Editing Factual Knowledge and Explanatory Ability of Medical Large Language Models

Anonymous ACL submission

Abstract

Model editing aims to precisely modify the behaviours of large language models (LLMs) on specific knowledge while keeping irrelevant knowledge unchanged. It has been proven effective in resolving hallucination and out-of-date issues in LLMs. As a result, it can boost the application of LLMs in many critical domains (e.g., medical domain), where the hallucination is not tolerable. In this paper, we propose two model editing studies and validate them in the medical domain: (1) directly editing the factual medical knowledge and (2) editing the explanations to facts. Meanwhile, we observed that current model editing methods struggle with the specialization and complexity of medical knowledge. Therefore, we propose MedLaSA, a novel **La**yer-wise **S**calable **A**dapter strategy for medical model editing. It employs causal tracing to identify the precise location of knowledge in neurons and then introduces scalable adapters into the dense layers of LLMs. These adapters are assigned scaling values based on the corresponding specific knowledge. To evaluate the editing impact, we build two benchmark datasets and introduce a series of challenging and comprehensive metrics. Extensive experiments on medical LLMs demonstrate the editing efficiency of MedLaSA, without affecting irrelevant knowledge that is not edited.

1 Introduction

Recent researches have demonstrated that the large language models (LLMs) can serve as a knowledge base to store facts about the world and possess remarkable understanding ability to facts (Petroni et al., 2019; Geva et al., 2022). Considering the substantial cost of retraining LLMs, there has been an increasing interest in model editing (also known as knowledge editing), which seeks to modify the behaviors of LLMs by precisely manipulating a part of knowledge while ensuring other stored knowledge unaffected (Zhang et al., 2024).

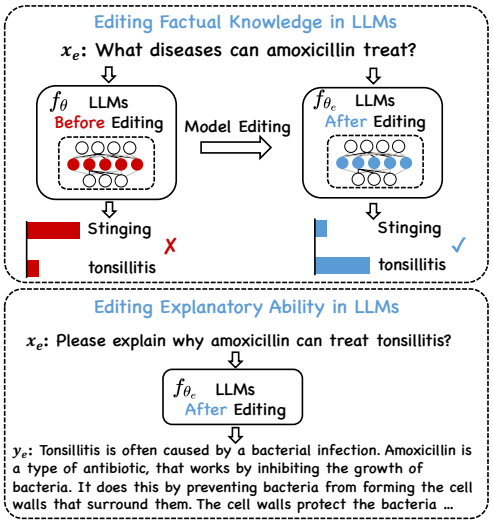

Figure 1: Examples of two editing methods.

However, the potential of model editing to modify specialized knowledge, particularly in medical domain, has not been fully explored. While LLMs have proven to be valuable tools, they may still provide outdated factual information or even experience hallucinations, which is particularly concerning when deployed in real-world medical scenarios. (Hartvigsen et al., 2022; Feng et al., 2023). This research gap presents a question: *Can model editing techniques effectively address the challenges when integrating LLMs in medical domain?*

In order to explore and promote model editing techniques to solve out-of-date and hallucination problems in medical LLMs by editing medical knowledge, we propose two preliminary studies as shown in Figure 1: (1) editing factual medical knowledge within LLMs to ensure up-to-date information, and (2) editing LLMs to enhance its ability to explain these facts to mitigate hallucinations. To facilitate our exploration, we construct two corresponding benchmarks, namely Medical Counter Fact (MedCF) and Medical Fact Explanation (MedFE), enabling us to evaluate model

editing approaches from four perspectives: *Efficacy*, *Generality*, *Locality*, and *Fluency* (Zhang et al., 2024). Additionally, given the higher demand for reliability in the medical domain (Zhou et al., 2023), we propose more challenging and comprehensive metrics for *Locality* evaluation.

Despite the remarkable achievements of model editing methods in general domains (Meng et al., 2022a), we observed that they struggle to overcome the challenges in the medical model editing due to the specialization and complexity of medical language (Karabacak and Margetis, 2023). These methods either overlook the storage of complex medical knowledge across different layers of LLMs (Zheng et al., 2023) or may introduce substantial modifications to original parameters, which consequently affect the model's unrelated knowledge (not the target of editing) and lead to suboptimal performance (Meng et al., 2022b).

To handle medical model editing, we propose MedLaSA, a novel **La**yer-wise **S**calable **A**dapter strategy. MedLaSA employs a casual tracing method (Meng et al., 2022a) to associate medical knowledge to corresponding layers. By focusing on the layers where knowledge-to-edit resides, the targeted knowledge can be modified efficiently and other knowledge can be left unchanged. Extensive experiments conducted on MedCF and MedFE have demonstrated superior performance of MedLaSA across a range of metrics. Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to propose medical LLM editing for factual knowledge and explanatory abilities in medical LLMs by constructing two benchmarks with comprehensive evaluation metrics.[1]
- We propose MedLaSA to dynamically adjust the adapters across different layers of LLMs based on medical knowledge and automatically categorize whether the input knowledge requires editing.
- We conduct extensive experimental analysis of complex and specialized medical knowledge model editing, which demonstrates that MedLaSA significantly outperforms the existing cutting-edge methods.

[1]The data and code will be open-sourced upon publication.

2 Related Work

We present current model editing works in two categories following Yao et al. (2023).

2.1 Memories or Additional Parameters

The methods of this category typically involve creating explicit memories to store the required knowledge for editing, or adding additional trainable parameters to LLMs for learning new knowledge (Yu et al., 2023; Dong et al., 2022; Hartvigsen et al., 2022). SERAC (Mitchell et al., 2022b) utilized explicit memory for storing edits and incorporated a scope classifier to understand the editing scope. Given a sample within the editing scope, it utilized a separate model to make edits, ensuring that the original model remains unaffected. Inspired by the in-context learning ability of LLMs, IKE (Zheng et al., 2023) designed demonstration formatting and organization strategies, including the copy, update, and retain templates, and retrieved relevant knowledge facts from the editing memories as demonstration inputs to guide the editing process. T-Patcher (Huang et al., 2023) retained all original parameters to preserve overall performance while adding trainable neuron patches to the last Feed-Forward Network (FFN) layer of a Transformer for handling sequential model editing. Despite their success, the above methods lack the exploration of the mechanics of knowledge storage in LLMs, which ultimately leads to poor performance in handling complex medical knowledge.

2.2 Modifying LLMs' Parameters

The methods of this category aim to comprehend how knowledge is stored in LLMs and how it can be effectively altered by changing the parameters (De Cao et al., 2021; Geva et al., 2021; Wu et al., 2023). KN (Dai et al., 2022) proposed a knowledge attribution method to identify the neurons associated with specific knowledge without fine-tuning, updating facts, and erasing relations by directly modifying the corresponding parameters in FFN. MEND (Mitchell et al., 2022a) introduced auxiliary hyper-networks to transform the gradient during the fine-tuning process, and trained the hyper-networks to ensure edit success and locality when updating LLMs' parameters. ROME (Meng et al., 2022a) applied causal mediation analysis (Pearl, 2022; Vig et al., 2020) to identify decisive neuron activation and modify FFN weights by solving a least squares problem with a linear equality constraint using the

Lagrange multiplier. As an extension of ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b) introduced a multi-layer algorithm to update multiple cases simultaneously. PMET (Li et al., 2023) further improved MEMIT (Meng et al., 2022b) by simultaneously optimizing hidden states of self-attention and FFN. Despite impressive progress made by these methods, they often introduce significant modifications to the original parameters. Consequently, unrelated knowledge is affected, resulting in a noticeable impact on *Locality* and *Fluency*, as demonstrated in Section 4.

3 Methodology

3.1 Prelimimaries

Model editing is a recently emerging field that aims to modify specific knowledge within a neural network while preserving the network's behaviours for other knowledge (Zhang et al., 2024; Yao et al., 2023). In contrast to vanilla fine-tuning for updating LLMs, model editing seeks to precisely manipulate and update the specific knowledge in LLMs, resulting in a more thorough and strict evaluation (Wang et al., 2023b). Formally, we denote a model as $f(x;\theta)$, which maps an input x to its prediction y with the pretrained model parameters θ, and the post-edited model is denoted as $f'(\theta_e)$. To be considered effective, model editing typically needs to satisfy the following four properties (Huang et al., 2023; Zhang et al., 2024):

Property 1 Efficacy. The post-edited model should establish an effective mapping between the edit pair (x_e, y_e), i.e., $f'(x_e, \theta_e) = y_e$.

Property 2 Generality. When an input sentence x_s with a similar meaning to x_e (e.g., a rephrased sentence) is provided, the post-edited model is expected to produce the corresponding output y_e as well, i.e., $f'(x_s, \theta_e) = y_e$.

Property 3 Locality. The editing process should remain local and precise, meaning the post-edited model should not impact the prediction of irrelevant example pairs (x_i,y_i), i.e., $f'(x_i, \theta_e) = y_i$.

Property 4 Fluency. The post-edited model should maintain generation ability and thus a high level of fluency in output, which is evaluated by calculating a weighted average of bi- and tri-gram entropies, as described by Meng et al. (2022a).

3.2 Casual Tracing

We first introduce casual tracing, which aims to identify factual associations to specific neuron activations by calculating the contribution of each state towards factual predictions (Meng et al., 2022a). The knowledge and its associations in the network can be effectively utilized to regulate model editing and scaling operations in our model, as described in Section 3.3. This process involves three forward propagation runs: (1) **Clean run.** A factual knowledge x is fed into model, and the hidden activations $\{h_i^l | i \in [1,T], l \in [1,L]\}$ of every token i of T tokens and every layer l of L layers are collected. (2) **Corrupted run.** The subject of x is obfuscated by introducing Gaussian noise $\epsilon \sim N(0; v)$ with zero mean and standard deviation of v to the subject embedding of x, and we can get a set of corrupted activations $\{h_{i*}^l | i \in [1,T], l \in [1,L]\}$. (3) **Corrupted-with-restoration run.** The input noisy embeddings are kept the same as in the corrupted run, but the hidden activations h_{i*}^l of each token and layer are replaced with h_i^l as in the clean run. The probability of restoring the correct output, as in the clean run, indicates the causal association between knowledge and hidden states. The restoration operation is performed separately on each token within every layer for a single piece of knowledge and generates an impact matrix $M \in \mathbb{R}^{T \times L}$. We present heatmaps of the impact matrix of the MedCF and MedFE datasets in Appendix D for better understanding.

3.3 MedLaSA

In this section, we introduce MedLaSA, a simple yet effective model editing strategy. MedLaSA is designed to modify each layer in a tailored manner by taking into account the associations between multiple layers and medical knowledge while ensuring that irrelevant knowledge remains unaffected during the modification process. We first apply causal tracing to each piece of medical knowledge (as shown in Section 3.2), which has been proven effective in identifying specific hidden states that are crucial when recalling a fact (Meng et al., 2022a). Unlike previous methods such as ROME (Meng et al., 2022a), which directly modify the MLP weights of corresponding layers, we argue that adding an adapter to dense weights is a more effective way to insert new knowledge while mostly preserving the original abilities of LLMs.

Our motivation lies in enabling the model to automatically discriminate whether the input knowledge requires editing (*Efficacy* and *Generality*) or not (*Locality*, *Fluency*), which is achieved by applying different scales of adjustment to adapter of

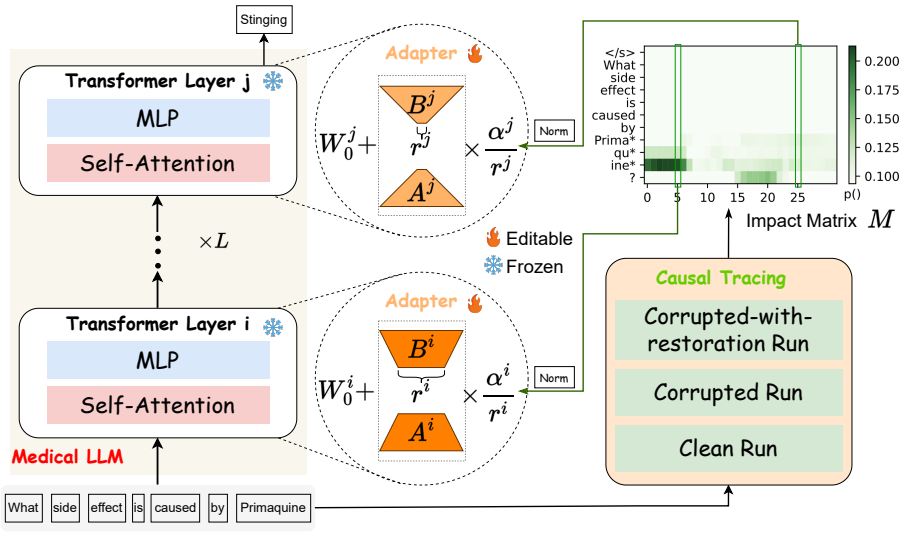

Figure 2: The overview of our proposed MedLaSA. We demonstrate the process of inputting individual pieces of data. This process is applicable to both edited and unedited knowledge. The activation scale of adapters in different layers varies depending on the knowledge in data, ensuring that unrelated knowledge is not significantly impacted after the editing process.

the layer where knowledge is located. For instance, as illustrated in Figure 2, when data that needs editing is inputted, we increase its scale on i-th layer and decrease its impact on j-th layer, ensuring that knowledge updates occur at i-th layer. Conversely, when locality data is input, the influence of adapter on j-th layer is activated and reduces impact of adapter on i-th layer. Inspired by LoRA (Hu et al., 2021), we incorporate a parallel low-rank adapter into dense layers to enable parameter-efficient fine-tuning. Specifically, a pre-trained weight matrix $W_0 \in \mathbb{R}^{d\times k}$ from attention or MLP module is updated with two trainable matrics $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. For layer l, we have:

$$h^l = W_0^l x + \frac{\alpha^l}{r^l} B^l A^l x, \tag{1}$$

where, the rank $r^l \ll min(d, k)$ and the α^l in our work are utilized to adjust the number and location of trainable parameters required for knowledge updating, respectively.

Scaling α. Parameter α is used to measure the weight of the adapter relative to the original network. Each individual knowledge has a specific α value and can generate an impact matrix M through casual tracing in Section 3.2. The impact I_α:

$$I_\alpha = norm(\sum_{t\in\mathbb{E}_s} M_t), \tag{2}$$

where $\mathbb{E}_s$ is defined as the set of index of subject tokens, $norm(\cdot)$ denotes max-min normalization. The final scale of layer l can be computed by multiplying hyper-parameter α_o with I_α value in the i-th layer: $\alpha^l = \alpha_o \times I_\alpha^l$.

Scaling Rank. The rank r is used to control the number of additional parameters required to update new knowledge, which is generalized for knowledge in whole dataset $\mathbb{D}$ and is specific to each layer l:

$$I_r = norm(\sum_{M\in\mathbb{D}} \sum_{t\in\mathbb{E}_s} M_t). \tag{3}$$

The final rank of layer l can be computed by multiplying hyper-parameter r_o: $r^l = \lceil r_o \times I_r^l \rceil$.

What is more, a transformer block typically includes self-attention and MLP modules, whose effects can be separately analyzed with causal tracing. For example, to measure the impact matrix M_{attn} of the attention module, the MLP calculation is cut off and frozen in its corrupted run state, so that it is not affected by the insertion of a clean state. We conducted a thorough analysis to examine the influence of various weights and hyperparameters on model editing, in Section 4.

4 Experiments

4.1 Medical Model Editing Benchmarks

We aim to investigate the effectiveness of model editing techniques in the medical domain. On one hand, there are potential concerns regarding medical knowledge stored in medical LLMs (e.g., outdated information and hallucination), which could

result in errors in diagnosis or treatment recommendations (Zhou et al., 2023). On the other hand, real medical scenarios demand a high level of reliability, which has led to increased emphasis on the explanatory ability of LLMs, such as their ability to demonstrate a logical chain-of-thought during the decision-making process (Karabacak and Margetis, 2023). Therefore, we construct the MedCF dataset for (1) editing factual medical knowledge and the MedFE dataset for (2) editing explanation ability of LLMs. The statistics are shown in Table 1.

Medical Counter Fact Dataset. We build the MedCF dataset using a medical knowledge graph (Ioannidis et al., 2020) and corresponding text (Xu et al., 2023) as the source. To evaluate the ability to edit knowledge with unknown prediction results of LLMs, same as Meng et al. (2022a), we replace the tail entity t in triplets (h, r, t) and construct a set of false facts (h, r, t_*). We then use ChatGPT (OpenAI, 2023) to generate questions of $(h, r, ?)$ and form edit pair (x_e, y_e), as well as generate rephrased data for these questions, as shown below.

Question: What side effect is caused by Primaquine?
Rephrase: What adverse effect is attributed to Primaquine?
Ground Truth: Nausea. **Edit Target**: Stinging

Medical Fact Explanation Dataset. We build the MedFE dataset by utilizing MedMCQA (Pal et al., 2022), a dataset designed for answering medical entrance exam questions. To generate an edit pair (x_e, y_e), we combined the question and correct choice to form a factual statement, and we used the expert's explanation as a source for the target edit.

Fact: In obesity which of the following hormone levels is decreased? Adiponectin.
Rephrase: In cases of obesity, which hormone experiences a decrease in levels? Adiponectin.
Explanation: Adiponectin is an abundant adipose-derived protein and enhances insulin sensitivity and lipid oxidation. Its levels are reduced in obesity Obesity is associated with significant disturbances in endocrine function. Hyper insulinemia and insulin resistance are the best known changes in obesity. Thyroxine, GH, and adiponectin have lipolytic effects, hence their levels are reduced in obesity.

4.2 Locality Evaluation Metrics

The aforementioned data can be utilized to assess *Efficacy*, *Generality*, and *Fluency*. In terms of *Locality*, previous benchmarks have either employed out-of-distribution data (e.g., zsRE (Mitchell et al., 2022a)) or solely relied on data with the same ground truth (e.g., CounterFact (Meng et al., 2022a)). Nevertheless, we argue that a comprehensive evaluation of *Locality* is necessary to prevent the inadvertent modification of irrelevant knowledge and ensure high reliability of the medical domain. The post-edited model should be evaluated based on the following categories: (1) **Target Distribution**: Does the editing change the probability distribution of ground truth tokens? (2) **Entity Mapping**: Does the editing only learn the mapping relationship between head and tail entities? (3) **Structural Similarity**: Does the editing affect unrelated knowledge with similar structures? (4) **Textual Similarity**: Does the editing affect unrelated knowledge with similar text? (5) **Consistent Topic**: Does the editing affect unrelated knowledge with the same topic?

Based on these requirements, we collected corresponding data for Locality evaluation separately, which allows for a comprehensive analysis of the impact of model editing techniques on other knowledge within the medical domain. The data sampling for metrics (1), (2), and (5) can be achieved by simple retrieval. Metric (3) is attained by employing the knowledge graph embedding method (e.g., RotatE (Sun et al., 2018)) to learn embeddings of entities and relations, which measures similarity in terms of the graph structure. Metric (4) is achieved by employing BioBERT (Lee et al., 2020) to produce textual embeddings and compare similar question-answer pairs. Detailed samples can be found in Appendix B.

4.3 Experimental Setup

Metrics. we utilize the metrics constructed in Section 4.1 as our evaluation. The calculation of Fluency follows ROME (Meng et al., 2022a). The computation of other metrics follows EasyEdit (Wang et al., 2023a), which are measured as average accuracy between the token matching of the predicted output and the expected output. For ease of presentation, we employ abbreviations to represent each metric: *Efficacy* (Eff.), *Generality* (Gen.), *Locality* (Loc.), and *Fluency* (Flu.). For the sub-metrics of *Locality*, we use the following abbreviations: Target Distribution (TD), Entity Mapping (EM), Structural Similarity (SS), Textual Similarity (TS), and Consistent Topic (CT). Due to limitations of the original data, we measure TD, EM, SS, TS for MedCF and measure TS and CT for

Dataset	#Type	#Train	#Valid	#Test
MedCF	17	2,407	817	801
MedFE	9	2,533	851	841

Table 1: Statistics of MedCF and MedFE. The term 'Type' refers to the subject or relation types of datasets, such as 'Medicine' and 'Skin'. The number of samples for different types is kept consistent, further details are provided in the Appendix A.

MedFE. To examine the trade-off between edit success and locality, we further report the weighted mean by: $Avg. = (v_{Edit}+v_{Loc.})/2$, where $v_{Loc.} = \frac{1}{|Loc.|}\sum_{m\in Loc.} m$, and $v_{edit} = (Eff. + Gen.)/2$.

Backbones and Baselines. Due to the lack of datasets for medical model editing, all our experiments are conducted on MedCF and MedFE datasets, In our experiments, we focus on single-edit problem, and employ two LLMs that have been retrained in the medical domain as the to-be-edited models: ChatDoctor-13B (Yunxiang et al., 2023) and Meditron-7B (Chen et al., 2023). We compare various model editing methods, including FT-M (Fine-tuning on multiple layers), LoRA (Hu et al., 2021), ROME (Meng et al., 2022a), MEND (Mitchell et al., 2022a), and MEMIT (Meng et al., 2022b). All hyper-parameters are set according to optimal values in validation set of corresponding works. More details are shown in Appendix E.

4.4 Main Results

In this section, we present the main results compared with baselines. As indicated in Table 2, MedLaSA demonstrates significant improvements over all the baselines across most metrics. For instance, MedLaSA exhibits superior performance of *Fluency* on both datasets and two LLM backbones, validating our method's ability to maintain generation capability. The experimental results of FT-M indicate that excessive retraining of the parameters of an LLM could result in model collapse, causing the model to lose its original generating capability (i.e., much lower *Fluency*). LoRA introduces supplementary parameters but fails to consider the significant impact on unrelated knowledge. It also overlooks the specific positioning of knowledge in LLMs and the dependency of knowledge on different layers. As a result, when compared to MedLaSA, LoRA may have a similar level of editing success, but it performs poorly in terms of *Locality*. MEND has high requirements for initialization conditions and struggles to adapt to MedCF and MedFE datasets, resulting in lower average performance on these datasets. ROME focuses solely on single-layer knowledge editing of LLMs, without taking into account the knowledge stored in multi-layers, thus the performance tends to deteriorate. For the MedCF dataset, MEMIT is effective in editing counterfactual data by locating the key through the subject in the prompt and optimizing the value to select the object. This can improve factors such as *Efficacy* and *Generality*. However, MEMIT underperforms compared to MedLaSA in terms of *Locality*, especially for Entity Mapping. This is because MEMIT only learns the mapping relationship between the head and tail entities, leading to consistent predictions when the subject in the locality prompt is the same as the editing prompt. Furthermore, MEMIT's performance on the MedFE dataset is inferior due to its inability to handle long text output and complex multiple knowledge. MEMIT, which relies on subject-to-object localization, is not suitable for such scenarios. In contrast, our proposed MedLaSA addresses this issue by dynamically adjusting the scale of additional parameters and ensuring the insertion of complex knowledge.

4.5 Comparison of Strategies

In this section, we evaluate different strategies, including the Random and Fixed strategies, in comparison to our layer-wise scalable adapter strategy to assess their impact, as shown in Table 3. From Table 3, it is evident that our designed strategy outperforms both the Random and Fixed strategies across all metrics, which proves the effectiveness of MedLaSA. Moreover, Random strategy's performance is hindered by the unpredictability of parameter selection. This randomness leads to lower *Efficacy* and lower *Generality* compared to Fixed strategy. Despite these shortcomings, the Random strategy's varying scales for different knowledge and layers result in a lesser impact on irrelevant knowledge compared to the Fixed strategy, leading to higher scores in terms of *Locality* and *Fluency*.

4.6 Ablation Study

In this section, we analyze the effects on the performance of the model after removing Scaling Alpha (SA) and Scaling Rank (SR) in the self-attention (Attn) and MLP layers, as shown in Table 4. We can observe that the removal of SR leads to a decline in all metrics. Most notably, *Locality*-TS

Datasets	MedFE						MedCF							
Models	Eff.	Gen.	Loc. CT	Loc. TS	Flu.	Avg.	Eff.	Gen.	Loc. TD	Loc. EM	Loc. SS	Loc. TS	Flu.	Avg.
ChatDoctor-13B														
FT-M	61.39	61.04	73.09	70.78	516.44	66.57	61.55	61.48	60.74	63.02	59.66	58.74	356.31	61.03
LoRA	94.45	88.56	83.24	79.75	570.12	86.50	72.01	71.90	93.52	91.88	91.76	92.72	575.71	82.21
MEND	40.66	40.51	50.43	44.52	385.75	44.03	24.72	24.71	75.29	75.17	74.85	75.24	449.94	49.92
ROME	84.01	69.37	92.88	81.98	572.82	82.06	72.73	72.51	92.27	61.20	89.41	86.51	556.11	77.48
MEMIT	84.59	70.23	**95.80**	82.46	566.95	83.27	**82.20**	**82.03**	94.61	62.12	92.09	91.01	563.31	83.54
MedLaSA	**98.11**	**93.58**	89.25	**84.11**	**576.13**	**91.26**	72.37	70.80	**96.16**	**95.24**	**95.59**	**95.19**	**583.49**	**83.56**
Meditron-7B														
FT-M	62.82	62.68	67.62	64.94	473.66	64.51	65.97	65.36	48.91	50.39	48.13	46.25	327.76	57.04
LoRA	94.01	89.29	83.75	79.13	571.42	86.55	72.19	71.80	92.29	91.11	91.36	92.42	572.33	81.90
MEND	34.21	31.34	30.03	34.23	404.19	32.46	22.87	22.93	71.16	71.21	71.03	72.29	428.38	47.16
ROME	84.59	69.22	**95.78**	**86.44**	564.75	84.01	72.69	72.91	92.79	61.80	90.06	86.93	559.82	77.84
MEMIT	84.91	70.80	95.40	82.02	566.95	83.28	**83.10**	**83.23**	95.01	62.62	92.99	90.50	563.31	**84.22**
MedLaSA	**98.77**	**94.81**	87.41	81.67	**575.58**	**90.66**	72.37	71.06	**95.71**	**94.84**	**95.04**	**94.90**	**582.80**	83.42

Table 2: Model editing results compared with other state-of-the-art methods on MedCF and MedFE benchmarks. The best results are highlighted in bold, and larger values for all metrics indicate better performance. It should be noted that Locality metrics TD, EM, and SS necessitate source data that is structured in a knowledge graph format, thus can only be utilized for MedCF. Metric CT requires a more specific topic for the question, making it applicable only to MedFE.

Strategy	Eff.	Gen.	Loc. CT	Loc. TS	Flu.	Avg.
Random	92.15	86.98	85.62	80.11	572.76	86.22
Fixed	94.45	88.56	83.24	79.75	570.12	86.50
MedLaSA	98.77	94.81	87.41	81.67	575.58	90.66

Table 3: Comparison of different editing strategies on MedFE. The Random strategy involves randomly selecting the scale values of rank r_o and alpha α_o of all layers, instead of using casual tracing to determine knowledge location. The reported results were obtained through five random sampling experiments. The Fixed strategy maintains fixed scale values of rank r_o and alpha α_o to all data (factual knowledge), same with MedLaSA, across all layers.

	Eff.	Gen.	Loc. CT	Loc. TS	Flu.	Avg.
ALL	98.11	93.58	89.25	84.11	576.13	91.26
w/o SR	96.85	93.24	84.88	79.02	573.56	88.50
w/o SA	96.90	94.44	82.16	75.84	571.36	87.34
w/o SA&SR	94.45	88.56	83.24	79.75	570.12	86.50
w/o Attn	98.11	93.58	89.25	84.11	573.34	91.26
w/o Attn&SR	96.11	91.57	87.19	82.10	574.85	89.24
w/o Attn&SA	96.86	94.12	87.51	80.19	575.47	89.67
w/o MLP	92.94	85.11	88.18	84.17	578.41	87.60
w/o MLP&SR	90.96	84.13	86.19	82.20	577.30	85.87
w/o MLP&SA	96.05	91.07	84.57	79.27	576.13	87.74

Table 4: Ablation study on MedFE dataset. The abbreviation 'w/o' indicates that the following module is removed, and the term 'ALL' indicates that both scaling strategies are employed in both MLP and Attn layers.

experiences a decrease of approximately 5%. This suggests that SR plays a crucial role in maintaining the overall performance. On the other hand, when SA is removed, there is an improvement in the model's *Generality*. However, this improvement comes at the cost of a significant decrease in *Locality*-TS (e.g., from 84.11% reduced to 75.84%). This indicates that while SA helps in minimizing the model's modification of irrelevant knowledge, it concurrently compromises the model's generalization to rephrases. Similar results can also be observed when our proposed method is applied exclusively to Attn or MLP, which further demonstrates the effectiveness of SA and SR in medical model editing.

4.7 Hyper-parameters Analysis

In this section, we analyze the impact of different alpha α_o and rank r_o values on MedLaSA under the MedCF and MedFE datasets. The results of *Locality* are derived by averaging all its sub-metrics. As shown in Figure 3, we observe that as α_o increases, both *Efficacy* and *Generality* also increase. However, *Locality* decreases concurrently. This suggests that the size of α_o significantly influences the model's ability to successfully incorporate new knowledge and its impact on irrelevant knowledge. When selecting α_o, the trade-off between the factors must be considered, and the best average result

Weight Type	Editable Weight	Eff.	Gen.	Loc. CT	Loc. TS	Flu.	Avg.
Attn Weights	W_q, W_v	80.63	73.49	92.33	89.58	579.29	84.01
	W_k, W_o	78.65	71.76	94.71	91.88	582.62	84.25
	W_q, W_v, W_k, W_o	92.07	84.16	90.23	86.21	577.86	88.17
MLP Weights	W_{up}	82.62	76.30	96.47	93.40	580.81	87.20
	W_{down}	80.73	74.88	94.09	91.03	578.29	85.18
	$W_{up}, W_{down}, W_{gate}$	96.47	90.93	91.89	87.31	576.53	91.65
Attn + MLP	$W_q, W_v, W_{up}, W_{down}$	96.99	91.43	88.93	84.22	577.0	90.39
	$W_q, W_v, W_{up}, W_{down}, W_{gate}$	98.27	93.70	88.73	83.58	576.13	91.07
	$W_q, W_v, W_k, W_o, W_{up}, W_{down}, W_{gate}$	98.70	94.63	87.66	82.10	574.19	90.77

Table 5: Comparison of the impact of editing weights. The framework of ChatDoctor and Meditron are based on Llama (Touvron et al., 2023), which includes Attn weights (W_q, W_v, W_k, W_o) and MLP weights ($W_{up}, W_{down}, W_{gate}$).

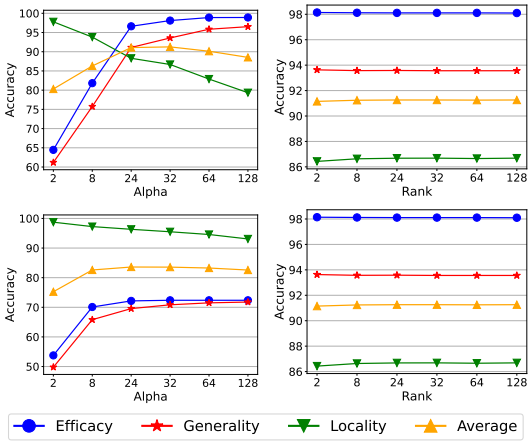

Figure 3: Analysis of hype-parameters r_o and α_o. The top two figures depict the results on MedFE, while the bottom two figures display the results on MedCF.

is achieved when α_o is set to 24. On the other hand, as the value of rank increases, there is no significant change in all metrics. Only when the rank is too small (e.g., equal to 2), does the model's editing *Locality* suffer certain negative effects, which indicates that for single-edit problems, the size of the rank is not a major determinant of the model's performance.

4.8 Comparison of Editing Weights

To analyze and evaluate the specific model weights that are more suitable for editing in MedLaSA, we conduct a comparison of editing weights, as shown in Table 5. it can be observed that for Attn weights, W_q and W_v demonstrate a higher editing success rate compared to weights W_k and W_o, and exhibit better generalization ability for rephrasing text. However, W_q and W_v show a poorer performance in terms of the *Locality* metrics. When all four weights are edited together, the learning of editing text is improved, but this improvement is accompanied by a decrease in the *Locality* metrics.

On the other hand, when comparing the weights of MLP, W_{up} consistently outperforms W_{down} in all metrics. This suggests that W_{up} may have the ability to retain more knowledge and is more suitable for editing medical models. Furthermore, when W_{up}, W_{down}, and W_{gate} are used together, there is a notable enhancement in both *Efficacy* and Average performance. It is worth noting that when both the Attn and MLP weights are simultaneously made trainable for editing, there are additional enhancements in *Efficacy* and *Generality*. However, this comes at the cost of significant decreases in *Locality* and *Fluency*. This suggests that by incorporating more trainable adapter parameters, the success rate of medical model editing increases. Consequently, it leads to a stronger impact on the prediction of irrelevant information.

5 Conclusion

In this paper, we focused on the editing of medical knowledge in LLMs and proposed two preliminary studies: editing factual medical knowledge and editing the explanations of LLMs. Two corresponding benchmarks were constructed to evaluate model editing methods, and more comprehensive and challenging metrics were proposed for *Locality* evaluation. What is more, we proposed MedLaSA to address the challenges faced in medical model editing due to the specialization and complexity of medical language. Extensive experiments conducted on MedCF and MedFE demonstrated the drawbacks of the existing methods and the outperformance of MedLaSA over them.

6 Limitations

There are several aspects to consider for both our datasets and our method in terms of limitations.

Regarding the MedCF and MedFE datasets, they consider different aspects of medical model editing and include a comprehensive evaluation of *Locality*. However, our proposed datasets do not consider more robust evaluations, such as portability (Yao et al., 2023) to assess whether editing was successful. Meanwhile, due to the constraints of the original data, the number of samples for medical model editing is relatively small, which does not support simultaneous editing on a larger scale.

As for our method, MedLaSA, although it effectively addresses the challenges of specialization and complexity of medical knowledge in model editing, it may have some negative impact on *Generality*. MedLaSA is primarily designed for single-edit cases and lacks considering batch editing (Meng et al., 2022b), which involves multiple edits (Huang et al., 2023) at once, or sequence editing, where models must retain previous modifications while implementing new ones. Furthermore, its performance on encyclopedic data (Mitchell et al., 2022a) remains to be explored.

7 Ethics Statement

The main objective of this paper is to propose two benchmarks and a novel medical model editing framework that aims to solve out-of-date and hallucinations problem in medical LLMs. It is important to note that our method does not produce uncontrollable outputs. On the contrary, the model editing method has the potential to enhance the controllability and reliability of medical LLMs. Additionally, the datasets and codes used in this study are constructed using publicly available data and tools, ensuring that there are no negative social consequences or ethical concerns.

References

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. MEDITRON-70B: Scaling Medical Pretraining for Large Language Models.

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, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing Factual Knowledge in Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506.

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*.

Zhangyin Feng, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. Trends in Integration of Knowledge and Large Language Models: A Survey and Taxonomy of Methods, Benchmarks, and Applications. *arXiv preprint arXiv:2311.05876*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45.

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. 2022. Aging with GRACE: Lifelong Model Editing with Discrete Key-Value Adaptors. In *NeurIPS 2022 Workshop on Robustness in Sequence Modeling*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

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*.

Vassilis N. Ioannidis, Xiang Song, Saurav Manchanda, Mufei Li, Xiaoqin Pan, Da Zheng, Xia Ning, Xiangxiang Zeng, and George Karypis. 2020. DRKG - Drug Repurposing Knowledge Graph for Covid-19. https://github.com/gnn4dr/DRKG/.

Mert Karabacak and Konstantinos Margetis. 2023. Embracing Large Language Models for Medical Applications: Opportunities and Challenges. *Cureus*, 15(5).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. PMET: Precise Model Editing in a Transformer. *arXiv preprint arXiv:2308.08742*.

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. 2022a. Fast Model Editing at Scale. In *International Conference on Learning Representations*.

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

OpenAI. 2023. GPT-4 Technical Report. *ArXiv*, abs/2303.08774.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. MedMCQA: A Large-scale Multi-subject Multi-choice Dataset for Medical domain Question Answering. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Judea Pearl. 2022. Direct and indirect effects. In *Probabilistic and causal inference: the works of Judea Pearl*, pages 373–392.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases? 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 2463–2473.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2018. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023a. EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models. *arXiv preprint arXiv:2308.07269*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2023b. Knowledge Editing for Large Language Models: A Survey.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. DEPN: Detecting and Editing Privacy Neurons in Pretrained Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886.

Derong Xu, Jingbo Zhou, Tong Xu, Yuan Xia, Ji Liu, Enhong Chen, and Dejing Dou. 2023. Multimodal Biological Knowledge Graph Completion via Triple Co-attention Mechanism. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3928–3941. IEEE.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing Large Language Models: Problems, Methods, and Opportunities. *arXiv preprint arXiv:2305.13172*.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2023. MELO: Enhancing Model Editing with Neuron-Indexed Dynamic LoRA. *arXiv preprint arXiv:2312.11795*.

Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge. *arXiv preprint arXiv:2303.14070*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024. A Comprehensive Study of Knowledge Editing for Large Language Models. *arXiv preprint arXiv:2401.01286*.

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? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

Hongjian Zhou, Boyang Gu, Xinyu Zou, Yiru Li, Sam S Chen, Peilin Zhou, Junling Liu, Yining Hua, Chengfeng Mao, Xian Wu, et al. 2023. A survey of large language models in medicine: Progress, application, and challenge. *arXiv preprint arXiv:2311.05112*.

A Knowledge Type

In this section, we provide a comprehensive overview of the knowledge types of MedCF and MedFE, as shown in Table 6 and 7, respectively. The knowledge types included in MedCF are categorized based on their relations, whereas the knowledge types in MedFE are categorized according to medical subjects.

B Data Samples

We provide data samples for the MedCF and MedFE datasets in Figure 4 and 5, comprising of the knowledge that requires editing, rephrase knowledge, and irrelevant knowledge obtained through our proposed methodology in Section 4.2.

C Templates

The data of *Efficacy* for MedCF and *Generality* for both MedCF and MedFE are generated by querying ChatGPT. The template for querying is shown in 8.

D Case of Casual Tracing

In this section, we showcase heatmaps illustrating individual instances of causal tracing from the MedCF and MedFE datasets on ChatDoctor network parameters (including Attn and MLP networks) in Figure 4 and 5. These heatmaps provide a visual representation of the causal tracing, allowing us to visually understand how the model makes decisions and pinpoint areas that can be enhanced. By examining these heatmaps, we can extract valuable information about how the model makes decisions and establish factual connections with particular neuron activations (Meng et al., 2022a).

E Hyper-parameters Selection

All experiments was conducted on four NVIDIA V100 32G. During our hyper-parameters search, we focused on finding the optimal hyperparameters for r, α, and editable weights W. The search scope for hyperparameters r and α was limited to the values [2, 8, 24, 32, 64, 128]. The analysis of r and α can be seen in Figure 3. The editable weights we considered were $W_q, W_v, W_k, W_o, W_{up}, W_{down}, W_{gate}$, which is compared in Table 5. We selected the parameter with the highest Average value as the best choice. The hyper-parameters we use are as follows:

- MedCF dataset on ChatDoctor-13B: r: 24, α: 32, learning rate: 1e-05, the number of steps: 70, the maximum length: 40, the editable weights: $W_q, W_v, W_{up}, W_{down}$.
- MedCF dataset on Meditron-7B: r: 24, α: 32, learning rate: 1e-05, the number of steps: 70, the maximum length: 40, the editable weights: $W_q, W_v, W_k, W_o, W_{up}, W_{down}, W_{gate}$.
- MedFE dataset on ChatDoctor-13B: r: 24, α: 32, learning rate: 5e-05, the number of steps: 40, the maximum length: 200, the editable weights: W_{up}, W_{down}.
- MedFE dataset on Meditron-7B: r: 24, α: 24, learning rate: 5e-05, the number of steps: 40, the maximum length: 200, the editable weights: $W_q, W_v, W_k, W_o, W_{up}, W_{down}, W_{gate}$.

Entity type	Relation type	#Samples
Compound:Compound	drug-drug interaction	255
Compound:Disease	compound treats the disease	256
Compound:Disease	inhibits cell growth (esp. cancers)	256
Compound:Disease	role in disease pathogenesis	256
Compound:Disease	biomarkers (of disease progression)	256
Compound:Disease	alleviates, reduces	256
Compound:Disease	prevents, suppresses	256
Compound:Disease	side effect/adverse event	256
Compound:Disease	treatment/therapy (including investigatory)	256
Compound:Side Effect	causes	160
Compound:Disease	palliation	256
Compound:Compound	resemblence	256
Compound:Disease	treatment	256
Anatomy:Disease	localization	133
Disease:Symptom	presents	152
Diisease:Disease	resemblence	256
Compound:Pharmacologic Class	inclusion	256

Table 6: Knowledge type of MedCF.


```
{
    "prompt": "What side effect is caused by Primaquine?",
    "target_new": "Stinging",
    "subject": "Primaquine",
    "ground_truth": "Nausea",
    "rephrase_prompt": "What adverse effect is attributed to Primaquine?",
    "locality_target_prompt": "What side effect is caused by Orciprenaline?",
    "locality_target_ground_truth": "Nausea",
    "locality_mapping_prompt": "Which drug is involved in a drug-drug interaction with Primaquine?",
    "locality_mapping_ground_truth": "Lobeglitazone",
    "locality_struc_prompt": "What side effect is caused by Domperidone?",
    "locality_struc_ground_truth": "Breast pain",
    "locality_tokenSem_prompt": "What disease does Primaquine treat?",
    "locality_tokenSem_ground_truth": "PJP"
}
```


Figure 4: A sample of the MedCF dataset.

Knowledge Type	#Samples
Pharmacology	510
Ear, nose, and throat	510
Medicine	505
Pediatrics	509
Orthopaedics	503
Pathology	508
Gynaecology & Obstetrics	509
Ophthalmology	510
Skin	164

Table 7: Knowledge type of MedFE.

Template
Efficacy: Given a triplet ($[e_h]$, $[r]$, $[e_t]$), please express this triplet in a question-answer form. The $[e_h]$ and $[r]$ form a question, and you need to ask what the corresponding answer $[e_t]$ is for this question. ***Generality***: Please rephrase the question using medical terminology, without changing the semantics: q

Table 8: Templates for querying ChatGPT to generate the *Efficacy* and *Generality* data. e_h, r, and e_t denote the head entity, relation, and tail entity, respectively.

{
"subject": "Adiponectin",
"subject_type": "Medicine",
"topic_type": "Diabetes Mellitus",
"prompt": "Please provide an explanation for the following fact: \n In obesity which of the following hormone levels is decreased ? Adiponectin",
"target_new": "Adiponectin is an abundant adipose-derived protein and enhances insulin sensitivity and lipid oxidation .Its levels are reduced in obesity Obesity is associated to significant disturbances in endocrine function. Hyper insulinemia and insulin resistance are the best known changes in obesity. Thyroxine, GH and adiponectin have lipolytic effects, hence their levels are reduced in obesity",
"rephrase_prompt": "Please provide an explanation for the following fact: \n In cases of obesity, which hormone experiences a decrease in levels? Adiponectin.",
"locality_topic_prompt": "Please provide an explanation for the following fact: \n Which type of Insulin is used to manage a case of Diabetic ketoacidosis? Regular",
"locality_topic_ground_truth": "Drug of choice for DKA - IV/ SC regular insulin IV NS 5L/ 24 hours Bed time Insulin - Glargine IV Insulin - 1. Acute Hyperkalemia 2. DKA 3. Hyperosmolar coma SC Insulin - DMI",
"locality_tokenSem_prompt": "Please provide an explanation for the following fact: \n Which protein secreted by adipocytes prevents obesity? Leptin",
"locality_tokenSem_ground_truth": "Leptin is a peptide produced by the ob gene; Its name derived from the Greek root leptos, meaning thinLeptin is secreted by adipose cells and acts primarily through the hypothalamusHigh leptin levels decrease food intake and increase energy expenditure, thereby preventing obesity"
}

Figure 5: A sample of the MedFE dataset.

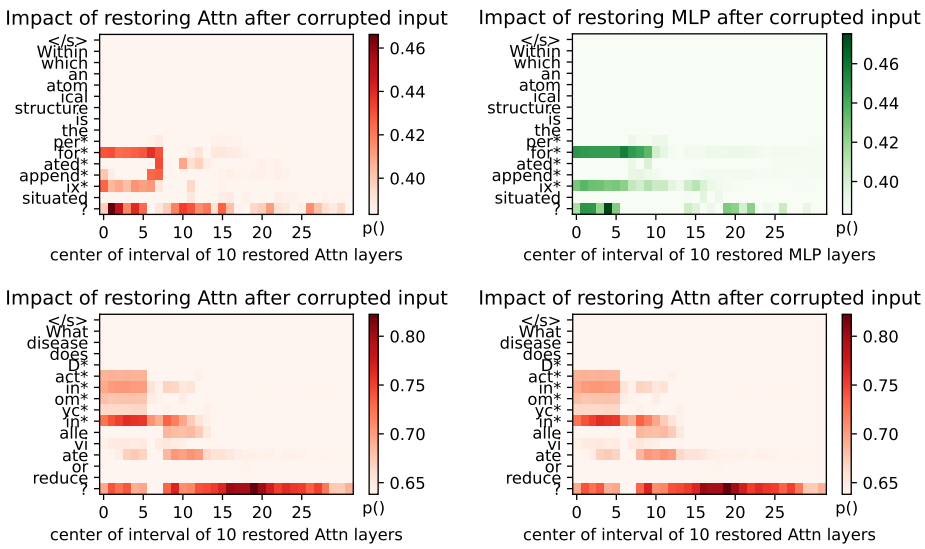

Figure 6: Case of casual tracing on the MedCF dataset.

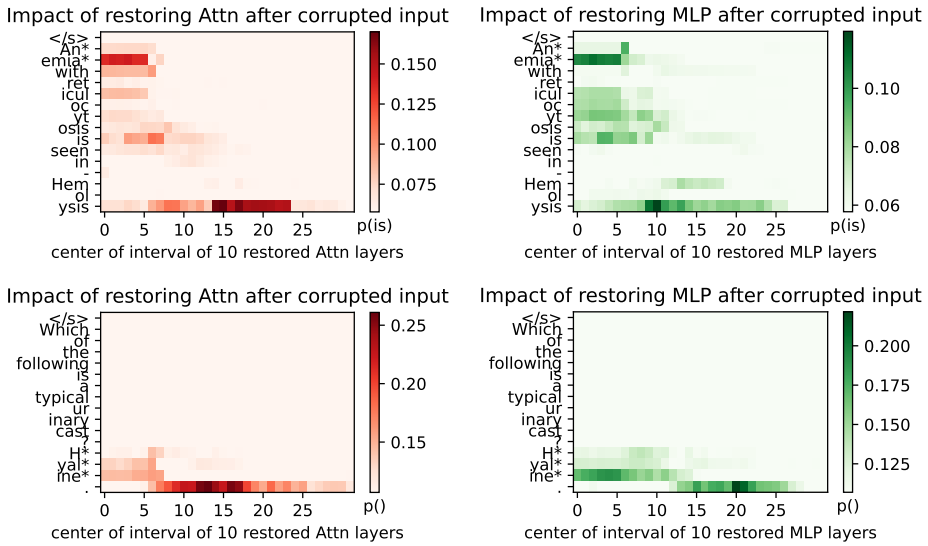

Figure 7: Case of casual tracing on the MedFE dataset.