 2-7B with LoRA and KnowLA. Our layer is inserted after the 32nd layer of Llama 2. We also consider LLaMA 1 and the instruction data of Vicuna2 (Chiang et al., 2023) in Sect. (4.10).

During the training process, we set the batch size to 128 and the learning rate to 3e-4, and use the AdamW optimizer to train 3 epochs. We keep the hyperparameters the same for different models to ensure the fairness of the experiment. We also keep the input prompts the same in the experiments. To study the impact of the number of trainable parameters, we train two LoRA models with different ranks: $r = 16$ and 32. They both perform better than ranks $r = 4, 8$ on most datasets. All models are finetuned on A800 GPUs. The code is publicly available at our GitHub repository.[1]

4.2 Datasets and Settings

We consider three types of tasks: multi-choice QA, closed-book QA, and truthful QA. We pick CommonsenseQA (Talmor et al., 2019) and SIQA (Sap et al., 2019) as the multiple-choice QA datasets, and choose 15 challenging multi-choice tasks from BIG-Bench Hard (BBH) (Suzgun et al., 2023). We use WebQuestionSP (Yih et al., 2016) and TriviaQA (Joshi et al., 2017) for closed-book QA evaluation. We also use TruthfulQA (Lin et al., 2022) to evaluate whether KnowLA is truthful in generating answers to questions. Appendix A complements more details. To assess the direct improvement of our KnowLA to enhance PEFT, we employ zero-shot settings for all tasks.

4.3 KGs and Configurations

We select WordNet (Miller, 1995), ConceptNet (Speer et al., 2017), and Wikidata (Vrandecic and Krötzsch, 2014) as the KGs in our method. See Appendix A for more descriptions.

For RAG methods, we consider the overlap between questions and knowledge sources. For multi-

[1] https://github.com/nju-websoft/KnowLA

Methods	#Parameters	CommonsenseQA		SIQA		BIG-Bench Hard	
		Accuracy	Score	Accuracy	Score	Accuracy	Score
Llama 2 (7B)	7B	45.37	36.40	46.42	40.58	26.95	24.87
Alpaca2 ($r = 16$)	+0.24%	56.92	46.55	52.61	46.18	28.93	**25.42**
Alpaca2 ($r = 32$)	+0.50%	57.90	46.81	53.17	46.21	28.79	25.36
Contriever (WordNet)	+0.50%	57.15	46.09	52.58	46.13	-	-
Contriever (ConceptNet)		57.06	45.30	52.51	45.51	-	-
KAPING (WordNet)	+0.50%	57.21	45.91	52.51	45.89	-	-
KAPING (ConceptNet)		57.58	45.64	52.66	46.15	-	-
KnowLA (Random)	+0.55%	57.49	47.82	52.61	46.56	29.26	25.34
KnowLA (WordNet)		58.07	**48.35**	**53.22**	46.76	30.00	25.39
KnowLA (ConceptNet)		**58.39**	48.19	**53.22**	**46.81**	**30.19**	25.29
KnowLA (Wikidata)		57.90	47.39	53.21	46.64	29.39	**25.42**

Table 1: Multi-choice QA results on CommonsenseQA, SIQA, and BBH. For KnowLA, the rank of LoRA is $r = 16$. The percentage of trainable parameters are similar in Tables 2 and 3.

choice QA, we use ConceptNet and WordNet. For TriviaQA, we use Wikidata and Wikipedia.

For KG embeddings, we follow (Zhang et al., 2019) and pre-train entity embeddings with TransE (Bordes et al., 2013) as the external knowledge. The maximum number of relevant entities selected for each textual token in a question is set to 5. Furthermore, we evaluate the side effects and additional latency of KnowLA. See Appendix B and Appendix C for more details.

4.4 Experiments on Multi-choice QA

To evaluate the effectiveness and robustness of KnowLA, we compare it to Llama 2 and Alpaca2 ($r = 16, 32$) on multi-choice QA. In addition to accuracy, we follow (Shwartz et al., 2020) and compute scores using cross entropy, which indicate the confidence of a model for correct answers. We use three KGs: WordNet, ConceptNet, and Wikidata. We also consider randomly initialized vectors as a baseline of KG embeddings.

Table 1 presents the results. Our KnowLA variants show the best performance across the three datasets. Furthermore, Alpaca2 ($r = 32$) outperforms Alpaca2 ($r = 16$), because more trainable parameters usually lead to better performance.

KAPING generally performs better than Contriever on CommonsenseQA. This indicates that the RAG methods rely on the quality of prompts retrieved from the knowledge sources. Both KAPING and Contriever are inferior to Alpaca2 ($r = 32$) on CommonsenseQA and SIQA, as invalid prompts may cause damage to the performance.

KnowLA is different from RAG methods. RAG methods retrieve text information to augment the input of LLMs, while KnowLA uses KG embeddings to improve the effectiveness of PEFT. KnowLA works in the finetuning phase of LLMs and does not change the input of LLMs. Our method with LoRA ($r = 16$) achieves better performance than all baselines, indicating that it can effectively work with PEFT to inject knowledge. Specifically, when combined with ConceptNet, it achieves an accuracy increase from 56.92% to 58.39% on CommonsenseQA, from 52.61% to 53.22% on SIQA, and from 28.93% to 30.19% on BBH. Since ConceptNet stores rich conceptual knowledge and more relation types compared to WordNet, its entity embeddings can better enhance Llama 2's reasoning ability. Furthermore, ConceptNet recognizes more relevant entities in the question than Wikidata. This suggests that extensive entity coverage in KnowLA brings a significant performance increase.

Additionally, the performance of KnowLA (random) is inferior to that of KnowLA with KGs, highlighting the greater utility of entity knowledge for LLMs. Based on the scores of each model on the correct answers, it can be seen that after incorporating KnowLA, all models assign higher confidence to the correct answers. Therefore, KnowLA can offer a certain degree of improvement for LLMs in commonsense reasoning.

4.5 Experiments on Closed-book QA

We evaluate KnowLA using WebQuestionSP and TriviaQA. Following the answer matching strategy in (Tan et al., 2023), we use the subtree labels provided by the constituent tree to extract all noun phrases from the textual answers, calculate their similarities, and determine the correctness of answers exceeding a certain threshold (e.g., 0.5).

The accuracy results are shown in Table 2. We

Case 1: Who is featured on Puff Daddy's Can't Hold Me Down?

Answer: Mase

Alpaca2: Puff Daddy's 1997 single "Can't Hold Us Down" features R&B singers Faith Evans and 112. It was released as the second single from Puff Daddy's debut album, No Way Out.

KnowLA (ConceptNet): The song "Can't Hold Me Down" by Puff Daddy features R&B singer Faith Evans and rapper Mase. The song was released in 1997 as the lead single from Puff Daddy's debut studio album, No Way Out.

KnowLA (WordNet): The song "Can't Hold Me Down" by Puff Daddy features R&B singer Faith Evans, rapper Mase, and R&B singer 112. It was released in 1997 as the lead single from Puff Daddy's debut album, No Way Out.

Case 2: In Lewis Carroll's poem The Hunting of the Snark, what did the elusive, troublesome snark turn into to fool hunters?

Answer: Boojum

Alpaca2: The elusive, troublesome snark was able to fool the hunters by transforming itself into a variety of shapes and forms. The snark was able to take the shape of a dragon, a boa constrictor, a rhinoceros, a hippopotamus, a crocodile, and a camel.

KnowLA (ConceptNet): The snark was able to fool hunters by transforming into a variety of shapes and forms. The snark first turned into a Boojum, a creature that could only be seen by those who believed in it.

KnowLA (Wikidata): The snark was able to fool the hunters by transforming into a variety of shapes and forms. The snark was able to take the shape of a beetle, a butterfly, a snake, a fish, a bird, and a mouse, all in an attempt to escape the hunters.

Figure 2: Examples of Alpaca2 and KnowLA for TriviaQA.

Methods	WebQuestionSP	TriviaQA
Alpaca2 ($r = 16$)	67.55	68.70
Alpaca2 ($r = 32$)	67.43	67.97
Contriever (Wikipedia)	-	68.71
KAPING (Wikidata)	67.11	66.05
KnowLA (Random)	67.68	69.34
KnowLA (WordNet)	67.43	69.27
KnowLA (ConceptNet)	**68.12**	**69.40**
KnowLA (Wikidata)	67.49	68.92

Table 2: Closed-book QA results on WebQuestionSP and TriviaQA. For KnowLA, $r = 16$.

find that Alpaca2 ($r = 16$) obtains better performance than Alpaca2 ($r = 32$). The reason may be that more parameters in LoRA are prone to overfitting in the closed-book QA tasks. Moreover, Contriever (Wikipedia) only slightly exceeds Alpaca2 ($r = 16$) and performs better than KAPING. This is because KAPING cannot guarantee the correctness of the extracted triples.

According to the results, KnowLA combined with WordNet improves the results from 68.70% to 69.27% on TriviaQA, while combined with ConceptNet, the performance is further enhanced to 69.40%. This indicates that the parameterized entity embeddings can enrich the textual representations. The experimental results demonstrate that the knowledge-enhanced textual representations after finetuning with LoRA can help mitigate the hallucination problem of Llama 2 to some extent.

On WebQuestionSP, KnowLA (WordNet) and KnowLA (Wikidata) produce similar results. Also, the two Alpaca2 models with different ranks perform similarly. This suggests that the reasoning ability of Alpaca2 is good on this task, and the performance does not change significantly after knowledge enhancement with KnowLA. We attribute this bottleneck to the model size and the training data of Llama 2 and Alpaca2.

Methods	BLEU	Rouge-1	Rouge-2	Rouge-L
Alpaca2 ($r = 16$)	0.1657	0.4094	0.2831	0.3892
Alpaca2 ($r = 32$)	0.1637	0.4048	0.2802	0.3851
KnowLA (Random)	0.1677	0.4110	0.2850	0.3897
KnowLA (WordNet)	0.1714	0.4143	0.2874	0.3927
KnowLA (ConceptNet)	**0.1747**	**0.4190**	**0.2922**	**0.3975**
KnowLA (Wikidata)	0.1703	0.4135	0.2895	0.3931

Table 3: Results on TruthfulQA. For KnowLA, $r = 16$.

4.6 Experiments on TruthfulQA

We use TruthfulQA to measure whether KnowLA is truthful in generating answers to questions. Here, we evaluate the content generated by the models based on the best answer provided by TruthfulQA, using the commonly used metrics BLEU, Rouge-1, Rouge-2, and Rouge-L. Table 3 shows the results.

Alpaca2 ($r = 32$) still underperforms Alpaca2 ($r = 16$). This further substantiates our conclusion that larger parameters do not necessarily guarantee the accuracy and reliability of the model's output. KnowLA (ConceptNet) performs best among these models, which indicates that the integration of our KnowLA with LoRA can mitigate the hallucination problem of Llama 2 to some extent and generate content of better quality.

Besides, we observe that KnowLA (ConceptNet) outperforms KnowLA (WordNet) in all evaluation tasks, and KnowLA (WordNet), in turn, surpasses KnowLA (Wikidata). This further indicates that the commonsense knowledge within ConceptNet is more suitable for both LoRA and Llama 2.

4.7 Case Study

Figure 2 presents some improved results of Alpaca2 by incorporating WordNet, ConceptNet, and Wikidata in KnowLA. In Case 1, we discover that after integrating ConceptNet and WordNet with KnowLA, the response precisely describes the correct answers. The contents generated by KnowLA (ConceptNet) and KnowLA (WordNet) are very similar. The content generated by Alpaca2 not only misses significant answers but also misinterprets the song "Can't Hold Me Down" in the question. Therefore, we believe that KnowLA helps the model better understand questions.

By examining the answers of the three models in Case 2, it can be observed that Alpaca2 does not provide an accurate and relevant response, which is similar to the content generated by KnowLA (Wikidata). They both generate deceptive answers. However, after incorporating ConceptNet, KnowLA accurately provides the correct answer in the response. According to Table 2, we believe that the enhancement is not accidental. Moreover, by examining the token-to-entity linking results, we find that *the answer entity "Boojum" does not exist in ConceptNet.* Therefore, we conclude that KnowLA can stimulate the underlying reasoning abilities of LLMs by working with LoRA.

4.8 Why Knowledgeable Adaptation Works?

We delve into why KnowLA collaborates effectively with LoRA, focusing on space alignment of KGs and LLMs, and knowledge recall in LLMs.

Perspective of Space Alignment. Our KnowLA incorporates pre-trained KG embeddings into a pre-trained LLM for instruction tuning with LoRA. We hereby investigate whether the two heterogeneous representation spaces of the KG and the LLM are aligned, to understand how KnowLA works. The results are illustrated in Figure 3, where the last column represents the "sentinel" entity. We first acquire the representations of the input tokens in a specific layer, e.g., the 32nd layer. Then, we retrieve the top five similar entity embeddings in the KG for each token. Next, to establish the relevance of each token and its corresponding entities, we calculate the attention weights between them. A larger weight suggests a stronger semantic correlation between the token and the mapped entity.

In the case of Llama 2 (depicted in the left part of Figure 3), the similarities between entity embeddings and token representations appear to be random, lacking any discernible patterns. However, after applying KnowLA, the results show improved accuracy specifically for the most relevant entities (i.e., e_1 on the x-axis). For token "underrated", the relevant entities in ConceptNet are "underrated", "underrate", etc. After finetuning, the token "underrated" exhibits the highest correlation with the entity "underrated". This observation indicates that KnowLA can effectively align the KG and the LLM through instruction tuning with LoRA.

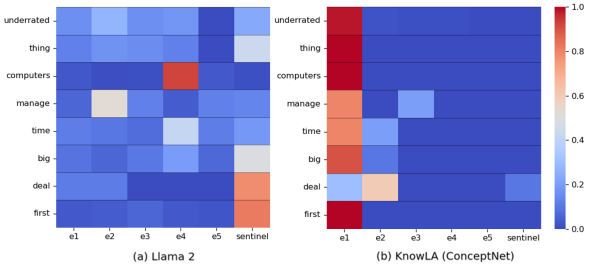

Figure 3: The similarity heatmap between the output representations of text tokens and their corresponding entity embeddings. The x-axis denotes the top-5 similar entities with tokens on the y-axis. (a) The left heatmap presents the similarity of Llama 2 without finetuning, and (b) the right heatmap presents the similarity after finetuning with our KnowLA (ConceptNet).

Perspective of Knowledge Recall. We study the role of KnowLA in activating an LLM's knowledge. According to (Li et al., 2023; Geva et al., 2021; Meng et al., 2022), the feed-forward network (FFN) layers, which constitute two-thirds of an LLM's parameters, primarily capture its own knowledge. So, we explore the impact of KnowLA on the FFN layers to see how KnowLA affects these layers in activating knowledge stored in the LLM.

We compute the differences between the hidden state representations of the last token before and after each FFN layer in the LLM. We analyze the trends in differences of all 32 layers after inserting KnowLA. We use the 100 questions from TriviaQA as queries to explore the knowledge stored in the FFN layers of Llama 2 (7B). The last token representation in each input aggregates information from all tokens. According to (Li et al., 2023), there is a positive correlation between the similarity of hidden states and the consistency of knowledge. Intuitively, we believe that higher differences in representations indicate the model's ability to capture more information from the FFN layers. Therefore, we extract the representations of the last token before and after each FFN layer

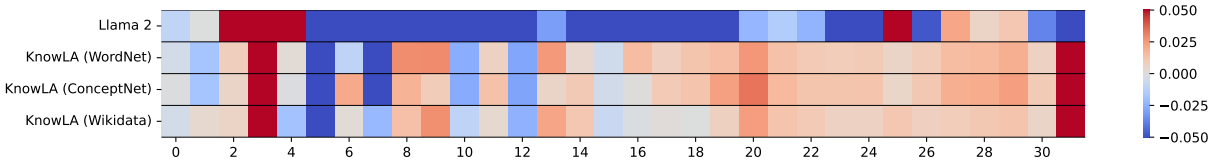

Figure 4: The heatmap indicates the capabilities of KnowLA and Llama 2 in capturing knowledge compared to Alpaca2, which is measured by averaging the changes in cosine similarities of the last token representations from 100 queries across all FFN layers. The x-axis denotes the 32 layers of Llama 2.

and compute the cosine similarities for Llama 2, KnowLA, and Alpaca2, which are denoted by s_1, s_2, and s_3, respectively. Given the token similarities, we further evaluate the capacities of KnowLA and Llama 2 in capturing hidden knowledge. The capacities are measured by $s_3 - s_2$ and $s_3 - s_1$.

The results are shown in Figure 4. The red color indicates that the representation of the last token, after introducing KnowLA and undergoing the FFN layers, exhibits a greater change compared to that of Alpaca2. The blue color shows the opposite. We think the representations with greater changes capture more internal knowledge.

After introducing entity embeddings, KnowLA enables the LLM to activate richer knowledge at the FFN layers. In contrast, Llama 2 captures less knowledge than Alpaca2. According to the work (Geva et al., 2021), lower FFN layers tend to capture shallow knowledge patterns, while higher FFN layers learn more semantic patterns. Our KnowLA demonstrates enhanced knowledge activation capabilities at the higher layers, and thus achieves superior results over Alpaca2. By examining the differences in similarity across the last 16 layers, we find that KnowLA (ConceptNet) shows the greatest similarity difference in the three KGs and performs best on TriviaQA. This further emphasizes that the introduction of ConceptNet substantially activates more knowledge stored internally in Llama 2.

4.9 Impact of KG Embedding Models

The KG embedding learning models are used to learn entity embeddings (Bordes et al., 2013; Nickel et al., 2011; Sun et al., 2019; Chen et al., 2023). We study the impact of embedding learning models for KnowLA. We obtain entity embeddings of ConceptNet by three representative KG embedding models: RESCAL (Nickel et al., 2011), TransE (Bordes et al., 2013), and RotatE (Sun et al., 2019). We show the results of KnowLA with these embeddings on the CommonsenseQA, SIQA, and BBH datasets in Table 4.

	CommonsenseQA		SIQA		BBH	
	Accuracy	Score	Accuracy	Score	Accuracy	Score
RESCAL	**58.39**	46.71	52.10	44.91	27.50	**25.96**
TransE	**58.39**	**48.19**	**53.22**	**46.81**	**30.19**	25.29
RotatE	57.58	46.05	52.00	44.65	27.31	24.94

Table 4: Comparison of KG embedding learning models on CommonsenseQA, SIQA, and BBH, which are pretrained on ConceptNet for Llama 2.

We can observe that the entity embeddings obtained by TransE achieve favorable results. This is attributed to the fact that the TransE embeddings have a good generalization ability and are thus more suitable for Llama 2. RotatE employs complex vector representations for entities and obtains subpar results on Llama 2. This suggests that aligning the complex space of entities with the semantic space of Llama 2 during finetuning is challenging, leading to a loss of original entity knowledge.

4.10 Robustness of KnowLA

We evaluate the robustness of KnowLA against three factors: On the foundation model side, we use LLaMA 1 as another LLM. On the instruction data side, we finetune Llama 2 using the Vicuna multi-round dialog data (Chiang et al., 2023) to get Vicuna2 and KnowLA (Vicuna2). On the PEFT method side, we use AdaLoRA (Zhang et al., 2023) to replace LoRA and get Alpaca2 (AdaLoRA) and KnowLA (AdaLoRA). On the rank side, we finetune Llama 2 using the Alpaca data with rank $r = 8$ and get Alpaca2 ($r = 8$) and KnowLA ($r = 8$).

Table 5 lists the performance of the above models on the commonsense reasoning dataset CommonsenseQA. We can see that the three KnowLA variants still outperform all baselines. This experiment shows that KnowLA is robust and can bring stable improvement when combined with different LLMs, instruction data, PEFT methods, and ranks.

	Methods	Accuracy	Score
LLM side	Alpaca1	56.59	46.03
	KnowLA (LLaMA 1)	**57.74**	**46.81**
Data side	Vicuna2	51.52	42.31
	KnowLA (Vicuna2)	**53.56**	**49.09**
PEFT side	Alpaca2 (AdaLoRA)	57.58	**46.67**
	KnowLA (AdaLoRA)	**57.66**	46.30
Rank side	Alpaca2 ($r = 8$)	56.92	46.25
	KnowLA ($r = 8$)	**57.74**	**46.93**

Table 5: Results with different LLMs, instruction data, PEFT methods, and ranks on CommonsenseQA

5 Conclusion

In this paper, we propose a knowledgeable adaptation method KnowLA. It works with LoRA and injects entity embeddings into an LLM in the PEFT process. Compared to Alpaca2, which is finetuned with LoRA alone, KnowLA with Llama 2 shows better performance on six benchmark datasets. We show that pre-trained KG embeddings are compatible with Llama 2. Moreover, we find that KnowLA can align the KG space and the LLM space, and activate the hidden knowledge related to input in LLMs, thereby achieving improved performance.

Limitations

Currently, our work only incorporates one KG to enhance PEFT. As KGs are incomplete by nature, integrating multiple KGs into our method may further improve performance with knowledge fusion and transfer. Recent work (Huang et al., 2022) reveals that multi-source KG embeddings are more expressive than the embeddings of a single KG. We show preliminary results in Appendix E.1 and will study multi-source KnowLA in future work.

We have not attempted other LLMs such as ChatGLM (Zeng et al., 2023) in this work. In the future, we will consider how to efficiently inject KG knowledge with smaller parameters. Additionally, we have observed that, with the introduction of random perturbations, Llama 2 seems to outperform Alpaca2 on some tasks. This discovery may provide interesting directions for future research.

Ethical Considerations

LLMs may produce incorrect and potentially biased content. Experiments show that our method can alleviate this problem to a certain extent, but LLMs will inevitably generate offensive answers. Therefore, extreme caution should be exercised if deploying such systems in user-facing applications. All datasets and models used in this work are publicly available under licenses.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272219) and the CCF-Tencent Rhino-Bird Open Research Fund.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106, Toronto, Canada. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Zirui Chen, Xin Wang, Chenxu Wang, and Zhao Li. 2023. PosKHG: A position-aware knowledge hypergraph model for link prediction. *Data Sci. Eng.*, 8(2):135–145.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yuanning Cui, Yuxin Wang, Zequn Sun, Wenqiang Liu, Yiqiao Jiang, Kexin Han, and Wei Hu. 2023. Lifelong embedding learning and transfer for growing knowledge graphs. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 4217–4224. AAAI Press.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314.

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*, pages 4171–4186.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 320–335.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Zijie Huang, Zheng Li, Haoming Jiang, Tianyu Cao, Hanqing Lu, Bing Yin, Karthik Subbian, Yizhou Sun, and Wei Wang. 2022. Multilingual knowledge graph completion with self-supervised adaptive graph alignment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 474–485, Dublin, Ireland. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4*, pages 1601–1611.

Anne Lauscher, Ivan Vulic, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavas. 2019. Informing unsupervised pretraining with external linguistic knowledge. *CoRR*, abs/1909.02339.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sensebert: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4656–4667.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4582–4597, Online.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. PMET: precise model editing in a transformer. *CoRR*, abs/2308.08742.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, PA, USA.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: enabling language representation with knowledge graph. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2901–2908.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. KG-BART: knowledge graph-augmented BART for generative commonsense reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, Virtual Event, February 2-9, 2021*, pages 6418–6425.

Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. 2021. KELM: knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *CoRR*, abs/2109.04223.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NeurIPS*.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816.

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

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany*.

Noam Shazeer. 2020. GLU variants improve transformer. *CoRR*, abs/2002.05202.

Sheng Shen, Le Hou, Yanqi Zhou, Nan Du, Shayne Longpre, Jason Wei, Hyung Won Chung, Barret Zoph, William Fedus, Xinyun Chen, Tu Vu, Yuexin Wu, Wuyang Chen, Albert Webson, Yunxuan Li, Vincent Zhao, Hongkun Yu, Kurt Keutzer, Trevor Darrell, and Denny Zhou. 2024. Mixture-of-experts meets instruction tuning:a winning combination for large language models. In *ICLR*.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4615–4629.

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.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4396–4402.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. 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, Toronto, Canada, July 9-14, 2023*, pages 13003–13051.

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*, pages 4149–4158, Minneapolis, Minnesota.

Yiming Tan, Dehai Min, Yu Li, Wenbo Li, Nan Hu, Yongrui Chen, and Guilin Qi. 2023. Evaluation of chatgpt as a question answering system for answering complex questions.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

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, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1405–1418.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10658–10671.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Xinyi Wang, Zitao Wang, Weijian Sun, and Wei Hu. 2022. Enhancing document-level relation extraction by entity knowledge injection. In *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings*, volume 13489 of *Lecture Notes in Computer Science*, pages 39–56.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12360–12371.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019*, pages 1441–1451.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China.

A Datasets and KGs

The details of the datasets are described as follows:

- In **CommonsenseQA** (Talmor et al., 2019), each sample consists of a question, five candidate answers, and a correct answer. To run LLMs for CommonsenseQA, we adopt the same setting as in (Shwartz et al., 2020) and consider it as a text completion task. We test the LLMs with the validation dataset.
- **SIQA** (Sap et al., 2019) is a QA dataset about social commonsense, where each sample consists of a question, three candidate answers, and a correct answer. To evaluate prompt-based methods, we do not use the provided knowledge in the dataset. The settings are the same as in CommonsenseQA. We test the LLMs with the validation dataset.
- **BBH** (Suzgun et al., 2023) is a popular benchmark that focuses on tasks challenging for LLMs. To compare scores of different methods on correct answers, we select 15 multiple-choice QA datasets from this benchmark.
- **WebQuestionSP** (Yih et al., 2016) is a KBQA dataset that enhances the WebQuestion dataset

Datasets	Alpaca2 ($r = 16$)	KnowLA (ConceptNet)
Temporal sequences	14.80	15.20
Date understanding	72.00	73.20
Geometric shapes	9.20	19.20
Snarks	51.12	53.37
Logical deduction	35.20	36.40

Table 6: Results on knowledge-unrelated tasks

by annotating each answer with corresponding SPARQL queries and removing ambiguous, unclear, or unanswerable questions. In this paper, we treat it as a closed-book QA task.

- **TriviaQA** (Joshi et al., 2017) includes 95K question-answer pairs authored by trivia enthusiasts, which provide high-quality distant supervision for answering the questions. In this paper, we treat it as a closed-book QA task and select 7,500 questions from TriviaQA to test LLMs.
- **TruthfulQA** (Lin et al., 2022) is a benchmark to measure whether a language model is truthful in generating answers to questions.

The used KGs are introduced as follows:

- **WordNet** (Miller, 1995) is a lexical KG in English. Nouns, verbs, adjectives, and adverbs are arranged into synsets, each denoting a separate notion.
- **ConceptNet** (Speer et al., 2017) is a multilingual conceptual KG of things people know and computers should know.
- **Wikidata** (Vrandecic and Krötzsch, 2014) is a factual KG across diverse domains. It encompasses various entity types, including individuals, places, concepts, etc.

B Knowledge-Unrelated Tasks

We analyze the side effects of KnowLA on knowledge-unrelated tasks. In this experiment, five knowledge-unrelated tasks from BBH are picked. The results in Table 6 show that even if these tasks are knowledge-unrelated, our KnowLA can still improve the LLM. This is due to the enhanced ability of the LLM to activate its own knowledge.

C Additional Latency on Efficiency

Retrieving the embeddings of related entities during each finetuning step would slow down the training process. We move it to the data processing step. We use eight workers to process 50,538 training samples in parallel. During inference, we compare the overall inference time of KnowLA (ConceptNet) and Alpaca2 on CommonsenseQA using an A6000 GPU card. Table 7 shows the results.

Models	Data processing	Inference
Alpaca2	9 s	19.02 min.
KnowLA (ConceptNet)	16 s	19.20 min.

Table 7: Time overhead of Alpaca2 and KnowLA

Prompts	Alpaca2 ($r = 16$)	KnowLA (ConceptNet)
Below is an instruction that describes a task, paired with an input that provides further context. Choose a correct answer that appears in the candidate answers.	56.92	58.39
Below is an instruction that describes a task, paired with an input that provides further context. Please answer the following question.	52.74	54.95
Below is an instruction that describes a task, paired with an input that provides further context. Give an answer that appropriately completes the question.	53.73	56.10
Please answer the following question.	55.20	56.35

Table 8: Accuracy of KnowLA when using different prompts on CommonsenseQA

Alpaca2 spends 9 seconds on data processing, while KnowLA (ConceptNet) spends 16 seconds. During inference, KnowLA (ConceptNet) takes 19 minutes and 12 seconds, while Alpaca2 takes 19 minutes and 1 second. We believe that the additional latency caused by KnowLA is tolerable compared to the performance boost.

D Robustness to Different Prompts

We try different prompts to evaluate the robustness of KnowLA. Table 8 compares the accuracy of Alpaca2 ($r = 16$) and KnowLA on CommonsenseQA with different prompts. KnowLA outperforms Alpaca2 on all prompts, indicating its good robustness. We use the first prompt in the main experiments due to its superior performance.

E Discussion on Extension of KnowLA

We hereby discuss the extension of KnowLA to integrate multiple KGs for PEFT and incrementally incorporate knowledge updates in a KG.

E.1 Multiple KGs

To leverage multiple KGs and try to benefit from their potential knowledge transfer, we design a sim-

Methods	Accuracy	Score
KnowLA (WordNet)	58.07	**48.35**
KnowLA (ConceptNet)	**58.39**	48.19
KnowLA (Wikidata)	57.90	47.39
KnowLA (multiple KGs)	57.61	47.24

Table 9: Results of multiple KGs on CommonsenseQA

ple baseline that merges different KGs into a large graph using entity alignment (Sun et al., 2018) and learns entity embeddings from this large KG. If there is no entity alignment, we use entity embeddings from these KGs simultaneously. For balanced training, we limit the maximum number of related entities from each KG to two, and each token has up to six entity embeddings.

In this experiment, we merge WordNet, ConceptNet, and Wikidata. According to Table 9, the accuracy of this baseline on CommonsenseQA is 57.61, which is slightly lower than the result of KnowLA (ConceptNet). We think this straightforward baseline may not effectively leverage knowledge transfer between KGs. A more promising mechanism, e.g., Mixture of Experts (Shen et al., 2024), is necessary to combine multiple KGs. KnowLA is adaptable to integrate this improvement. We leave this direction for future work.

E.2 Knowledge Updates

Knowledge updates for KnowLA require the support of incremental learning for KGs and LLMs. When some new entities and triples are added to a KG, only a small number of parameters need to be re-trained to complete knowledge updates by using lifelong KG embedding learning (Cui et al., 2023) and continual PEFT (Wang et al., 2023).