Commonsense Knowledge Editing Based on Free-Text in LLMs

Anonymous ACL submission

Abstract

Knowledge editing technology is crucial for maintaining the accuracy and timeliness of large language models (LLMs) . However, the setting of this task overlooks a significant portion of commonsense knowledge based on free-text in the real world, characterized by broad knowledge scope, long content and non instantiation. The editing objects of previous methods (e.g., MEMIT) were single token or entity, which were not suitable for commonsense knowledge in free-text form. To address the aforementioned challenges, we conducted experiments from two perspectives: knowledge localization and knowledge editing. Firstly, we introduced Knowledge Localization for Free-Text(KLFT) method, revealing the challenges associated with the distribution of commonsense knowledge in MLP and Attention layers, as well as in decentralized distribution. Next, we propose a Dynamics-aware Editing Method(DEM), which utilizes a Dynamics-aware Module to locate the parameter positions corresponding to commonsense knowledge, and uses Knowledge Editing Module to update knowledge. The DEM method fully explores the potential of the MLP and Attention layers, and successfully edits commonsense knowledge based on free-text. The experimental results indicate that the DEM can achieve excellent editing performance.

1 Introduction

Large-scale Language Models (LLMs) have demonstrated remarkable performance in various natural language processing tasks. Nevertheless, errors or outdated knowledge are inevitable in LLMs (Meng et al., 2022a). Directly fine-tuning a large language model demands significant computational resources (Gupta et al., 2023), making it economically prohibitive and limiting its popularity as a preferred approach (Ding et al., 2023).

Knowledge editing serves as an effective approach to update LLMs. Existing knowledge editing methods predominantly concentrate on editing triple-based facts such as entity-relation pairs (Meng et al., 2022b), events (multiple triplets) (Peng et al., 2024; Liu et al., 2024). These approaches commonly utilize strategies involving neuron localization and editing (Meng et al., 2022a), assuming that entities and phrases within factual triplets are stored in a limited set of neurons. By manipulating these select neurons, knowledge editing can be accomplished. As shown in Figure 1, factual knowledge editing involves rectifying outdated triplets like <America, President, Trump> to accurate ones like <America, President, Biden>.

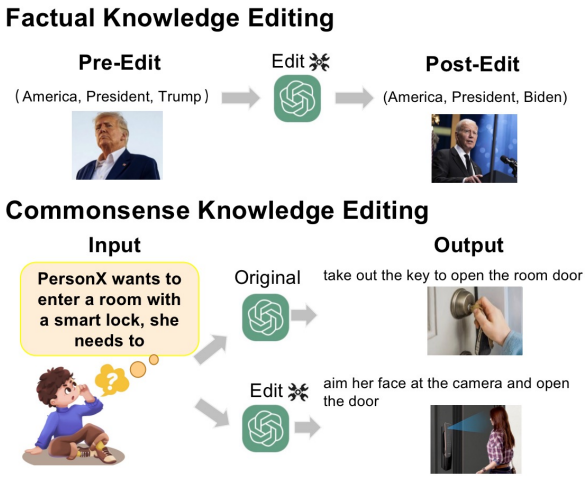

Fig. 1: An example with factual knowledge and commonsense knowledge, and obtaining the correct answer by editing the model.

However, in real-world scenarios, structured entity-relation triplets often fall short in adequately describing many knowledge pieces, especially when it comes to commonsense knowledge (Hwang et al., 2021). The data characteristics of commonsense knowledge are broad knowledge scope, long content and non instantiation, which limits the effectiveness of traditional knowledge editing methods. In addition, when using LLMs, users often need to obtain commonsense knowl-

edge in the form of free-text, rather than structured entity level information. This user preference indicates that commonsense knowledge editing based on triplet forms does not meet their needs. Therefore, we propose a more challenging commonsense knowledge editing task based on free-text, which has wider practicality.

Compared to previous methods, commonsense knowledge editing based on free-text presents some new challenges, as shown below: (1) The previous knowledge localization methods (e.g. Causal Tracing (Meng et al., 2022a)) typically used the probability value of the editing target as the response value of the knowledge storage location. The success of this method is based on the fact that the editing target is a single token or entity. However, the editing target of commonsense knowledge based on free-text editing has multiple tokens, which limits the effectiveness of previous methods. (2) Previous knowledge editing methods typically assumed that factual knowledge was stored on a single or small number of neurons, and knowledge editing could be achieved through operations on a small number of neurons. However, the experiments conducted in Section 3 indicate that commonsense knowledge based on free-text does not conform to this assumption. Commonsense knowledge based on free-text has a wide range of storage locations, is more dispersed, and is less prone to localization. Therefore, previous knowledge editing methods are insufficient for handling commonsense knowledge editing based on free-text.

To address the aforementioned challenges, we conducted experiments from two perspectives: knowledge localization and knowledge editing. Firstly, we introduce a Knowledge Localization for Free-Text(KLFT) method that include knowledge location and recall. Specifically, knowledge location experiments are utilized to determine whether commonsense knowledge is stored in the local hidden states of transformers, as well as to explore the form of storage. The knowledge recall experiment is used to verify whether specific hidden states storing commonsense knowledge have a significant contribution to that knowledge. Two experiments together indicate that, in comparison to triple facts, commonsense knowledge predominantly resides in the MLP layers and Attention (Attn) layers, the storage of knowledge is not local but rather dispersed throughout. This means that the previous editing methods (e.g., editing local layers in ROME(Meng et al., 2022a) and PMET (Li et al., 2024)) were unreasonable.

Secondly, we propose a Dynamics-aware Editing Method(DEM). Specifically, we introduce a Dynamic-aware module for real-time detection of the storage location of each commonsense knowledge, and selected the layer with the highest contribution to knowledge as the editing layer. Subsequently, we employ a Knowledge Editing module to perform targeted knowledge editing on specific MLP and Attn layers. The experimental results validated the effectiveness of the method.

To address the issue of insufficient commonsense knowledge datasets for editing based on free-text, we have developed Commonsense Knowledge Editing Benchmark (CKEBench) . This dataset has 15600 samples and six evaluation indicators, which is more challenging than the existing dataset. To the best of our knowledge, we are the first to introduce an Commonsense Knowledge Editing Benchmark. Additionally, we investigate the storage and recall of commonsense knowledge and propose an effective editing method. Our contributions can be summarized as follows:

- We constructed a Commonsense Knowledge Editing Benchmark (CKEBench) dataset that provides a benchmark for editing Commonsense knowledge based on free-text.
- Through Knowledge Localization for Free-Text (KLFT), we found that compared to triple facts, commonsense knowledge predominantly resides in the MLP layers and Attn layers, the storage of knowledge is not local but rather dispersed throughout.
- To edit commonsense knowledge based on free-text, we propose a Dynamics-aware Editing Method(DEM). Specifically, the DEM includes a Dynamic-aware Module and a Knowledge Editing Module. The experimental results validated the effectiveness of the method.

2 Constructing CKEBench Dataset

In this section, we constructed an Commonsense Knowledge Editing Benchmark(CKEBench). This datasets consist of 15,600 samples.

2.1 Dataset Construction

Based on the ATOMIC (Sap et al., 2019) database, we constructed a Commonsense Knowl-

Commonsense Knowledge Editing Benchmark(CKEBench) < ATOMIC Data Source >		
IDx	**Commonsense Prompt**	**Target Answer**
Sample 1	PersonX about to get married, as a result, PersonX wants to	live happily ever after
Sample 2	PersonX accepts PersonY appointment, resulting in	personX travels to appointment
Sample 3	PersonX can tell PersonY that PersonY is being solipsist and insolent, as a result,	others want to to stop what they're doing

Table 1: An example of converting source data from ATOMIC database into directly generated(DG), multiple-choice questions(MQ), and true/false questions(T/F).

edge Editing Benchmark(CKEBench). ATOMIC is a well-known commonsense database that was developed by Allen Institute and subsequently optimized for its version (Hwang et al., 2021). The CKEBench contains 23 types of relationships and describes commonsense knowledge based on free-text, they fall into three natural categories based on their meaning: physical-entity, social- interaction and event-centered commonsense.

2.2 Dataset Preparation

In ATOMIC, the data format is $< \text{Event}_1, \text{Relationship}, \text{Event}_2 >$, which contains some unrecognized markers (e.g. ___, etc.) and invalid characters (e.g. &, etc.), which we manually filter out. In addition, the relationship types in ATOMIC are abbreviated and not easily understood by humans. Even if ATOMIC provides corresponding annotations, it is still not enough to form a smooth statement when constructing the prompt. as shown in the Appendix A, we have rewritten the 23 relationship categories in ATOMIC into templates that can be read by humans and counted their sample sizes. Afterwards, we will use the reorganized dataset dataset as the initial data to construct the CKEBench dataset.

2.3 Dataset Analysis

After filtering and rewriting, we obtained a total of 15600 high-quality samples, of which "xAttr" had the highest number of samples, totaling 3224. The average length of "Commonsense Prompt" is 72 tokens, and the average length of Target Answer is 16 tokens. After testing on LLaMA-3 (8B) (Touvron et al., 2023a), the Perplexity (PPL) of the dataset is 7.3, indicating that the text of the entire dataset is smoother and the quality of the dataset is higher. The appendix C shows an example.

3 Knowledge Localization for Free-Text

To locate commonsense knowledge based on free-text within LLMs, we propose a Knowledge Localization for Free-Text (KLFT) method, which involves two experiments : knowledge location and recall.

3.1 KLFT Method

Inspired by causal tracing (Meng et al., 2022a), we adopt KLFT method to explore the way knowledge is stored. Similar to the causal tracing, a clean run that predicts the fact, a corrupted run where the prediction is damaged, and a corrupted-with-restoration run that tests the ability of a single state to restore the prediction.

- In the **clean run**, we pass a commonsense prompt $x = [x_1, ..., x_T]$ into model $\mathcal{F}_\theta$ and collect all hidden activations $\{h_i^l | i \in [1, T], l \in [1, L]\}$, L represents the number of hidden layers in the model. Table 1 provides an Sample 1 illustration with the commonsense prompt: " PersonX about to get married, as a result, PersonX wants to", the expected target answer is "live happily ever after".

- In the **corrupted run**, There are 23 relationship categories in the CKEBench. We consider the text before the relationship as the subject, and the text after the relationship as the object. The subject is obfuscated from $\mathcal{F}_\theta$ before the network runs. Concretely, immediately after x is embedded as $[h_1^0, h_2^0, ..., h_T^0]$, we set $h_i^0 = h_i^0 + \delta$ for all indices i that correspond to the subject entity, where $\delta \in \mathcal{N}(0, \sigma^2)$. $\mathcal{F}_\theta$ is then allowed to continue normally, giving us a set of corrupted activations $\{h_{i*}^l | i \in [1, T], l \in [1, L]\}$. Because $\mathcal{F}_\theta$ loses some information about the subject, it will likely return an incorrect answer.

- In the **corrupted-with-restoration run**, We have the $\mathcal{F}_\theta$ run calculations on noise embeddings, except in some tokens $x_{i'}$ and layers l'. Afterwards, we hook $\mathcal{F}_\theta$ and forced it to output clean state $h_{i'}^{l'}$. Future calculations can

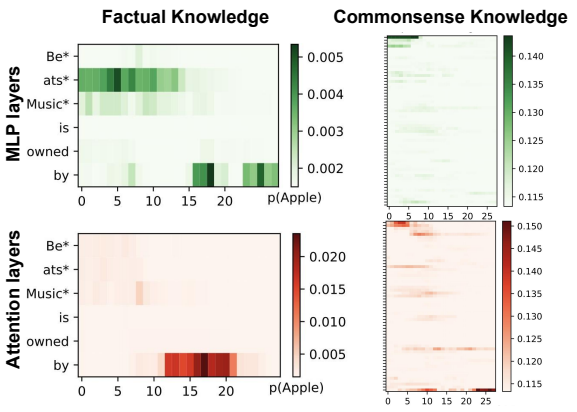

Fig. 2: Storing Factual and Commonsense Knowledge in LLMs.

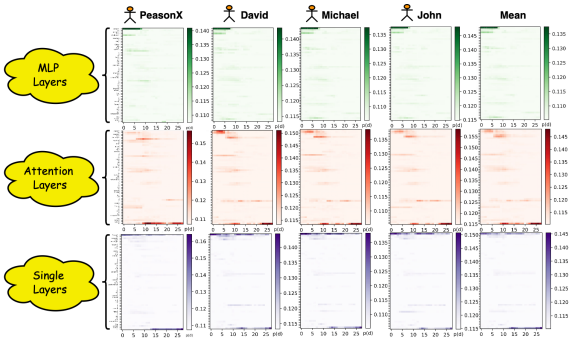

Fig. 3: The storage of commonsense knowledge after decoupling factual knowledge.

continue without intervention. Afterwards, The ability of a few clean states to restore correct facts afterwards indicates their importance in the calculation graph. Previous work (Meng et al., 2022a; Gupta et al., 2023) has shown that the last token x_{s-last} of the subject with added noise contributes the most to knowledge localization, and we fix the token x_{s-last} as the $x_{i'}$.

The probability value $P_{l'}$ of restoring the target answer will be used as the contribution of this layer l' to common sense knowledge. The larger $P_{l'}$, the greater the probability that commonsense knowledge is stored in this layer. For commonsense knowledge based on free-text , the target answer is usually a complete sentence with multiple tokens, and $P_{l'}$ cannot be directly obtained. We utilize GPT-4 (Achiam et al., 2023) and LLaMA-3 (8B) (Touvron et al., 2023a) to evaluate the semantic similarity $S_1^{l'}$ and $S_2^{l'}$ between the text output of the model and the target output, and then make $P_{l'} = \frac{S_1^{l'} + S_2^{l'}}{2}$.

3.2 Knowledge Location

3.2.1 Locating commonsense knowledge before decoupling

We compared the differences between factual and commonsense knowledge in storage locations by KLFT method. As show in the Figure 2, the fact prompt is "Beats Music is owned by", the target answer is "Apple", the commonsense knowledge is sample 3 in Table 1. The horizontal axis represents the layers in LLMs, and the vertical axis represents the tokens x_i of different knowledge. The depth of color is determined by $P_{l'}$, and the larger $P_{l'}$, the darker the color, indicating a higher probability of storing knowledge in that layer.

Unlike factual knowledge, which is typically stored in fixed MLP layers(Meng et al., 2022a), commonsense knowledge is not limited to specific layer neurons. Evidence of storage can be observed in both the MLP and Attn layers.

3.2.2 Locating commonsense knowledge after decoupling

Commonsense knowledge is non instantiation and is often abstractly represented. By contrast, facts are usually instanciated. To more accurately locate commonsense knowledge and decouple it from factual elements, we perform multiple same-type text replacements for the factual elements that may be contained in free text. For example, we replace "personX" in free-text with multiple person names and take the intersection of the located results.

As shown in Figure 3, we obtained the storage situation of commonsense knowledge decoupled from factual knowledge (The "Mean" column). Unlike factual knowledge, which is stored in the middle and front layers of MLP in LLMs, we found that commonsense knowledge is dispersed in the MLP and Attn layers, which poses a challenge for commonsense knowledge editing.

3.2.3 Locating commonsense knowledge of the entire dataset

We conducted KLFT experiment on each relationship category, selecting 100 samples for each relationship category, totaling 2300 samples. The experiment selected top k=3 layers as the storage location of knowledge. As shown in the Figure 4, the storage location of the MLP layer is mainly in the middle and front layers, but other layers also store some knowledge. Unlike the experimental

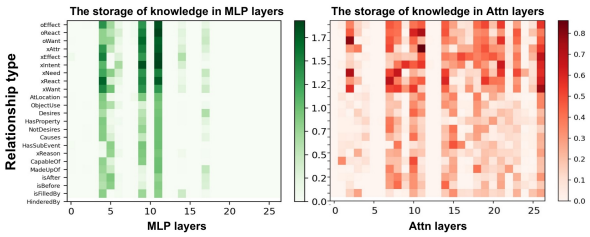

Fig. 4: Display the storage location of samples for each relationship category in the MLP and Attn layers. The horizontal axis represents the parameter layer of the model, and the vertical axis represents the relationship category. The darker the color, the more knowledge stored in that layer.

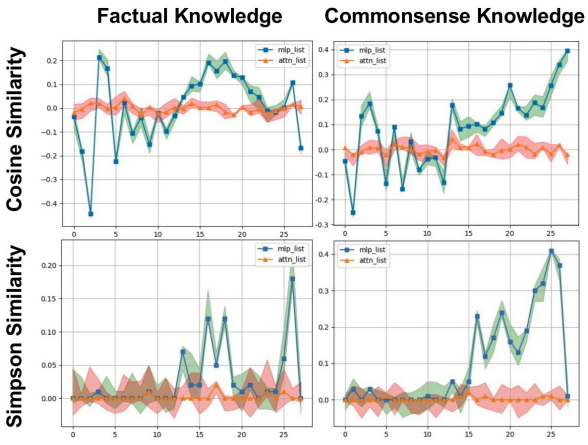

Fig. 5: The comparison of activation response results between factual and commonsense knowledge in knowledge recall process. Among them, the green line represents the MLP layer, the orange line represents the Attention layer. The horizontal axis represents different layers, and the vertical axis represents the numerical value of similarity.

results of MLP, the knowledge storage in the Attn layer is relatively scattered, with most layers storing knowledge.

3.3 Knowledge Recall

To verify the conclusions of commonsense knowledge based on free-text in localization, we recorded the contribution of MLP and Attn layers to knowledge during the recall process.

Experimental design. After passing through each layer of parameters in the model, the information flow undergoes certain changes, which we consider as an indicator to evaluate the contribution of parameter layers to knowledge. We hook model $\mathcal{F}_\theta$ and obtain the hidden states $\{h^l_{in}, h^l_{out} | l \in [1, L]\}$. Specifically, we directly compare the hidden states h^l_{in} and h^l_{out} passing through the l-th parameter layers , utilizing cosine similarity as the evaluation metric. At the same time, we utilize the h^l_{in} and h^l_{out} as the input for the final prediction lm_head layer of the model, then obtain the corresponding predicted token probabilities p^l_{in} and p^l_{out}. We take tokens with top k=50 as candidate sets, and use the Simpson algorithm to calculate the similarity between the p^l_{in} and p^l_{out}.

Data selection. For factual and commonsense knowledge, we selected 1150 samples each to explore the process of knowledge recall. Among them, there are a total of 23 relationship categories for commonsense knowledge, with 50 samples selected for each relationship category. We assume that the similarity is inversely proportional to the contribution of corresponding knowledge. When the similarity is close to zero, it indicates that the layer has the greatest impact on knowledge during the knowledge recall process.

Result analysis. As shown in the Figure 5, for the MLP layer, the similarity of factual knowledge is much greater than zero in the middle part and close to zero in the rest, while the similarity of commonsense knowledge is only close to zero in the middle and front parts. For the Attn layer, the similarity between factual knowledge and commonsense knowledge is close to zero at most layer, but there is also a certain difference in values. The experimental results show that the localization results of the KLFT method are consistent with the parameter layer response of knowledge recall process. For commonsense knowledge based on free-text, which is mainly stored in the middle and front layers of MLP as well as most Attn layers.

4 Dynamics-aware Editing Method

To edit commonsense knowledge based on free-text, we propose a Dynamics-aware Editing Method(DEM). Specifically, the DEM includes a Dynamics-aware Module and Knowledge Editing Module.

4.1 Dynamics-aware Module

Through section 3, we conclude that unlike factual knowledge, commonsense knowledge is stored in the MLP and Attn layers, and the storage locations of knowledge are relatively scattered. The existing knowledge editing methods always edit all factual knowledge at fixed parameter layer. For example, when editing all samples on GPT-J (6B)

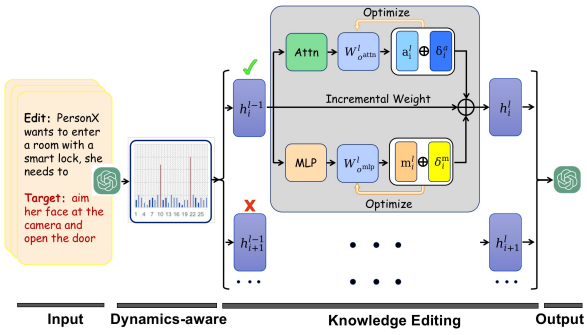

Fig. 6: The overall architecture of the Dynamics-aware Editing Method.

(Wang and Komatsuzaki, 2021) model, the edited layers for the ROME (Meng et al., 2022a) and PMET (Li et al., 2024) methods are fixed [5] and [3,4,5,6,7,8], respectively, which is obviously unreasonable for editing commonsense knowledge.

As shown in the Figure 6, we propose a Dynamics-aware module for selecting MLP and Attn layers for editing. When commonsense prompts $x = [x_1, ..., x_T]$ input to model $\mathcal{F}_\theta$, the information flow will change after passing through parameters layer. We hook $\mathcal{F}_\theta$ to obtain the last token's hidden state $\{h(T)^l_{in}, h(T)^l_{out} | l \in [1, L]\}$. The $h(T)^l_{in}$ and $h(T)^l_{out}$ represent the hidden states of the token's input and output in l-th layer, respectively. Then we utilize Cosine Similarity as an indicator for selecting editing layers:

$$\text{Cosine\_Similarity} = \frac{h(T)^l_{in} \cdot h(T)^l_{out}}{\|h(T)^l_{in}\| \|h(T)^l_{out}\|} \quad (1)$$

the closer the Cosine Similarity is to zero, the greater the contribution of this layer to knowledge. Select layers with top k=3 for editing.

4.2 Knowledge Editing Module

We edit the selected layers $\hat{l}$ of the dynamic perception module in section 4.1. For a given question $x = [x_1, ..., x_T]$, where x_i represents the i-th token of the question, and T represents the number of question tokens. The model $\mathcal{F}_\theta$ generates text by iteratively sampling from a conditional token distribution $\mathbb{P}(o_1, ..., o_n | x_1, ..., x_T)$, where o_j represents the j-th token of the output. We utilize $\{h^l_i | i \in [1, T], l \in [1, L]\}$ to represent the hidden state of x_i in the l-th layer.

4.2.1 Step1: Obtaining Incremental Weights

DEM first computes the target answer representations in the selected layers $\hat{l}$ of MLP and Attn by simultaneously optimizing the TC (Transformer Component, namely MLP and Attn) hidden states. Secondly, DEM updates both MLP and Attn weights in the critical layers through target answer o_j representations. Overall, DEM optimizes an objective function to obtain target weights (Meng et al., 2022b):

$$W_{\text{MLP}}, W_{\text{Attn}} \triangleq \underset{W}{\text{argmin}}((\sum_{i=1}^{n}(\|Wk_i - v_i)\|^2 + \sum_{i=n+1}^{n+u}(\|Wk_i - v_i)\|^2)) \quad (2)$$

where $k_i \triangleq k_i^{\hat{l}}$ and $v_i \triangleq v_i^{\hat{l}}$ represent the sets of keys and values, respectively, encoding the commonsense prompt in $\hat{l}$-th layer. $\sum_{i=1}^{n}(\|Wk_i - v_i)\|^2$ indicates that we want to retain n pieces of knowledge, while $\sum_{i=n+1}^{n+u}(\|Wk_i - v_i)\|^2$ indicates that we want to modify $u >> 1$ pieces of knowledge. We represent the keys and val- ues as matrices stacked horizontally: $[k_1|k_2|...|k_n] \triangleq K$ and $[v_1|v_2|...|v_n] \triangleq V$, and we consider the target weight W_{MLP} and W_{Attn} as the sum of the original weight W_0^{MLP} and W_0^{Attn}, and the incremental weight $\triangle$ (i.e. $W_{\text{MLP}} = W_0^{\text{MLP}} + W_\triangle^{\text{MLP}}$ and $W_{\text{Attn}} = W_0^{\text{Attn}} + W_\triangle^{\text{Attn}}$). Based on the derivation from MEMIT (Meng et al., 2022b), the formal expression for the incremental weight is:

$$\begin{aligned} \triangle^{MLP} &= R^{MLP}(k_1^{MLP})^T(C_0^{MLP} + k_1^{MLP}(k_1^{MLP})^T)^{-1} \\ \triangle^{Attn} &= R^{Attn}(k_1^{Attn})^T(C_0^{Attn} + k_1^{Attn}(k_1^{Attn})^T)^{-1} \end{aligned} \quad (3)$$

where $R^{MLP} \triangleq V_1^{MLP} - W_0^{MLP} K_1^{MLP}$ represents the residual between the values V_1^{MLP} (namely target answer representations) corresponding to the keys K_1^{MLP} of the target knowledge and the model original knowledge $W_0^{MLP} K_1^{MLP}$. $C_0^{MLP} \triangleq k_0^{Attn}(k_0^{Attn})^T = \alpha \mathbb{E}[kk^T]$ is an estimate of the set of previously memorized keys obtained through sampling. Here, α is a hyperparameter which balances the degree of model modification and preservation.

We consider modifying the original answers related to commonsense prompts $x = [x_1, ..., x_T]$ in LLMs to target answers $o = [o_1, ..., o_n]$. Assuming that the set of previously memorized keys C_0^{MLP} has already been obtained through sampling, and knowledge clues x_i have been inputed into the original model to obtain $W_0^{MLP} K_1^{MLP}$, we then need the sets of keys and values for the target knowledge, denoted as K1 and V1, respectively. Similar to MEMIT (Meng et al., 2022b), we calculate the target answer set of the edited layer

$L = max(R^{MLP})$. The relevant parameters of Attn and MLP layers are similar.

4.2.2 Step2: Updating Weights

As shown in the Figure 6, a_i^l and m_i^l are the hidden states of the Attn and MLP of the l-th layer and the i-th token, respectively. The general forms of the Attn and MLP at the l-th layer and the i-th token x_i^l are given by:

$$\begin{aligned} a_i^l &= W_{o^{attn}}^l Attn^l(\gamma(h_1^{l-1}, h_2^{l-1}, ..., h_i^{l-1})), \\ m_i^l &= W_{o^{mlp}}^l \Phi(W_I^l \gamma(h_j^{l-1})) \end{aligned} \tag{4}$$

Where $W_{o^{attn}}^l$ and $W_{o^{mlp}}^l$ are the output weights of the Attn and MLP at the l-th layer, respectively. W_I^l are the input weights of the MLP at the l-th layer. The Φ represents the non-linear activation function.

DEM adds optimizable parameters δ_i^m and δ_i^a to hidden states v_i^m and v_i^a at the l-th layer, respectively. DEM retains the optimized hidden state of MLP and Attn to update their weights separately, denoted as $v_i^m = m_i^l + \delta_i^m = argmin\mathcal{L}(v_i^m)$ and $v_i^a = a_i^l + \delta_i^a = argmin\mathcal{L}(v_i^a)$. The formulas $\mathcal{L}(v_i^m)$ and $\mathcal{L}(v_i^a)$ are similar, with the main difference being their application in MLP and Attn calculations. The $\mathcal{L}(v_i^m)$ is defined as follows:

$$\begin{aligned} \mathcal{L}\left(v_i^m\right) &= \alpha \cdot D_{\mathrm{KL}}\left(\mathbb{P}_{\mathbb{F}_\theta^\dagger}\left[\boldsymbol{y}^m \mid p^m\right] | \mathbb{P}_{\mathcal{F}_\theta}\left[\boldsymbol{y}^m | p^m\right]\right) \\ &+ \beta \cdot \frac{1}{P} \sum_{j=1}^{P} -\log \mathbb{P}_{\mathcal{F}_\theta^\dagger}\left[\boldsymbol{y}_i^{Z_t} | \operatorname{pref}_j \oplus p\left(\boldsymbol{x}_i^m\right)\right]. \end{aligned} \tag{5}$$

Where $\mathcal{F}_\theta^\dagger \triangleq \mathcal{F}_\theta(a_i^l += \delta_i^a)$ represents the optimizable parameters δ_i^a is added to the hidden states of Attn at the l-th layer of the model $\mathcal{F}_\theta$. The α and β are hyperparameters used to balance reliability and specificity. $\operatorname{pref}_j \oplus p\left(\boldsymbol{x}_i^m\right)$ is utilized to enhance the prefix of target knowledge generalization and commonsense knowledge generalization (such as randomly replacing person names).Simultaneously calculate KL divergence and stack the calculation results into matrix V_1.

With this, DEM follows the same algorithm steps as PMET (Li et al., 2024) to update MLP and Attn weights.

5 Experiments

In the section, we investigated the effectiveness of DEM method and existing editing methods in editing commonsense knowledge based in free-text.

5.1 Experimental Setup

Baselines and Datasets. Our experiments are conducted on GPT-J (6B) (Wang and Komatsuzaki, 2021) and LLaMA-2 (7B) (Touvron et al., 2023b). The baseline methods include the learning-based method MEND, and locating and editing the methods Fine-Tuning (FT+W) (Zhu et al., 2020), MEND (Mitchell et al., 2021), MEMIT (Meng et al., 2022b) and PMET (Li et al., 2024). We chose the CKEBench dataset we constructed as the benchmark.

Evaluation. For CKEBench datasets, the target answer is an free-text that contains multiple tokens. Therefore, we utilize the GPT-4 (Achiam et al., 2023) model to determine the similarity between the generated text and the original text as the experimental result. Similar to the factual knowledge, the evaluation metrics include Score, Efficiency, Generalization, Specificity, Fluency and Consistency. In addition, we have added a Commonsense indicator to evaluate the ability of the method to edit commonsense knowledge. The data in the "sub_neighborhood_prompts" at Appendix C is utilized to evaluate this indicator.

5.2 Overall Results

We conduct experiments on commonsense knowledge datasets to verify the effectiveness of our method DEM.

Results on the GPT-J (6B). The Table 2 shows that DEM performs better than baselines methods. Specifically, DEM built upon GPT-J (6B), is **+4.5** better on indicator Score than PMET, and obtains a new state-of-the-art(SOTA) result. Meanwhile, our method achieves **13.8**% improvements of Commonsense score on the true/false questions dataset. The significant performance gain of our method over the baselines demonstrates that the proposed DEM is very effective for this task.

Results on the LLaMA-2 (7B). As show in Table 2, Our method improves upon the basic PMET method by **15.7**% and **5.6**% in term of F1 Commonsense score and Specificity score on the LLaMA-2 (7B), respectively. Meanwhile, our DEM achieves **3.0**% improvements of Score. We attribute the improvements to that our method DEM takes advantage of Dynamics-aware and Knowledge Editing Module, thus achieving superior performance than the previous model PMET.

Editor	Score	Efficacy	Generalization	Specificity	Fluency	Consistency	Commonsense
GPT-J (6B)	12.4	14.5	12.1	9.4	605.3	20.9	7.2
FT-W	22.7	39.3	20.4	21.5	313.5	25.7	11.8
MEND	25.8	29.5	22.7	31.3	501.2	27.8	14.7
MEMIT	31.6	45.6	21.8	35.6	556.9	33.7	21.8
PMET	39.8	56.8	53.3	48.8	**619.7**	44.7	27.9
DEM(ours)	**44.3**(↑4.5)	**60.3**(↑3.5)	**57.4**(↑4.1)	**50.3**(↑1.5)	611.3	**45.6**(↑0.9)	**41.7**(↑13.8)
LLaMA-2(7B)	13.7	18.7	13.5	12.3	617.7	19.9	9.2
MEMIT	33.5	42.9	27.3	36.3	600.8	33.5	23.8
PMET	40.5	58.7	55.9	47.3	**615.5**	47.2	27.7
DEM(ours)	**43.5**(↑3.0)	**62.2**(↑3.5)	**57.3**(↑1.4)	**52.9**(↑5.6)	609.8	**50.3**(↑3.1)	**43.4**(↑15.7)

Table 2: The main results directly generated in the CKEBench dataset. The performance of our method is followed by the improvements (↑) over the previous method.

Model	Efficacy	Commomsense
GPT-J (6B)(DEM)	60.3	41.7
w/o DA	57.6 (↓ 2.7)	31.5 (↓ 10.2)
w/o EM	18.8 (↓ 41.5)	9.3 (↓ 32.4)
w/o EA	58.5 (↓ 1.8)	40.1 (↓ 0.6)

Table 3: Ablation study of DEM. We turn off different components of the model one at a time.

5.3 Ablation Study

To show the efficacy of our proposed techniques, we conduct an ablation study experiment by turning off one component at a time. 1) w/o DA, which removes the Dynamics-aware module; 2) w/o EM, which does not edit MLP layers in the Knowledge Editing module, only the Attn layers; 3) w/o EA, which does not edit Attn layers in the Knowledge Editing module, only the MLP layers; . We present the results of ablation study in Table 3. From the results, we can observe that:

(1) **Effectiveness of Dynamics-aware module.** When we remove the Dynamics-aware module from the DEM, the Score drops by 10.2% on commonsense knowledge dataset. It proves the Dynamics-aware module is very effective for the task.

(2) **Effectiveness of not editing MLP layers.** Not editing the MLP layer, the performance drops significantly. Specifically, the Efficacy score drops from 60.3% to 18.8% on the commonsense dataset.

(3) **Effectiveness of not editing Attn layers.** Compared without editing Attn layers, our method DEM achieves 1.8% improvements of Efficacy score on the commonsense dataset. It demonstrates that the Attn layer is crucial for editing commonsense knowledge.

6 Related Work

The existing knowledge editing dataset can be divided into triplet form and event form. In triplet format dataset, commonsense knowledge dataset includes PEP3k and 20Q (Porada et al., 2021; Gupta et al., 2023), factual knowledge includes ZsRE (Levy et al., 2017), CounterFact (Meng et al., 2022a), Fact Verification (Mitchell et al., 2022) , Calibration (Dong et al., 2022), MQuAKE (Zhong et al., 2023) and RaKE (Wei et al., 2023). In event format dataset, datasets with only factual knowledge, including ELKEN (Peng et al., 2024) and EVEDIT (Liu et al., 2024).

The previous editing methods mainly focused on editing knowledge in the form of triples, with a small amount of knowledge in the form of editing events. The methods for editing triplet forms mainly include : (1)Locate-Then-Edit method (Dai et al., 2021; Meng et al., 2022a,b; Li et al., 2024), (2) Memory-based method (Mitchell et al., 2022; Madaan et al., 2022; Zhong et al., 2023; Zheng et al., 2023), (3) Hyper-network method (Mitchell et al., 2021; De Cao et al., 2021; Tan et al., 2023). The method for editing event forms is Self-Edit (Liu et al., 2024).

7 Conclusion

In this paper, we aim to edit commonsense knowledge based on free-text. Firstly, we constructed CKEBench dataset that provides a benchmark for editing Commonsense knowledge based on free-text. Additionally, we propose a KLFT method, and concluded that commonsense knowledge is dispersed in the MLP and Attn layers. Finally, we propose the DEM method to edit commonsense knowledge, and the experimental results verify the effectiveness of this method.

8 Limitations

Due to limitations in computing resources, we did not conduct relevant experiments on larger language models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. *arXiv preprint arXiv:2210.03329*.

Anshita Gupta, Debanjan Mondal, Akshay Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket Tandon. 2023. Editing common sense in transformers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8214–8232.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. On symbolic and neural commonsense knowledge graphs.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18564–18572.

Jiateng Liu, Pengfei Yu, Yuji Zhang, Sha Li, Zixuan Zhang, and Heng Ji. 2024. Evedit: Event-based knowledge editing with deductive editing boundaries. *arXiv preprint arXiv:2402.11324*.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.

Hao Peng, Xiaozhi Wang, Chunyang Li, Kaisheng Zeng, Jiangshan Duo, Yixin Cao, Lei Hou, and Juanzi Li. 2024. Event-level knowledge editing. *arXiv preprint arXiv:2402.13093*.

Ian Porada, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2021. Modeling event plausibility with consistent conceptual abstraction. *arXiv preprint arXiv:2104.10247*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*.

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. *arXiv preprint arXiv:2302.13971*.

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. *arXiv preprint arXiv:2307.09288*.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Yifan Wei, Xiaoyan Yu, Huanhuan Ma, Fangyu Lei, Yixuan Weng, Ran Song, and Kang Liu. 2023. Assessing knowledge editing in language models via relation perspective. *arXiv preprint arXiv:2311.09053*.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *arXiv preprint arXiv:2305.12740*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Appendix A

Relations	Human Readable Template	Size
oWant	as a result, personY want to as a result, others want to *	7775
xEffect	as a result, PersonX will resulting in *	13862
xIntent	because PersonX wanted which means *	8558
xNeed	which means PersonX need	13734
xWant	as a result, PersonX wants	7775
xReact	which indicates that personX	10689
oEffect	resulting in personY resulting in others *	5181
oReact	and personY's reaction is and others's reaction is *	4740
xAttr	which means that PersonX which means that *	19441
AtLocation	located in the	234
ObjectUse	are used to	311
Desires	desires	271
HasProperty	has the property of	428
NotDesires	have no desire to	287
Causes	causes	322
HasSubEvent	The sub event of E1 is to E2	118
xReason	The reason for E1is E2	290
CapableOf	is/are capable of	512
MadeUpOf	made up of	291
isAfter	happens after	465
isBefore	happens before	164
isFilledBy	blank can be filled by	174
HinderedBy	can be hindered by	612

Table 4: The correspondence between relationships and rewriting templates in the ATOMIC database. Among them, "*" represents that the token "personX/personY" in $< \text{Event}_1, \text{Relationship}, \text{Event}_2 >$ is not in Event_1 or Event_2. "E1" and "E2" represent Event_1 or Event_2.

B Appendix B: Effects of the existing methods

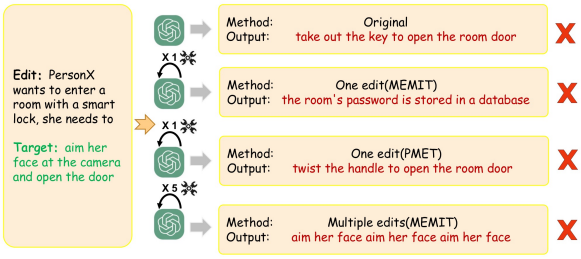

Fig. 7: Examples of commonsense knowledge editing using existing methods.

We conducted experiments to evaluate the efficacy of existing methods in editing commonsense knowledge based on free-text. As show in Figure 7, "One edit" refers to editing the sample once, while "Multiple edits" involves editing the sample five times. It is observed that both the original output and the utilization of MEMIT (Meng et al., 2022b) methods (including One and Multiple edits) fail to effectively edit commonsense knowledge. Furthermore, multiple edits lead to repeated instances of partial target answers. These experimental findings highlight the limitations of existing methods in editing commonsense knowledge based on free-text.

C Appendix C


```
{
  "case_id": 9313
  "requested_rewrite": {
    "prompt": "{}, as a result, others want to to stop what they're doing",
    "subject": "PersonX can tell PersonY that PersonY is being solipsist and insolent",
    "subsubject": "PersonX",
    "target_new": {
      "str": "to stop what they're doing"
    },
  },
  "paraphrase_prompts": [
    "PersonX can inform PersonY that they are acting solipsistic and disrespectful, as a result, personY want to to
        stop what they're doing",
    "PersonX has the ability to inform PersonY about their solipsistic and insolent behavior, as a result, personY
        want to to stop what they'redoing",
  ],
  "neighborhood_prompts": [
    "PersonX breaks the enemy 's ___, as a result, others want to to stop what they're doing",
    "PersonX asks the neighbors, as a result, others want to to stop what they're doing",
  ],
  "sub-neighborhood_prompts": [
    "James can tell PersonY that PersonY is being solipsist and insolent, as a result, personY want to to stop
        what they're doing",
    "David can tell PersonY that PersonY is being solipsist and insolent, as a result, personY want to to stop what
        they're doing",
  ],
  "sub-neighborhood_prompts_rewrite": [
  ],
  "generation_prompts": [
    "PersonX can tell PersonY that PersonY is being solipsist and insolent,  and personY's reaction is to stop
        what they're doing",
    "PersonX can tell PersonY that PersonY is being solipsist and insolent,  resulting in personY to stop what
        they're doing",
  ],
}
```


Fig. 8: Sample id:9313 of CKEBench dataset. Due to space constraints, this sample only displays the structure rather than the entirety