InfuserKI: Enhancing Large Language Models with Knowledge Graphs via Infuser-Guided Knowledge Integration

Fali Wang
The Pennsylvania State University
University Park, USA
fqw5095@psu.edu

Runxue Bao
GE Healthcare
Bellevue, USA
runxue.bao@gehealthcare.com

Suhang Wang
The Pennsylvania State University
University Park, USA
szw494@psu.edu

Wenchao Yu
NEC Laboratories America
Princeton, USA
wyu@nec-labs.com

Yanchi Liu
NEC Laboratories America
Princeton, USA
yanchi@nec-labs.com

Wei Cheng
NEC Laboratories America
Princeton, USA
weicheng@nec-labs.com

Haifeng Chen
NEC Laboratories America
Princeton, USA
haifeng@nec-labs.com

ABSTRACT

Though Large Language Models (LLMs) have shown remarkable open-generation capabilities across diverse domains, they struggle with knowledge-intensive tasks. To alleviate this issue, knowledge integration methods have been proposed to enhance LLMs with domain-specific knowledge graphs using external modules. However, they suffer from data inefficiency as they require both known and unknown knowledge for fine-tuning. Thus, we study a novel problem of integrating unknown knowledge into LLMs efficiently without unnecessary overlap of known knowledge. Injecting new knowledge poses the risk of forgetting previously acquired knowledge. To tackle this, we propose a novel Infuser-Guided Knowledge Integration (InfuserKI) framework that utilizes transformer internal states to determine whether to enhance the original LLM output with additional information, thereby effectively mitigating knowledge forgetting. Evaluations on the UMLS knowledge graph demonstrate that InfuserKI can effectively acquire new knowledge and outperform state-of-the-art baselines by 9%, in reducing knowledge forgetting.

VLDB Workshop Reference Format:
Fali Wang, Runxue Bao, Suhang Wang, Wenchao Yu, Yanchi Liu, Wei Cheng, and Haifeng Chen. InfuserKI: Enhancing Large Language Models with Knowledge Graphs via Infuser-Guided Knowledge Integration. VLDB 2024 Workshop: LLM+KG.

1 INTRODUCTION

Large Language Models (LLMs) have revolutionized fields such as Question Answering (QA), dialogue, and information retrieval, demonstrating impressive capabilities in various language tasks [34, 35]. However, LLMs can generate misleading or inaccurate texts, especially in knowledge-intensive tasks such as open-domain QA [17], due to lack of domain knowledge and catastrophic forgetting after fine-tuning [18, 40]. Updating and customizing LLMs with *domain knowledge integration* is highly valued across applications. For example, companies might personalize models with specific product knowledge, while hospitals could tailor models using their case data.

Knowledge Graphs (KGs) serve as an ideal source for enhancing domain-specific knowledge due to their structured and quantifiable knowledge units. To leverage this knowledge, several strategies have been developed. Generally, these include instruction tuning LLMs with extensive knowledge entity explanations [37], creating triplet-based pre-training tasks [28, 36, 42], employing KGs as external sources for retrieval [31, 39], and directly using parameter-efficient fine-tuning (PEFT) methods such as LoRA [13] and adapters [12] or model editing (ME) methods such as T-Patcher [14] to inject knowledge in a triplet-to-text way [6, 7, 27]. However, pre-training or fine-tuning LLMs with the entire KGs is not only time-consuming but also leads to data inefficiencies, especially when models relearn knowledge they already have. To address this issue, we focus on integrating new, previously unknown knowledge only. This precise focus, however, introduces the risk of catastrophic forgetting, where the addition of new knowledge may affect existing knowledge. Thus, we pose a novel research question: ***How can we efficiently integrate new knowledge from domain-specific KGs into LLMs while preventing catastrophic forgetting?***

In this work, we propose the Infuser-guided Knowledge Integration (**InfuserKI**) framework, specifically designed for integrating domain-specific knowledge from KGs into LLMs. Inspired by [1], which shows the LLM's internal states can indicate the truthfulness of its own generated sentences, our framework features an infusing mechanism that checks whether LLMs possess current knowledge. This enables the adaptive selection of supplementary information for known and unknown knowledge, effectively reducing the impact on existing knowledge and mitigating knowledge forgetting. Moreover, InfuserKI utilizes knowledge adapters to encode new knowledge

Proceedings of the VLDB Endowment. ISSN 2150-8097.

while preserving the original model parameters. To inject new knowledge only, the InfuserKI framework begins by detecting knowledge unknown to LLMs. Following [30, 44], we then generate *multiple-choice questions* for a knowledge triplet <h, r, t> using predefined relational templates, with an example in Fig. 1, and inject them by fine-tuning knowledge adapters. Our main contributions are:

(1) We study a novel problem of effectively integrating unknown knowledge from KGs into LLMs without affecting known knowledge.
(2) We propose a new knowledge integration framework InfuserKI, which enables the adaptive selection of supplementary information for known and unknown knowledge, effectively mitigating knowledge forgetting.
(3) Evaluations on UMLS reveal InfuserKI's effective knowledge integration with less forgetting, sustained performance on large-scale data and superior generality across unseen templates and downstream tasks.

2 RELATED WORK

Knowledge Integration. LLMs often produce seemingly accurate but incorrect answers due to missing knowledge. Addressing this, knowledge integration (KI) into LLMs has become popular. KGs, which capture wide or domain-specific knowledge, serve as an ideal option due to their structured and quantifiable knowledge units. KI from KGs usually occurs during pre-training or fine-tuning. For example, ERNIE [33] injects KG's embeddings, such as TransE [8], into models using an entity-token alignment masking loss. However, retraining is time-consuming. In fine-tuning, methods including JointLK [32] and GreaseLM [43] apply graph neural networks to model knowledge subgraphs, relying on KGs until inference. Fully fine-tuning models such as PMC-LLaMa [37] is computationally costly; therefore PEFT methods, for instance, LoRA [13] and Adapters [12], are more feasible. Based on these works, MoP [27], K-Adapter [36], and KB-adapters [7] inject knowledge directly into model parameters but risk catastrophic forgetting of unrelated knowledge [26]. Thus, we focus on adapter-based integration that minimizes the impact on unrelated knowledge.

Model Editing. Model Editing (ME) for LLMs falls into two categories: gradient-based and extension-based. Gradient-based methods, as described by Dai et al. [4], modify specific weights related to knowledge edits. ROME [25] and MEMIT [26] take this further by updating entire Feedforward Network (FFN) layers to enhance model editing. These methods, however, are limited in the number of edits or may require considerable time for execution. On the other hand, extension-based methods add new parameters to correct inaccurate information. CALINET [6] and T-Patcher [14] incorporate memory slots or trainable "patches" into final FFN outputs. GRACE [10] employs a key-value adapter with a deferral mechanism for the selective use of knowledge based on input. However, the adapter-based modules positioned in top transformer layers are designed to calibrate false facts. Instead, our method aims to infuse new knowledge by placing adapters throughout transformer layers.

Catastrophic Forgetting. Catastrophic forgetting occurs when learning new information causes a drastic loss of previously learned knowledge [29]. This becomes particularly evident in sequential inter-task learning, where acquiring new task knowledge leads to the forgetting of earlier task knowledge [24]. To tackle this, various methods are developed. Xuhong et al. [38] applies regularization constraints to minimize parameter changes when learning new tasks. Elastic Weight Consolidation incorporates the Hessian matrix into parameter regularization to reduce forgetting [16]. Replay-based methods sample original training examples to aid memory [20]. The technique of knowledge distillation aligns the predictions of a fine-tuned model with those of the model before fine-tuning [3]. PEFT also mitigates forgetting. For instance, LoRA [13] uses low-rank matrices for weight modifications while keeping pre-trained parameters frozen, achieving performance comparable to full fine-tuning. However, these solutions focus on sequential inter-task transfer learning. Our focus shifts to intra-task knowledge forgetting, where integrating new knowledge leads to the potential loss of previously existing knowledge.

3 PROPOSED FRAMEWORK - INFUSERKI

3.1 Overview

The objective of our method is to leverage domain knowledge from KGs to enhance LLMs for knowledge-intensive tasks. Specifically, given an LLM $p_\theta \in \mathbb{P}$ and a set of knowledge triplets $\mathcal{T} \in \mathbb{T}$, our goal is to fine-tune the LLM p_θ into p'_θ, incorporating previously unknown knowledge $\mathcal{T}_{unk}$ without affecting existing knowledge $\mathcal{T}_{known}$. For efficiency, we only inject knowledge that is unknown to the LLM as:

$$\mathbb{F}_{\text{KI}} : \mathbb{P} \times \mathbb{T} \rightarrow \mathbb{P} \qquad p'_\theta = f_{\text{KI}}(p_\theta, \mathcal{T}_{unk})$$

The core design of our InfuserKI framework comprises two steps: knowledge detection and knowledge integration, as illustrated in Fig. 1. To be specific, we first detect previously unknown knowledge by feeding questions derived from knowledge triplets to the LLMs. Upon identifying a set of unknown knowledge, we employ the knowledge adapter, which is parallel to the original transformer layer and trained to store new knowledge. The core of our framework, the *knowledge Infuser*, is designed to strategically determine whether new knowledge from the knowledge adapter should be engaged. Throughout this process, we only fine-tune the knowledge adapter and the Infuser while keeping the original transformer parameters fixed.

3.2 Knowledge Detection

Given the inefficiency of fine-tuning LLMs on entire graphs, we aim to identify and integrate only the LLMs' unknown knowledge. To overcome the difficulty of evaluating open-ended questions, we convert triplets into multiple-choice questions [22], allowing for a precise assessment of LLMs' initial unknown knowledge ($\mathcal{N}_3 + \mathcal{N}_4$ in Fig. 2). This strategy enables efficient knowledge integration, using multiple-choice training data to enhance domain-specific performance.

Multiple-choice Question Generation. Given a knowledge triplet, it is transformed into multiple-choice questions using relation templates generated by GPT-4. For instance, the triplet <*Sutura cranii, has finding site, Acrocephalosyndactyly type 5*> is rephrased into the question with golden answer as *"What diagnosis is associated*

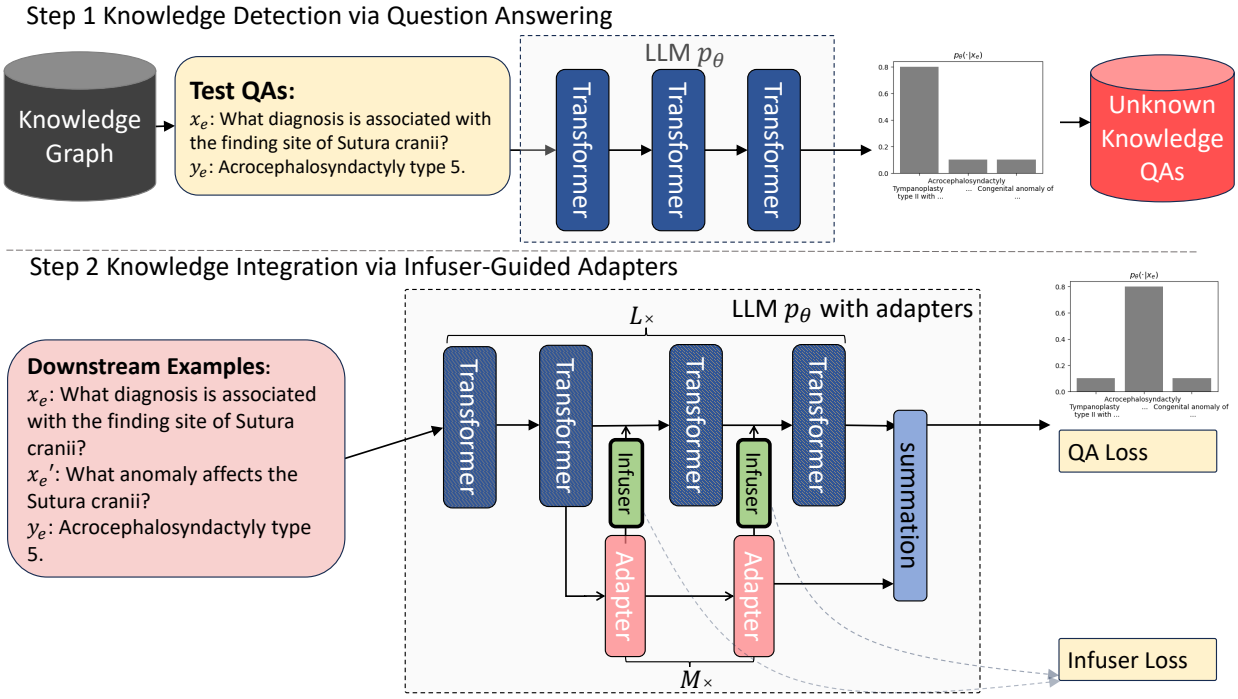

Figure 1: Infuser-Guided Knowledge Integration Framework.

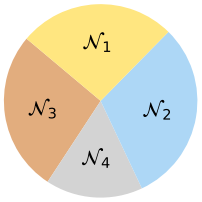

Figure 2: Knowledge Areas in LLMs: Original ($\mathcal{N}_1$+$\mathcal{N}_2$), Post-Fine-Tuning ($\mathcal{N}_1$+$\mathcal{N}_3$), Forgotten ($\mathcal{N}_2$), and Failed Integration ($\mathcal{N}_4$).

with the finding site of Sutura cranii? Answer: Acrocephalosyndactyly type 5." The prompt for generating templates and knowledge evaluation methods are detailed in Appendix 6.1.

Unknown Knowledge Detection. With multiple-choice questions, we input them into LLMs. The testing prompts are in Table 3 in Appendix. We use regular expressions to extract the chosen options from the output of LLMs, treating the response as incorrect if no options can be extracted. This helps us detect the LLMs' known and unknown knowledge. As shown in Fig. 2, the regions labeled $\mathcal{N}_1$ and $\mathcal{N}_2$ represent the set of known knowledge, denoted as $\mathcal{T}_{known}$, while the regions labeled $\mathcal{N}_3$ and $\mathcal{N}_4$ represent the set of unknown knowledge, as $\mathcal{T}_{unk}$. We then develop a new method to integrate this unknown knowledge into the LLMs without affecting existing knowledge.

3.3 Infuser-Guided Knowledge Integration

Next, we detail our Infuser-guided Knowledge Integration method that effectively and efficiently injects unknown knowledge of LLMs.

Knowledge Adapter. To improve parameter efficiency, we use parallel adapters as extra modules to learn new knowledge, keeping the original LLM parameters unchanged, as shown in Fig. 3. Existing works [4, 9] show that Feed-Forward Network (FFN) layers in transformer-based language models store knowledge effectively. Thus, we add adapters parallel to the last M FFN layers for the entire L layers. For the l-th selected adapter layer where $l \in [L-M+1, L]$, we combine the FFN input $\mathbf{H}_P^l \in \mathbb{R}^{n\times d}$ with the output $\mathbf{H}_A^{l-1}$ from the previous adapter layer as:

$$\widetilde{\mathbf{H}}_A^l = \mathbf{H}_A^{l-1} + \mathbf{H}_P^l \tag{1}$$

where n is the length of the LLM input sequence, and d is the hidden dimension. The initial $\mathbf{H}_A^{L-M}$ is set to a vector of all zeros. Following [11], the adapter layer utilizes a down-projection with $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d\times d'}$ to transform the combined input $\widetilde{\mathbf{H}}_A^l$ into a lower-dimensional space specified by the bottleneck dimension d' so as to facilitate the learning of new patterns with minimal extra space. This is followed by a nonlinear activation function σ, and subsequently, an up-projection is applied with $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d'\times d}$ as:

$$\mathbf{H}_A^l = \sigma(\widetilde{\mathbf{H}}_A^l \mathbf{W}_{\text{down}})\mathbf{W}_{\text{up}} \tag{2}$$

Typically, the adapter output directly merges with the original output from the FFN as follows:

$$\mathbf{H}_O^l = \mathbf{H}_A^l + \text{FFN}(\mathbf{H}_P^l) \tag{3}$$

$\mathbf{H}_O^l$ is then fed into either the next transformer attention layer or the final linear and softmax layer. However, this approach can *overload the LLM with unnecessary information about knowledge it already knows*, causing the forgetting issue.

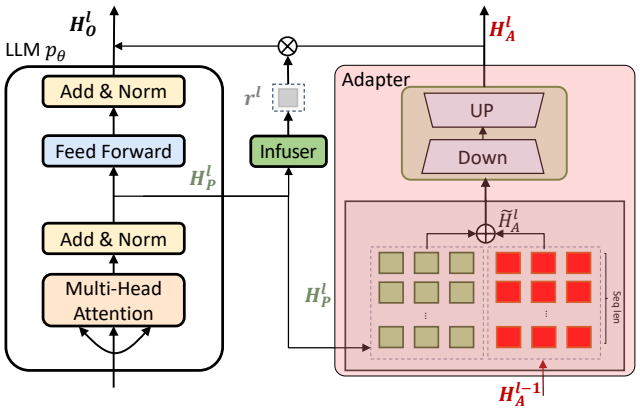

Figure 3: Infuser-Guided Knowledge Adapters.

Knowledge Infuser. To ensure that these extra modules do not confuse the LLM about its existing knowledge, we propose an Infuser model to more effectively infuse the knowledge from the knowledge adapter to the LLM. Intuitively, for a given question, the Infuser assesses if the LLM knows the knowledge at hand. If not, the Infuser can fuse more knowledge from $\mathbf{H}_A^l$ to LLM to provide extra information. If the LLM already knows, $\mathbf{H}_A^l$ should have less impact. Recent work [1] indicates that checking the LLM's internal states can determine if it knows the current question, which paves us a way to design the Infuser. Specifically, we derive an infusing score from the input of an FFN sublayer as follows:

$$r^l = f_{In}(\text{Mean}(\mathbf{H}_P^l)) \tag{4}$$

where f_{In} denotes the Infuser module implemented as a multilayer perceptron (MLP) with a sigmoid activation function and the Mean function averages the vector along the sequence length. This allows infusing score r^l to be mapped to the range $[0, 1]$, indicating how well the LLMs know about the knowledge based on their intermediate states in the l-th FFN layer ($\mathbf{H}_P^l$). As a result, the infusing mechanism helps LLMs learn new knowledge without forgetting what they already know. However, it is difficult for the Infuser to recognize existing knowledge if it only encounters new knowledge during fine-tuning. To fix this, we also include a modest quantity of samples representing knowledge the LLMs already have. Before fine-tuning, we first pre-train the Infuser on a binary infusing task with a balanced mix of known and unknown samples. The Infuser loss is a binary cross-entropy loss function as:

$$\mathcal{L}_{In} = \mathbb{E}_{x, y_{In}} \left[\text{BCE}(f_{In}(\mathbf{H}_P^l), y_{In}) \right] \tag{5}$$

where x is the sample and the infusing label y_{In} is 1 for new knowledge and 0 for previously acquired knowledge. Finally, we obtain an additive filtered adapter vector, which is integrated with the original FFN output:

$$\mathbf{H}_O^l = r^l \mathbf{H}_A^l + \text{FFN}(\mathbf{H}_P^l), \tag{6}$$

which can selectively incorporate knowledge from the adapter into the fixed base model.

Objective Function of InfuserKI. We employ unknown knowledge identified during the knowledge detection phase to fine-tune both the knowledge adapter and the Infuser. The InfuserKI framework is divided into two phases: Infuser tuning and QA (Question Answering) training as illustrated by the following objective function:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{In}, & \text{Infuser Tuning} \\ \mathcal{L}_{QA}, & \text{QA Training} \end{cases} \tag{7}$$

In terms of QA training, we use question-based instructions with standard answers as golden responses. The QA loss is akin to the conventional training loss used in transformer-based language models, tailored to adapt instructions within a specific domain:

$$\mathcal{L}_{QA} = \mathbb{E}_{x,y} \left[\frac{1}{|y|} \sum_{i=1}^{|y|} \text{CE}(p_\theta(\cdot|x, y_{1,\ldots,i-1}), y_i) \right] \tag{8}$$

where $\text{CE}(\cdot, \cdot)$ denotes the cross-entropy loss function, $y = y_1, \ldots,$ is the golden output, and $p_\theta(\cdot|x, y_{1,\ldots,\cdot,i-1})$ is the prediction of an LLM. Note that we also incorporate a small set of yes/no QA samples to enhance the model generality to various question types.

To be specific, given an LLM p_θ and a KG with knowledge triplets $<h, r, t>$, we generate question-based instructions q and standard answers y. The training is divided into two stages. Initially, we tune the Infuser using a small set of balanced samples of known and unknown, as per Eq. 5. In the second stage, we fine-tune the model using a QA loss to integrate unknown knowledge, following Eq. 8.

4 EXPERIMENTS

4.1 Experimental Setup

We evaluate our InfuserKI framework with competitive baselines on a domain UMLS knowledge graph and the corresponding downstream task in terms of reliability, locality, and generality.

Datasets. We conduct experiments on a medical KG **UMLS** [2] with **PubMedQA** [15] as the downstream task.

Metrics. Following [14] (see Appendix 6.3), as shown in Fig. 2 with areas for various knowledge dynamics, we use the following metrics: (1) **Newly-learned Rate (NR)** for reliability, calculated by $NR = \mathbb{E}_{x \in \mathcal{N}_3 + \mathcal{N}_4} [p_{known}(x)]$ with $p_{known}(x) = 1$ for correct answers and 0 for incorrect ones; (2) **Remembering Rate (RR)** for locality, defined as $RR = \mathbb{E}_{x \in \mathcal{N}_1 + \mathcal{N}_2} [p_{known}(x)]$; (3) **F1_T1 and F1_T2** for seen templates to assess reliability and locality and **F1_T3 to F1_T5** for unseen templates, with their average, denoted as **F1_Unseen**, serving to assess generality; and (4) **Downstream-Task F1** for the effectiveness of knowledge integration on downstream tasks.

Baselines. We compare InfuserKI against both PEFT methods and ME techniques. The **PEFT** baselines include: (i) **Prefix Tuning** [19] employs learnable prompts in input or intermediate layers; (ii) **LoRA** [13] uses trainable low-rank matrices for self-attention weights while freezing other parameters; (iii) **QLoRA** [5] quantizes pre-trained models to 4 bits based on LoRA.

All PEFT methods are tested with the same mix of unknown and known samples to ensure fairness. The adopted **Knowledge Model Editing Methods** are: (i) **CALINET** [6] corrects false knowledge by fine-tuning an adapter in a specific FFN layer while keeping original model parameters intact; (ii) **T-Patcher** [14] adds a few trainable neurons to the last FFN layer for error correction.

	Reliability	Locality			Generality				
Methods	**NR**	**RR**	**F1_T1**	**F1_T2**	**F1_T3**	**F1_T4**	**F1_T5**	**F1_Unseen**	**PubMedQA**
LLaMa-2-7B	-	-	0.41	0.53	0.42	0.50	0.39	0.44	0.38
CALINET	**1.00**	0.52	0.81	0.75	0.50	0.68	0.46	0.55	0.46
T-Patcher	0.73	0.06	0.45	0.71	0.30	0.65	0.32	0.42	0.40
Prefix Tuning	0.70	0.90	0.78	0.71	0.63	0.54	0.60	0.59	0.44
LoRA	0.92	0.80	0.87	0.74	0.82	0.72	0.78	0.77	0.47
QLoRA	0.97	0.88	0.93	0.78	0.79	0.64	0.81	0.75	0.49
Ours	0.99	**0.99**	**0.99**	**0.89**	**0.91**	**0.82**	**0.92**	**0.88**	**0.58**

Table 1: Comparative results of InfuserKI with PEFT and ME methods on the UMLS 2.5k triplets.

Experimental Details. We use LLaMa-2-7B [34] as our base LLM. Following MoP [27], we sample parts of the KG (2, 500 triplets for UMLS) in our experiments. During fine-tuning, we set the dimensionality d' to 10 and positioned the adapters in the last 30 layers out of 32. Our approach adds approximately 2.5M extra parameters. Using the AdamW optimizer [21] with a batch size of 8 and a learning rate of $1 \times e^{-4}$, training takes about 30 minutes per epoch for UMLS 2.5k on 4×A100 GPU servers. The PEFT baselines are implemented following LLaMa-Adapter [41] and PEFT [23].

4.2 Results and Analysis

Table 1 show a comparison of our InfuserKI against existing PEFT and ME methods on the UMLS with 2, 500 triplets. We can observe: (1) The performance of Vanilla LLaMa-2-7B underscores a lack of domain-specific knowledge, highlighting its knowledge limitations in specialized domains. (2) Our method outperforms ME baselines such as CALINET and T-Patcher, which focus on correcting existing knowledge by positioning adapters in earlier transformer layers. This emphasis makes them less suited for integrating new knowledge compared to our approach. (3) Compared to PEFT methods such as Prefix Tuning, LoRA, and QLoRA, our method achieves superior locality (RR). This improvement stems from our infusing mechanism's adaptive selection of supplementary information, which effectively prevents adapters from interfering with previously acquired knowledge. (4) Our method outperforms the T-Patcher across all metrics. Although T-Patcher reduces the impact on a minimal number of unrelated samples, it lacks robustness in locality, which our infusing mechanism effectively addresses.

4.3 Infuser Analysis

To delve deeper into the infusing mechanism, we visualize its values on the test set. As shown in Fig. 4, we display the infusing scores for both original known and unknown samples. Our observation is that infusing scores are lower on known samples, helping to block interfering information and thus mitigating knowledge forgetting.

4.4 Case Study

To intuitively understand the effectiveness of our framework, we compare the prediction score distributions over candidate choices from the vanilla LLaMa-2, LoRA, and our InfuserKI in two cases. Fig. 5 (a) shows that LLaMa-2, which initially gives incorrect answers, can provide correct answers after applying our InfuserKI and

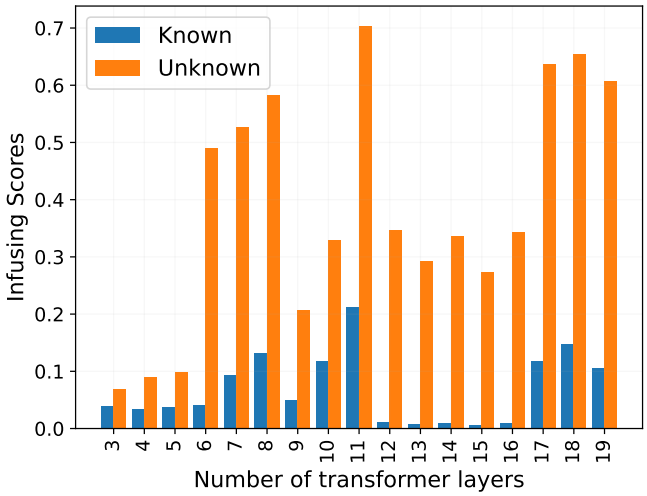

Figure 4: Infusing Scores for Known vs. Unknown Samples.

What diagnosis is associated with the finding site of Sutura cranii?
(A) Discharging mastoid cavity (finding)
(B) Congenital anomaly of basioccipital bone (disorder)
(C) Swelling over mastoid (finding)
(D) Overlapping cranial sutures (finding)

LLaMa:		LoRA:		InfuserKI:	
A 3e-7	**B 0.999**	A 3e-5	B 2e-5	A 1e-6	B 6e-7
C 3e-7	*D 3e-7	C 2e-5	***D 0.999**	C 3e-6	***D 0.999**

(a) LoRA and InfuserKI successfully inject knowledge into LLaMa.

What procedure is performed on the Process Mastoideus?
(A) Epithelial debris of mastoid cavity (finding)
(B) Congenital anomaly of basioccipital bone (disorder)
(C) Repair of mastoid antrum or cavity (procedure)
(D) Finding of moistness of mastoid cavity (finding)

LLaMa:		LoRA:		InfuserKI:	
A 5e-5	B 1e-4	A 0.02	B 1e-5	A 3e-6	B 1e-7
***C 0.999**	D 1e-4	*C 0.007	**D 0.97**	***C 0.998**	D 0.001

(b) LoRA forgets knowledge LLaMa knows and InfuserKI remembers.

Figure 5: Illustration of Infuser-Guided Knowledge Integration with less forgetting.

LoRA. However, LoRA induces forgetting for the second case, as depicted in Fig. 5 (b) while InfuserKI retains the knowledge.

5 CONCLUSION

In this study, we tackle a novel problem of integrating new knowledge from KGs into LLMs without affecting existing knowledge. We introduce the Infuser-guided Knowledge Integration framework, designed to selectively add new information to LLMs, minimizing the impact on prior knowledge and preventing catastrophic forgetting.

6 APPENDIX

I need five question-answer templates to analyze relationships in triplets formatted as <SUBJECT, RELATION, OBJECT>, focusing on the relation {RELATION}. Answers should be either the [OBJECT] entity or a yes/no response. Use placeholders [SUBJECT] and [OBJECT] to denote where the subject and object entities will be inserted. Context is provided by the following examples: {EXAMPLE TRIPLETS} Please create five unique question-answer templates, formatted as a JSON string. For clarity, the output should follow this format: { 'rel': { RELATION }, 'template#1': '[Question-answer template 1]', 'template#2': '[Question-answer template 2]', 'template#3': '[Question-answer template 3]', 'template#4': '[Question-answer template 4]', 'template#5': '[Question-answer template 5]', 'memo': '[Additional memo or notes]' } Note: ONLY OUTPUT A JSON STRING, NO ANY OTHER CONTENT. Output: <Your generated JSON string>

Table 2: Prompt to GPT-4 to generate QA templates.

Below is an instruction that describes a task. Write a response that appropriately completes the request. ### Instruction: {instruction} ### Response:

Table 3: Prompt to LLMs to answer MCQA.

6.1 Template Prompts and MCQA Construction

To facilitate an effective comparison between long-form answers from LLMs and standard answers for open-ended questions, we utilize a multiple-choice format, as detailed in Table 2. This format comprises a correct answer alongside three distractors. The first distractor is chosen for its minimal edit distance to the head entity, while the remaining two are randomly selected from a set of ten candidates based on their edit distance to the correct answer. Subsequently, these choices are randomized and presented as options (A), (B), (C), and (D) alongside the question, allowing for a precise assessment of LLMs' knowledge in specific domains.

6.2 Knowledge Graphs and Datasets

UMLS [2]: The Unified Medical Language System (UMLS) knowledge graph, developed by the US National Library of Medicine, integrates over 2 million terms for nearly 900,000 concepts from more than 60 biomedical vocabularies. These include the NCBI taxonomy, Gene Ontology, and Medical Subject Headings (MeSH), along with 12 million concept relations. For testing, we employ the PubMedQA dataset [15], a biomedical QA dataset derived from PubMed abstracts, featuring Yes/No/Maybe questions alongside context, as highlighted in [37].

6.3 Three Evaluation Properties

Following [14], the enhanced LLM should meet these properties:

Property 1, Reliability: The enhanced model p'_θ incorporates knowledge previously unknown to p_θ as

$$p'_\theta(x) = y \text{ if } p_\theta(x) \neq y \ . \tag{9}$$

Reliability is quantified using the Newly-learned Rate (NR) in our work.

Property 2, Locality: Knowledge integration should be localized and precise, ensuring the fine-tuned model p'_θ retains accuracy on $\mathcal{T}_{known}$, the knowledge previously known to p_θ as

$$p'_\theta(x) = y \text{ if } p_\theta(x) = y \ . \tag{10}$$

Here, this property is measured by the Remembering Rate (RR), which indicates the accuracy of the previously acquired knowledge.

Property 3, Generality: For any unknown sample x, let $\mathbb{E}_x = \{x' | y_{x'} = y_x\}$ denote a set of equivalent inputs. The model p'_θ should correctly answer all instances $x' \in \mathbb{E}_x$ as

$$\forall x' \in \mathbb{E}_x, p'_\theta(x') = y \ . \tag{11}$$

In this study, generality is assessed by averaging F1 scores (F1_Unseen) across three unseen templates during training as well as performance on downstream tasks.

REFERENCES

[1] Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734* (2023).

[2] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270.

[3] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems* 33 (2020), 15920–15930.

[4] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge Neurons in Pretrained Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 8493–8502.

[5] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314* (2023).

[6] Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating Factual Knowledge in Pretrained Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. 5937–5947.

[7] Denis Emelin, Daniele Bonadiman, Sawsan Alqahtani, Yi Zhang, and Saab Mansour. 2022. Injecting Domain Knowledge in Language Models for Task-oriented Dialogue Systems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 11962–11974.

[8] Miao Fan, Qiang Zhou, Emily Chang, and Fang Zheng. 2014. Transition-based knowledge graph embedding with relational mapping properties. In *Proceedings of the 28th Pacific Asia conference on language, information and computing*. 328–337.

[9] 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*. 5484–5495.

[10] Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors. In *NeurIPS Workshop on Robustness in Sequence Modeling*.

[11] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a Unified View of Parameter-Efficient Transfer Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=0RDcd5Axok

[12] 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 *International Conference on Machine Learning*. PMLR, 2790–2799.

[13] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[14] Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-Patcher: One Mistake Worth One Neuron. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=4oYUGeGBPm

[15] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. 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)*. 2567–2577.

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[17] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466.

[18] Shaobo Li, Xiaoguang Li, Lifeng Shang, Zhenhua Dong, Chengjie Sun, Bingquan Liu, Zhenzhou Ji, Xin Jiang, and Qun Liu. 2022. How Pre-trained Language Models Capture Factual Knowledge? A Causal-Inspired Analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1720–1732. https://doi.org/10.18653/v1/2022.findings-acl.136

[19] 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 (Volume 1: Long Papers)*. 4582–4597.

[20] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems* 30 (2017).

[21] Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[22] Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896* (2023).

[23] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

[24] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Vol. 24. Elsevier, 109–165.

[25] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* 35 (2022), 17359–17372.

[26] Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-Editing Memory in a Transformer. In *The Eleventh International Conference on Learning Representations*.

[27] Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. 2021. Mixture-of-Partitions: Infusing Large Biomedical Knowledge Graphs into BERT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 4672–4681.

[28] Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: Improving Entity and Relation Understanding for Pre-trained Language Models via Contrastive Learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 3350–3363. https://doi.org/10.18653/v1/2021.acl-long.260

[29] Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review* 97, 2 (1990), 285.

[30] Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2017. Knowledge questions from knowledge graphs. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*. 11–18.

[31] Rohit Sridhar and Diyi Yang. 2022. Explaining toxic text via knowledge enhanced text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 811–826.

[32] Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2022. JointLK: Joint Reasoning with Language Models and Knowledge Graphs for Commonsense Question Answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5049–5060.

[33] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223* (2019).

[34] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[35] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[36] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuan-Jing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 1405–1418.

[37] Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Further finetuning llama on medical papers. *arXiv preprint arXiv:2304.14454* (2023).

[38] LI Xuhong, Yves Grandvalet, and Franck Davoine. 2018. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*. PMLR, 2825–2834.

[39] Wenhao Yu, Chenguang Zhu, Lianhui Qin, Zhihan Zhang, Tong Zhao, and Meng Jiang. 2022. Diversifying Content Generation for Commonsense Reasoning with Mixture of Knowledge Graph Experts. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*. 1–11.

[40] Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2024. Investigating the Catastrophic Forgetting in Multimodal Large Language Model Fine-Tuning. In *Conference on Parsimony and Learning*. PMLR, 202–227.

[41] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199* (2023).

[42] Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2022. DKPLM: decomposable knowledge-enhanced pre-trained language model for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11703–11711.

[43] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2021. Greaselm: Graph reasoning enhanced language models. In *International conference on learning representations*.

[44] Ziwang Zhao, Linmei Hu, Hanyu Zhao, Yingxia Shao, and Yequan Wang. 2023. Knowledgeable Parameter Efficient Tuning Network for Commonsense Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 9051–9063.