

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

Anonymous ACL submission

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-2.5k and MetaQA domain knowledge graphs demonstrate that InfuserKI can effectively acquire new knowledge and outperform state-of-the-art baselines by 9% and 6%, respectively, in reducing knowledge forgetting.

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 (Touvron et al., 2023a,b). However, LLMs can generate misleading or inaccurate texts, especially in knowledge-intensive tasks such as open-domain QA (Kwiatkowski et al., 2019), due to lack of domain knowledge and catastrophic forgetting after fine-tuning (Zhai et al., 2024; Li et al., 2022). 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 (Wu et al., 2023), creating triplet-based pre-training tasks (Zhang et al., 2022; Qin et al., 2021; Wang et al., 2021), employing KGs as external sources for retrieval (Sridhar and Yang, 2022; Yu et al., 2022), and directly using parameter-efficient fine-tuning (PEFT) methods such as LoRA (Hu et al., 2021) and adapters (Houlsby et al., 2019) or model editing (ME) methods such as T-Patcher (Huang et al., 2023) to inject knowledge in a triplet-to-text way (Meng et al., 2021; Emelin et al., 2022; Dong et al., 2022). 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. Fig. 1 (a) and (b) show an illustrative comparison between a vanilla LLM and the fine-tuned version, revealing direct fine-tuning leads to forgetting previously known data, whereas Fig. 1 (c) depicts an expected outcome of seamlessly injecting new knowledge without losing existing information. 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

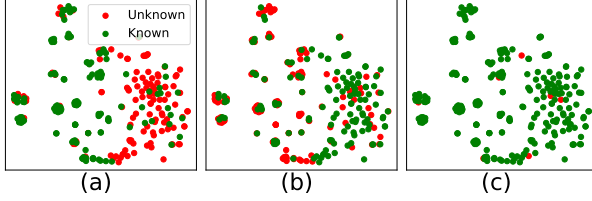


Figure 1: An illustrative comparison among (a) Vanilla LLM, (b) Fine-Tuned LLM, and (c) our Knowledge-Infused LLM.

by (Azaria and Mitchell, 2023), 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 while preserving the original model parameters. To inject new knowledge only, the InfuserKI framework begins by detecting knowledge unknown to LLMs. Following (Zhao et al., 2023; Seyler et al., 2017), we then generate a *knowledge statement* and *multiple-choice questions* for a knowledge triplet $\langle h, r, t \rangle$ using predefined relational templates, with an example in Fig. 3. To enhance the generality of the newly integrated knowledge, InfuserKI introduces a relation classification task aimed at improving the linguistic representations learned by the adapters, predicting relations within knowledge statements based on the adapter outputs on head and tail entities. This not only solidifies the integration of new knowledge but also helps the robustness of generalizing it to unseen situations. Our main contributions are:

1. We study a novel problem of effectively integrating unknown knowledge from KGs into LLMs without affecting existing knowledge.
2. We propose a new knowledge integration framework InfuserKI, which enables the adaptive selection of known and unknown knowledge to be integrated into LLM, effectively mitigating knowledge forgetting.
3. Extensive evaluations on UMLS and MetaQA datasets 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 (Sun et al., 2019) injects KG’s embeddings, such as TransE (Fan et al., 2014), into models using an entity-token alignment masking loss. However, retraining is time-consuming. In fine-tuning, methods including JointLK (Sun et al., 2022) and GreaseLM (Zhang et al., 2021) apply graph neural networks to model knowledge subgraphs, relying on KGs until inference. Fully fine-tuning models such as PMC-LLaMa (Wu et al., 2023) is computationally costly; therefore PEFT methods, for instance, LoRA (Hu et al., 2021) and Adapters (Houlsby et al., 2019), are more feasible. Based on these works, MoP (Meng et al., 2021), K-Adapter (Wang et al., 2021), and KB-adapters (Emelin et al., 2022) inject knowledge directly into model parameters but risk catastrophic forgetting of unrelated knowledge (Meng et al., 2022b). 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. (2022), modify specific weights related to knowledge edits. ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) 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 (Dong et al., 2022) and T-Patcher (Huang et al., 2023) incorporate memory slots or trainable "patches" into final FFN outputs. GRACE (Hartvigsen et al., 2023) 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 (Ratcliff, 1990). This becomes particularly evident in sequential inter-task learning, where acquiring new task knowledge leads to the forgetting of earlier task knowledge (McCloskey and Cohen, 1989). To tackle this, various methods are developed. Xuhong et al. (2018) applies regularization constraints to minimize parameter changes when learning new tasks. Elastic Weight Consolidation incorporates the Hessian matrix into parameter regularization to reduce forgetting (Kirkpatrick et al., 2017). Replay-based methods sample original training examples to aid memory (Lopez-Paz and Ranzato, 2017). The technique of knowledge distillation aligns the predictions of a fine-tuned model with those of the model before fine-tuning (Buzzega et al., 2020). PEFT also mitigates forgetting. For instance, LoRA (Hu et al., 2021) 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:

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

The core design of our InfuserKI framework comprises two steps: knowledge detection and knowledge integration, as illustrated in Fig. 3. 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 (Manakul et al., 2023), 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 and a knowledge statement 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 with the finding site of Sutura cranii? Answer: Acrocephalosyndactyly type 5," along with a knowledge statement as "The finding site for Sutura cranii is associated with Acrocephalosyndactyly type 5." The prompt for generating templates and knowledge evaluation method are detailed in Appendix A.1.

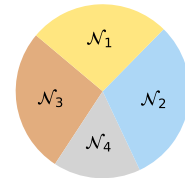
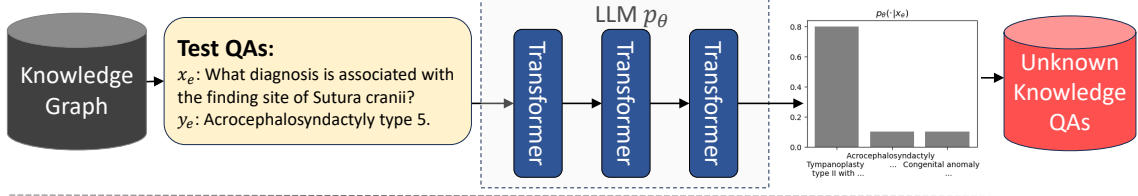


Figure 2: Knowledge Areas in LLMs: Original ($\mathcal{N}_1 + \mathcal{N}_2$), Post-Fine-Tuning ($\mathcal{N}_1 + \mathcal{N}_3$), Forgotten (\mathcal{N}_2), and Failed Integration (\mathcal{N}_4).

Unknown Knowledge Detection With multiple-choice questions, we input them into LLMs. The testing prompts are in Table 6 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

Step 1 Knowledge Detection via Question Answering



Step 2 Knowledge Integration via Infuser-Guided Adapters

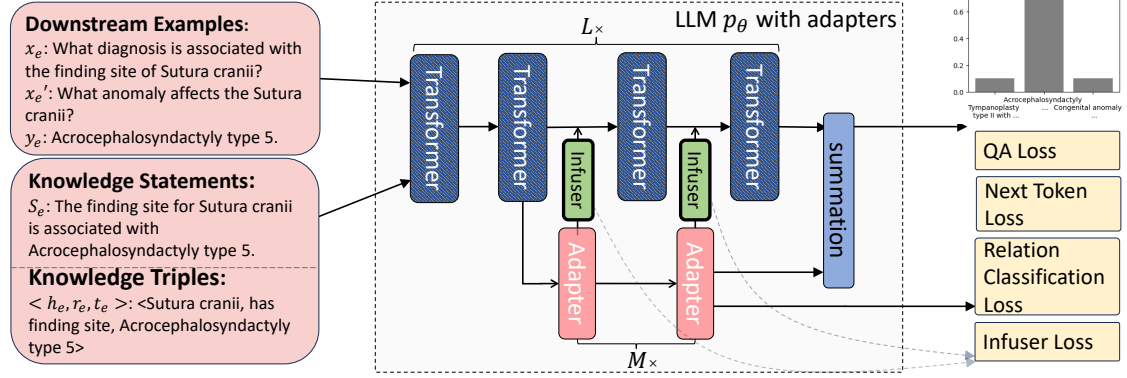


Figure 3: Infuser-Guided Knowledge Integration Framework.

and \mathcal{N}_4 represent the set of unknown knowledge, as \mathcal{T}_{unk} . We then develop a new method to integrate these 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. 4. Existing works (Dai et al., 2022; Geva et al., 2021) 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:

$$\tilde{\mathbf{H}}_A^l = \mathbf{H}_A^{l-1} + \mathbf{H}_P^l \quad (1)$$

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 (He et al., 2022), the adapter layer utilizes a down-projection with $\mathbf{W}_{down} \in \mathbb{R}^{d \times d'}$ to transform the combined input $\tilde{\mathbf{H}}_A^l$ 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 $\mathbf{W}_{up} \in \mathbb{R}^{d' \times d}$ as:

$$\mathbf{H}_A^l = \sigma(\tilde{\mathbf{H}}_A^l \mathbf{W}_{down}) \mathbf{W}_{up} \quad (2)$$

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

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

\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.

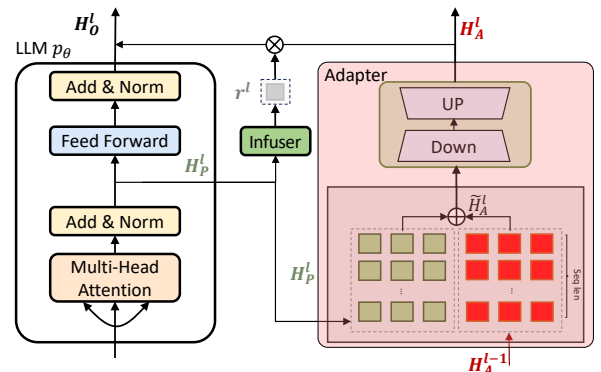


Figure 4: Infuser-Guided Knowledge Adapters.

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 (Azaria and Mitchell, 2023) 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:

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

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:

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

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:

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

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 three phases: Infuser tuning, QA (Question Answering) training, and RC (Relation Classification) training, as illustrated by the

following objective function:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{In}, & \text{Infuser Tuning} \\ \mathcal{L}_{QA}, & \text{QA Training} \\ \mathcal{L}_{NTL} + \lambda_{RC} \mathcal{L}_{RC}, & \text{RC Training.} \end{cases} \quad (7)$$

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:

$$\mathcal{L}_{QA} = \mathbb{E}_{x, y} \left[\frac{1}{|y|} \sum_{i=1}^{|y|} \text{CE}(p_\theta(\cdot | x, y_1, \dots, y_{i-1}), y_i) \right] \quad (8)$$

where $\text{CE}(\cdot, \cdot)$ denotes the cross-entropy loss function, $y = y_1, \dots$, is the golden output, and $p_\theta(\cdot | x, y_1, \dots, y_{i-1})$ 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 boost generality of InfuserKI, we adopt a relation classification task, following Zhao et al. (2023), to enhance our knowledge adapters’ understanding of relational facts. For a given knowledge statement k and its triplet $\langle h, r, t \rangle$, we perform mean pooling on the adapter output \mathbf{H}_A^L for the entity mentions, obtaining representations v^h and v^t . Following Qin et al. (2021), we form a relational representation $v^r = [v^h, v^t]$, treating r as a positive sample and other relations as negatives. The relation classification (RC) loss, employing the InfoNCE loss (Oord et al., 2018), aims to distinguish positive relations from negatives, as shown below:

$$\mathcal{L}_{RC} = \mathbb{E}_k \left[-\log \frac{\exp(f_1^R(v^r) \cdot f_2^R(r)/\tau)}{\sum_{r' \in \mathcal{E}} \exp(f_1^R(v^r) \cdot f_2^R(r')/\tau)} \right] \quad (9)$$

where τ acts as a temperature hyperparameter. The functions f_1^R and f_2^R align entity and relation embeddings into a unified dimensional space, respectively, with \mathcal{E} denoting the complete set of relations. Besides that, we also adopt the conventional training loss (i.e. next token loss) used in transformer models:

$$\mathcal{L}_{NTL} = \mathbb{E}_k \left[\frac{1}{|k|} \sum_{i=1}^{|k|} \text{CE}(P_\theta(k_i | k_{1, \dots, i-1})) \right] \quad (10)$$

The training algorithm is detailed in Appendix A.2. To be specific, given an LLM p_θ and a KG with knowledge triplets $\langle h, r, t \rangle$, we generate question-based instructions q , standard answers

y , and knowledge statements k . The training is divided into three 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. In the final stage, we use knowledge statements and triplets to enhance the model generality, according to Eq. 9 and 10.

4 Experiments

4.1 Experimental Setup

We evaluate our InfuserKI framework with competitive baselines on two domain KGs and their corresponding downstream tasks in terms of reliability, locality, and generality.

Datasets We conduct experiments on a medical KG UMLS (Bodenreider, 2004) with PubMedQA (Jin et al., 2019) and a movie KG MetaQA (Zhang et al., 2018) with MetaQA-1HopQA as the downstream task respectively. The detailed description is in Appendix A.3.

Metrics Following (Huang et al., 2023) (see Appendix A.4), 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** (Li and Liang, 2021) employs learnable prompts in input or intermediate layers; (ii) **LoRA** (Hu et al., 2021) uses trainable low-rank matrices for self-attention weights while freezing other parameters; (iii) **QLoRA** (Dettmers et al., 2023) 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** (Dong et al., 2022) corrects false knowledge by fine-tuning an adapter in a specific FFN

layer while keeping original model parameters intact; (ii) **T-Patcher** (Huang et al., 2023) adds a few trainable neurons to the last FFN layer for error correction.

Experimental Details We use LLaMa-2-7B (Touvron et al., 2023a) as our base LLM. Following MoP (Meng et al., 2021), we sample parts of the KG (2,500 and 25,000 triplets for UMLS, and 2,900 for MetaQA) 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. The RC loss temperature is set at $\tau = 0.7$. Our approach adds approximately 2.5M extra parameters. Using the AdamW optimizer (Loshchilov and Hutter, 2018) 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 and MetaQA, and 4 hours for UMLS 25k on $4 \times A100$ GPU servers. We adjust loss weights with $\lambda_{RC} = 10$. The PEFT baselines are implemented following LLaMa-Adapter (Zhang et al., 2023) and PEFT (Mangrulkar et al., 2022).

4.2 Results and Analysis

Table 1 and 2 show a comparison of our InfuserKI against existing PEFT and ME methods on the UMLS and MetaQA with 2,500 and 2,900 triplets respectively. 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. (5) Our approach demonstrates better generality on unseen templates and in the downstream tasks PubMedQA/1-HopQA, benefiting from our well-

	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.

	Reliability	Locality	Generality						
Methods	NR	RR	F1_T1	F1_T2	F1_T3	F1_T4	F1_T5	F1_Unseen	1HopQA
LLaMa-2-7B	-	-	0.57	0.45	0.53	0.42	0.52	0.49	0.47
CALINET	0.97	0.84	0.90	0.74	0.85	0.68	0.85	0.79	0.44
T-Patcher	0.39	0.75	0.60	0.69	0.57	0.62	0.61	0.81	0.36
Prefix Tuning	0.12	0.88	0.56	0.53	0.53	0.51	0.53	0.52	0.45
LoRA	0.90	0.80	0.84	0.79	0.81	0.76	0.82	0.80	0.62
QLoRA	0.93	0.90	0.91	0.82	0.89	0.80	0.90	0.86	0.69
Ours	0.99	0.96	0.97	0.88	0.97	0.86	0.94	0.92	0.67

Table 2: Comparative results of InfuserKI with PEFT and ME methods on the MetaQA KG.

designed relation classification task.

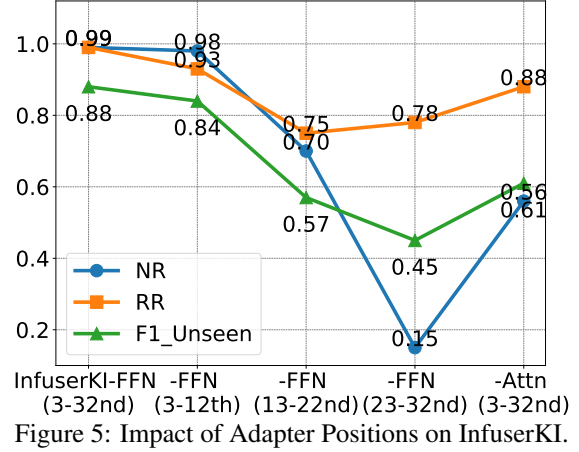
Besides, Table 3 reveals our method maintains excellent performance in reliability, locality, and generality when scaling from 2,500 to 25,000 triplets on the UMLS KG, proving its capability in large-scale knowledge integration. In contrast, traditional ME methods show a performance decline at a larger scale, indicating their limitation to small-scale editing. For additional results on more datasets and with more baselines, please refer to Appendices A.5 and A.6.

4.3 Ablation Study

To assess the impact of each component in InfuserKI, we compare it against variants without certain parts: (1) InfuserKI-w/o-RL, a variant without the Infuser loss; (2) InfuserKI-w/o-Ro, a variant without the Infuser module; (3) InfuserKI-w/o-RC, which excludes the relationship classification task. In Table 4, we notice: (1) Removing Infuser loss diminishes NR by 10%, indicating the role of infusing loss in distinguishing known from unknown information for effective integration. (2) Excluding the Infuser lowers RR by 7%, emphasizing its importance in minimizing knowledge forgetting. (3) Without the relation classification task, F1_Unseen decreases by 5%, showing its effectiveness in leveraging knowledge triplets to generalize new knowledge integration.

4.4 Impact of Adapter Position

To explore the benefits of adapter positions within the transformer architecture, we position adapters



in the 3rd to 12th (bottom), 13th to 22nd (middle), and 23rd to 32nd (top) FFN layers, as well as across the 3rd to 32nd attention layers. Fig. 5 shows that (1) NR diminishes from the bottom to the top layers, indicating that top-layer adapters are less effective for knowledge integration. This could be attributed to the fact that knowledge representations in the upper layers depend on information from the lower layers and any deficiencies in the lower layers can impact the integration of knowledge. This observation aligns with prior studies (Huang et al., 2023; Dong et al., 2022), suggesting that while top layers are better for refining abstract concepts and knowledge correction, bottom layers are more suited for injecting new information; and (2) placing adapters in attention layers proves less effective for new knowledge integration, confirming that FFN layers act as storage for factual knowledge, which also agrees to the findings at (Dai et al., 2022; Geva et al., 2021).

	Reliability	Locality	Generality						
Methods	NR	RR	F1_T1	F1_T2	F1_T3	F1_T4	F1_T5	F1_Unseen	PubMedQA
LLaMa-2-7B	-	-	0.35	0.47	0.36	0.50	0.36	0.41	0.38
CALINET	0.86	0.44	0.69	0.57	0.66	0.55	0.68	0.63	0.45
T-Patcher	0.63	0.20	0.45	0.55	0.38	0.53	0.37	0.43	0.43
Prefix-Tuning	0.82	0.80	0.82	0.59	0.79	0.61	0.77	0.72	0.47
LoRA	0.96	0.90	0.95	0.62	0.94	0.58	0.91	0.81	0.40
QLoRA	0.94	0.91	0.93	0.70	0.90	0.69	0.87	0.82	0.45
Ours	0.99	0.99	0.99	0.83	0.94	0.80	0.96	0.90	0.58

Table 3: Comparative results of InfuserKI with PEFT and ME methods on the UMLS 25k triplets.

Methods	NR	RR	F1_Unseen
InfuserKI	0.99	0.99	0.88
InfuserKI-w/o-RL	0.89	0.97	0.77
InfuserKI-w/o-Ro	0.97	0.92	0.87
InfuserKI-w/o-RC	0.96	0.97	0.83

Table 4: Ablation study on UMLS-2.5k.

4.5 Infuser Analysis

To delve deeper into the infusing mechanism, we visualize its values on the test set. As shown in Fig. 6, 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.

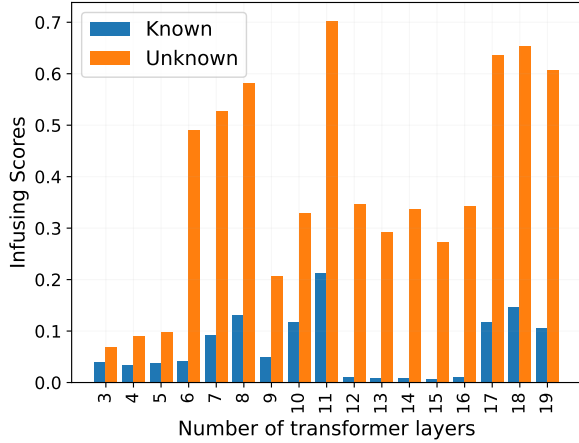


Figure 6: Infusing Scores for Known vs. Unknown Samples.

4.6 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. 7 (a) shows that LLaMa-2, which initially gives incorrect answers, can provide correct answers after applying our InfuserKI and LoRA. However, LoRA induces forgetting for the second

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:
 A 3e-7 B **0.999**
 C 3e-7 *D 3e-7

LoRA:
 A 3e-5 B 2e-5
 C 2e-5 *D **0.999**

InfuserKI:
 A 1e-6 B 6e-7
 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:
 A 5e-5 B 1e-4
 *C **0.999** D 1e-4

LoRA:
 A 0.02 B 1e-5
 *C 0.007 D **0.97**

InfuserKI:
 A 3e-6 B 1e-7
 *C **0.998** D 0.001

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

Figure 7: Illustration of Infuser-Guided Knowledge Integration with less forgetting.

case, as depicted in Fig. 7 (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. A relation classification task further enhances the model’s generality. Evaluations on UMLS and MetaQA demonstrate InfuserKI’s effectiveness in integrating knowledge with less forgetting, maintaining sustained performance with large-scale data, and exhibiting exceptional generality on unseen templates and downstream tasks. As for future work, we will study methods to test and integrate knowledge into LLMs with multi-hop knowledge triplets.

6 Limitations

We note that the effectiveness of our method is contingent upon the base language model’s ability to follow instructions accurately. In scenarios where the underlying model exhibits suboptimal instruction-following capabilities, the integration of knowledge, regardless of its quality, may not significantly improve performance on downstream tasks. Consequently, applying our knowledge integration framework to models with limited instruction-following proficiency presents a considerable challenge.

References

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

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

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*, 33:15920–15930.

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)*, pages 8493–8502.

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

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*, pages 5937–5947.

Hady Elsahar, Pavlos Vougiouklis, Arslan Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

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*, pages 11962–11974.

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*, pages 328–337.

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*, pages 5484–5495.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adapters. In *NeurIPS Workshop on Robustness in Sequence Modeling*.

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*.

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*, pages 2790–2799. PMLR.

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*.

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*.

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)*, pages 2567–2577.

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):3521–3526.

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*, 7:452–466.

693	Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension . In <i>Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)</i> , pages 333–342, Vancouver, Canada. Association for Computational Linguistics.	749
694		750
695		751
696		
697		
698		
699	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 <i>Findings of the Association for Computational Linguistics: ACL 2022</i> , pages 1720–1732, Dublin, Ireland. Association for Computational Linguistics.	752
700		753
701		754
702		755
703		756
704		757
705		758
706		759
707		760
708		761
709	Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In <i>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)</i> , pages 4582–4597.	762
710		763
711		764
712		765
713		
714	David Lopez-Paz and Marc’Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. <i>Advances in neural information processing systems</i> , 30.	766
715		767
716		768
717		769
718	Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In <i>International Conference on Learning Representations</i> .	770
719		
720	Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. <i>arXiv preprint arXiv:2303.08896</i> .	771
721		772
722		773
723		774
724	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 .	775
725		776
726		777
727		778
728		779
729	Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In <i>Psychology of learning and motivation</i> , volume 24, pages 109–165. Elsevier.	780
730		781
731		
732		
733		
734	Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. <i>Advances in Neural Information Processing Systems</i> , 35:17359–17372.	782
735		783
736		784
737		785
738	Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In <i>The Eleventh International Conference on Learning Representations</i> .	786
739		787
740		788
741		
742		
743	Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. 2021. Mixture-of-partitions: Infusing large biomedical knowledge graphs into bert. In <i>Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing</i> , pages 4672–4681.	789
744		790
745		791
746		792
747		793
748		794
	Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. <i>arXiv preprint arXiv:1807.03748</i> .	795
		796
		797
		798
		799
		800
	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 <i>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)</i> , pages 3350–3363, Online. Association for Computational Linguistics.	801
		802
		803
		804
		805
		806
	Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. <i>Psychological review</i> , 97(2):285.	
	Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2017. Knowledge questions from knowledge graphs. In <i>Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval</i> , pages 11–18.	
	Rohit Sridhar and Diyi Yang. 2022. Explaining toxic text via knowledge enhanced text generation. In <i>Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 811–826.	
	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. <i>arXiv preprint arXiv:1904.09223</i> .	
	Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2022. Jointlk: Joint reasoning with language models and knowledge graphs for commonsense question answering. In <i>Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 5049–5060.	
	Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. <i>arXiv preprint arXiv:2302.13971</i> .	
	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. <i>arXiv preprint arXiv:2307.09288</i> .	
	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 <i>Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021</i> , pages 1405–1418.	

Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2023. Pmc-llama: Further fine-tuning llama on medical papers. *arXiv preprint arXiv:2304.14454*.

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

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)*, pages 1–11.

Yuxiang 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*, pages 202–227. PMLR.

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*.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chenguang 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*, volume 36, pages 11703–11711.

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*.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Ziwan 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)*, pages 9051–9063.

A Appendix

A.1 Template Prompts and MCQA Construction

To facilitate an effective comparison between long-form answers from LLMs and standard answers for

I need five question-answer templates and a knowledge statement 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. The knowledge statement should be a VERY brief, declarative sentence illustrating the RELATION between [SUBJECT] and [OBJECT], incorporating the original relation words ‘possibly equivalent to’.

Context is provided by the following examples: {EXAMPLE TRIPLETS}

Please create five unique question-answer templates and one knowledge statement, 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]',
  'knowledge_statement': '[Knowledge statement]',
  'memo': '[Additional memo or notes]' }
```

Note: ONLY OUTPUT A JSON STRING, NO ANY OTHER CONTENT.

Output: <Your generated JSON string>

Table 5: Prompt to GPT-4 to generate QA templates

open-ended questions, we utilize a multiple-choice format, as detailed in Table 5. 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.

A.2 Algorithm

The algorithm is described in Algorithm 1.

A.3 Knowledge Graphs and Datasets

UMLS (Bodenreider, 2004): The Unified Medical Language System (UMLS) knowledge graph, de-

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

Response:

Table 6: Prompt to LLMs to answer MCQA

Algorithm 1 Infuser-Guided Knowledge Integration.

- 1: **procedure** ROUTERKI(p_θ, \mathcal{G}) \triangleright Target LLM p_θ and KG \mathcal{G} with triplets $\langle h, r, t \rangle$
- 2: **# Step 1: Knowledge Detection**
- 3: Convert triplets into MCQs q , with correct answers y and knowledge statements k , using relational templates.
- 4: Input MCQs into p_θ to identify unknown knowledge.
- 5: **# Step 2: Knowledge Integration**
- 6: Tune Infuser on a balanced mix of known and unknown samples as per Eq. 5.
- 7: Fine-tune adapters for templates #1 and #2 using QA loss in Eq. 8.
- 8: Apply relation classification to unknown statements, following Eq. 9 and Eq. 10.

veloped 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 (Jin et al., 2019), a biomedical QA dataset derived from PubMed abstracts, featuring Yes/No/Maybe questions alongside context, as highlighted in (Wu et al., 2023).

MetaQA (Zhang et al., 2018) serves as a multi-hop KGQA benchmark in the movie domain, presenting a knowledge graph with 135,000 triplets, 43,000 entities, and 9 relations. It organizes over 400,000 questions into 1-hop, 2-hop, and 3-hop categories, each annotated with head entities, answers, and reasoning paths. Our analysis concentrates on the 1-hop version for downstream testing.

A.4 Three Evaluation Properties

Following (Huang et al., 2023), the enhanced LLM should meet these properties:

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

p_θ as

$$p'_\theta(x) = y \text{ if } p_\theta(x) \neq y. \quad (11)$$

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

$$p'_\theta(x) = y \text{ if } p_\theta(x) = y. \quad (12)$$

Here, this property is measured by the Remembering Rate (RR), which indicates the accuracy on 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

$$\forall x' \in \mathbb{E}_x, p'_\theta(x') = y. \quad (13)$$

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.

A.5 Results on ME Datasets and YAGO

We conduct experiments on two Wikipedia-sourced datasets used in Model Editing (ME) methods: the Zero-Shot Relation Extraction (zsRE) dataset (Levy et al., 2017) and the T-REx dataset (Elsahar et al., 2018). We also perform comparative experiments using sampled knowledge graphs from YAGO. The results in Table 7, 8, and 9 show that the LLM backbone has deficiencies in handling world knowledge across three datasets, but performance improves with our knowledge injection method, achieving optimal specificity, locality, and generality.

A.6 Comparison with RAG Baselines

Both the Retrieval-Augmented Generation (RAG) method and our approach aim to enhance LLMs using external knowledge, necessitating a comparative analysis using the UMLS dataset. We have designed experiments to inject and assess knowledge specific to certain relation types, developing two RAG variants: RAG-TKS, which uses a BM25 retriever to utilize knowledge statements from the training set for context, and RAG-Google, which retrieves top-ranked content using Google. The results in Table 10 demonstrate that

Methods	NR	RR	F1_T1	F1_T2	F1_T3	F1_T4	F1_T5	F1_Unseen
LLaMa-2-7B	-	-	0.51	0.59	0.48	0.59	0.49	0.52
CALINET	0.61	0.49	0.54	0.66	0.53	0.63	0.49	0.55
LoRA	0.55	0.55	0.55	0.54	0.57	0.52	0.51	0.53
Ours	0.84	0.95	0.91	0.80	0.82	0.65	0.81	0.76

Table 7: Comparative results of InfuserKI with PEFT and ME methods on the zsRE-1k.

Methods	NR	RR	F1_T1	F1_T2	F1_T3	F1_T4	F1_T5	F1_Unseen
LLaMa-2-7B	-	-	0.64	0.62	0.66	0.63	0.62	0.64
CALINET	0.94	0.72	0.80	0.65	0.68	0.62	0.72	0.67
LoRA	0.66	0.63	0.64	0.64	0.68	0.57	0.68	0.64
Ours	1.00	0.98	0.99	0.89	0.97	0.79	0.97	0.84

Table 8: Comparative results of InfuserKI with PEFT and ME methods on the TReX-1k.

Methods	NR	RR	F1_T1	F1_T2	F1_T3	F1_T4	F1_T5	F1_Unseen
LLaMa-2-7B	-	-	0.63	0.58	0.61	0.61	0.60	0.61
CALINET	0.65	0.60	0.61	0.71	0.71	0.68	0.64	0.68
LoRA	0.81	0.79	0.80	0.83	0.80	0.62	0.57	0.66
Ours	1.00	0.90	0.94	0.95	0.95	0.79	0.79	0.84

Table 9: Comparative results of InfuserKI with PEFT and ME methods on the YAGO-1k KG.

our method, which integrates knowledge directly into the model parameters, significantly outperforms both RAG variants. This enhanced performance may be attributable to the direct integration of knowledge into the parameters, which effectively stimulates the model capability within specific domains. Moreover, our method exhibits lower inference latency than RAG, as it eliminates the need for external searches, and outperforms LLaMa-2-7B by delivering precise and concise answers without long explanatory texts.

Methods	F1	Latency (ms)
LLaMa-2-7B	0.40	933
RAG-Google	0.37	2027
RAG-TKS	0.42	1113
Ours	0.66	860

Table 10: Comparative results of InfuserKI with RAG methods on the UMLS KG.