Latent Knowledge Scalpel: Precise and Massive Knowledge Editing for Large Language Models

Xin Liu[1,2], Qiyang Song[1,2], Shaowen Xu[1,2], Kerou Zhou[3], Wenbo Jiang[4], Xiaoqi Jia[1,2], Weijuan Zhang[1,2], Heqing Huang, Yakai Li[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Tsinghua University
[4]University of Electronic Science and Technology of China
liuxin235@mails.ucas.ac.cn, {songqiyang, xushaowen, jiaxiaoqi, zhangweijuan, liyakai}@iie.ac.cn,
zkr22@mails.tsinghua.edu.cn, wenbo_jiang@uestc.edu.cn, Heqing.state@gmail.com

Abstract

Large Language Models (LLMs) often retain inaccurate or outdated information from pre-training, leading to incorrect predictions or biased outputs during inference. While existing model editing methods can address this challenge, they struggle with editing large amounts of factual information simultaneously and may compromise the general capabilities of the models. In this paper, our empirical study demonstrates that it is feasible to edit the internal representations of LLMs and replace the entities in a manner similar to editing natural language inputs. Based on this insight, we introduce the Latent Knowledge Scalpel (LKS), an LLM editor that manipulates the latent knowledge of specific entities via a hypernetwork to enable precise and large-scale editing. Experiments conducted on Llama-2 and Mistral show that even with the number of simultaneous edits reaching 10,000, LKS effectively performs knowledge editing while preserving the general abilities of the edited LLMs.

1 Introduction

The development of large language models (LLMs) has significantly advanced natural language processing (NLP) (Qin et al., 2024). However, challenges such as hallucinations (Huang et al., 2024; Xu et al., 2024), biases (Gallegos et al., 2024), and outdated information (Lazaridou et al., 2024) persist after pre-training. Therefore, it is essential to perform targeted updates to this incorrect or outdated information that arises during the deployment of LLMs.

Retraining or fine-tuning (Wei et al., 2022) can address this issue but requires substantial computational resources and time. Parameter-efficient fine-tuning (PEFT) methods (Lialin et al., 2024) provide more efficient alternatives, though they may lead to overfitting and are limited in reliability (Wang et al., 2024b; De Cao et al., 2021). Another class of methods modifies the behavior of LLMs by adding contextual information to the prompts, including prompt engineering (Sahoo et al., 2024) and retrieval-augmented generation (RAG) (Lewis et al., 2020). However, these methods may fail due to misalignment between LLMs and prompts (Hernandez et al., 2024). Moreover, they are constrained by prompt length, as they require ample context to be effective (Wang et al., 2024b).

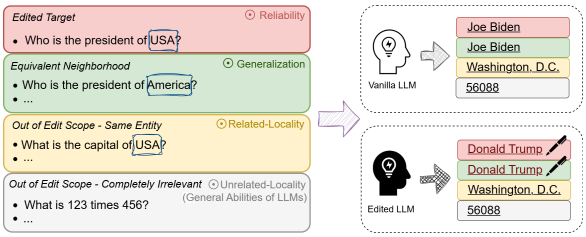

Figure 1: Illustration of model editing. Model editing modifies specific knowledge with minimal impact on unrelated inputs.

Model editing has emerged as a promising solution, aiming to make targeted modifications to specific model behaviors while minimizing changes to unrelated distributions, as shown in Figure 1. While previous works have introduced various enlightening editing approaches, there remains room for improvement. Gu et al. (2024) highlights that editing methods that modify model weights, such as Dai et al. (2022), Mitchell et al. (2022a), Meng et al. (2023a), and Meng et al. (2023b), can lead to overfitting on the edited facts, degrading the model's general abilities. Furthermore, methods such as De Cao et al. (2021), Dai et al. (2022), Mitchell et al. (2022a), and Meng et al. (2023a) become less effective when editing large volumes of factual information simultaneously (Mitchell et al., 2022a; Meng et al., 2023b). Hartvigsen et al. (2023) directly replaces the hidden states of the original model with the edit target to enable lifelong sequential editing, but it suffers from poor generalization and often fails to edit paraphrases of the targets.

In this paper, we propose Latent Knowledge Scalpel (LKS), an LLM editor capable of performing large-scale simultaneous knowledge editing without compromising the general abilities of LLMs. Unlike methods that modify the model's weights, we focus on editing the internal representations of specific entities. Previous studies (Petroni et al., 2019; Jiang et al., 2020; Li et al., 2021; Sun et al., 2024) have shown that the internal representations (or hidden states) of LLMs contain both factual knowledge and contextual information. For fine-grained editing, we associate knowledge with entities, which represent the smallest unit of knowledge in natural language (Cao et al., 2021). Our empirical study (§2) demonstrates that the internal representation of a single entity encapsulates both factual knowledge and semantic features, which we refer to as a **knowledge block (KB)**. Moreover, we show that the internal representations of LLMs preserve the syntactic structure of natural language, allowing operations similar to those on natural language itself.

Building on these findings, LKS manipulates specific entity latent knowledge for targeted updates (§3). During inference, if the input contains an entity within the edit scope, LKS uses a simple neural network to generate a new knowledge block (KB) for this entity and replace the original one, guiding the LLM to produce the desired output. This network is trained to integrate the new knowledge of entities within the edit scope, enabling it to generate optimal KBs. These KBs update specific entity features while preserving others, ensuring precise edits. Moreover, the use of the neural network allows LKS to handle large-scale, simultaneous updates. Our entity recognition mechanism ensures accurate identification of the edit scope, preventing LKS from triggering on inputs outside the scope, thereby enabling extensive edits without affecting unrelated distributions.

We conduct extensive experiments to evaluate our LKS editor (§4). Our experimental results demonstrate that LKS outperforms six other methods in factual knowledge editing on Llama-2-7B and Mistral-7B, achieving the best balance in reliability, generality, and locality. Additionally, during large-scale simultaneous editing, LKS can accurately perform 10,000 edits simultaneously, achieving high edit performance while maintaining the general abilities of the LLMs.

We make the following key contributions:

1. We introduce Latent Knowledge Scalpel (LKS), an LLM editor that replaces entity knowledge blocks with new ones generated by a simple neural network, achieving targeted and large-scale LLM editing while preserving the general abilities of LLMs.

2. We demonstrate that the entity knowledge blocks in LLMs contain semantic information, and the internal representations of LLMs retain the syntactic structure of natural language, allowing us to manipulate them like natural language.

3. Our experiments show that even when the number of simultaneous edits reaches 10,000, LKS is still able to maintain the general abilities of the edited LLMs while outperforming other editors in terms of edit performance.

2 Empirical Study

2.1 Semantic Information of a Single Entity Knowledge Block

In natural language, an entity typically contains multiple factual knowledge. For example, a person entity may include information such as age, occupation, and hobbies. This raises the question: does a single entity knowledge block from a LLM also contain sufficient semantic information?

To investigate this, we design a probe to differentiate between factual knowledge learned by the LLM and counterfactual knowledge it has not encountered. Specifically, we extract 10,000 entities along with their factual and counterfactual attributes from the Counterfact dataset (Meng et al., 2023a). The probe computes the cosine similarity between the entity KB and the internal representations of the last tokens from both factual and counterfactual knowledge, selecting the one with the higher similarity:

$$\underset{knowledge \in \mathcal{K}}{argmax} \; cosine\text{-}similarity(R_{entity}, R_{knowledge}) \tag{1}$$

where $\mathcal{K}$ contains both factual and counterfactual knowledge and R denotes internal representation. The probe's accuracy is defined as the proportion of correctly selected factual knowledge. Higher accuracy indicates that the entity KB is semantically closer to learned knowledge, suggesting it encodes meaningful semantic information.

Figure 2 presents the probe's accuracy across layers in Llama-2-7B-Chat (Touvron et al., 2023) and

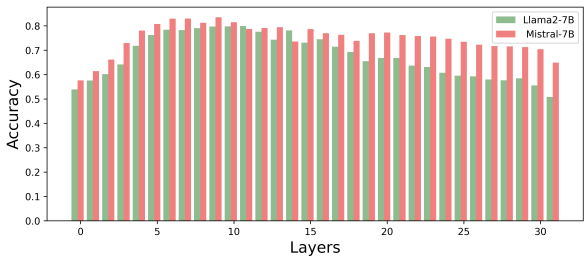

Figure 2: Probe accuracy for identifying factual knowledge across layers in Llama-2-7B and Mistral-7B. The results show that the probe accuracy exceeding 50% on average and peaking at 80%, demonstrating that a single entity KB retains semantic information.

Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). The probe achieves an average accuracy above 50%, surpassing random guessing, with peak accuracy reaching 80%. These results confirm that a single entity KB in a LLM retains its semantic information.

2.2 Syntactic Structure of Internal Representations

Natural language follows a syntactic structure, and replacing an entity name in a natural language prompt shifts the LLM's prediction toward the semantics of the new entity. Our research shows that the internal representations of LLMs exhibit a similar syntactic structure, as illustrated in Figure 3.

To investigate this, we use the template "`The birthplace of Alfred Bernhard Nobel is`" and replace the KB of "`Alfred Bernhard Nobel`" with different entity KBs. We then measure the rate at which the predicted birthplaces rank higher after replacement. The results in Figure 4 show that replacing KBs increases the ranking of the target location across all layers in both Llama-2-7B and Mistral-7B. Additionally, the effect diminishes as the layer number increases.

These findings confirm that LLMs' internal representations preserve syntactic structure to some extent. Furthermore, they suggest that during forward propagation, unchanged parts of the internal representation continue to influence predictions, explaining why the effect of KB replacement is stronger in earlier layers. If the goal is to introduce new information while preserving some original knowledge, modifying KBs in intermediate layers may be more effective.

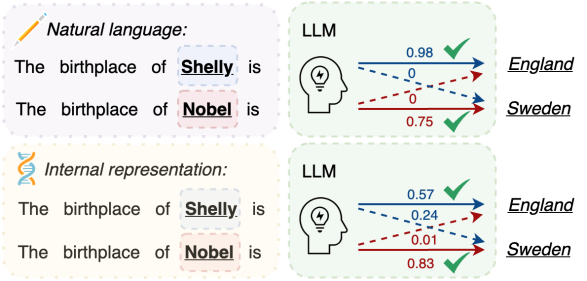

Figure 3: Upper: In natural language, replacing the entity "Shelly" with "Nobel" in the context of the "birthplace" causes the prediction from Llama-2-7B shifting from "England" to "Sweden". Lower: In internal representation, by obtaining the internal representations of two sentences and swapping the entity KB at a certain layer, similar to replacing entity names in a natural language prompt, the prediction of LLM changes and outputs the corresponding birthplaces.

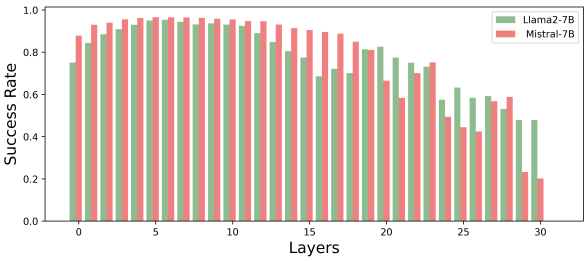

Figure 4: By replacing the name KB in the template with different entity KBs at each layer of Llama-2-7B and Mistral-7B, an increase in the ranking of the target birthplace across all layers in both models can be observed, confirming that internal representations of LLMs retain syntactic structure.

3 Method

3.1 Overview of LKS

Design Goal We aim to design an LLM editor that can effectively modify large-scale knowledge simultaneously while preserving the general abilities of LLMs. Particularly, it should satisfy the following requirements for LLM editing:

- Reliability: Accurately updates the specified targets.
- Generality: Consistently updates the equivalent neighborhoods of the specified targets.
- Locality: Ensures that knowledge outside the edit scope remains intact.

We propose Latent Knowledge Scalpel (LKS), an LLM editor that precisely updates the latent knowledge of LLMs using a hypernetwork. We extract entity-related knowledge from an LLM, con-

struct a self-supervised training dataset, and train a simple neural network (linear or MLP) specialized in entity-related knowledge. The new entity knowledge block (KB) generated by the network replaces the original one in the LLM. This updated entity KB is integrated into the LLM's forward propagation, guiding the model to produce the edited target within the edit scope while preserving its original predictions outside this scope.

The architecture of LKS is shown in Figure 5. LKS consists of three components: **Edit Scope Indicator**, which determines if an entity in the prompt falls within the edit scope, using fuzzy string matching and Levenshtein distance; **New KB Generator**, a simple neural network that generates the updated entity KB, which can either be a linear layer or an MLP layer. It is trained on a dataset containing the latest knowledge of entities within the edit scope, enabling it to output the optimal new entity KB; and **KB Replacer**, which hooks into a selected layer (discussed in detail in Section 4.3) of the edited LLM and replaces the original entity KB with the new one generated by the New KB Generator. The updated entity KB is then involved in the LLM's forward propagation, ultimately guiding the model's prediction.

If the Edit Scope Indicator determines that the prompt contains the entity to be edited, the New KB Generator generates the updated entity KB for that entity. The KB Replacer then replaces the original entity KB in the selected layer, and the inference process continues until the edited LLM's prediction is obtained. Otherwise, the last two components are not triggered, and the original model proceeds with the inference as usual.

3.2 Building a New Knowledge Block

LKS enables LLMs to generate updated predictions for inputs within the edit scope (target edits and their equivalent neighborhoods) while preserving predictions outside this scope. In other words, it selectively edits a semantic feature of an entity while maintaining unrelated content. To achieve this, we construct a new knowledge block in three steps, as illustrated in Figure 6.

Knowledge Extraction Inspired by Zhou et al. (2023), we extract text-based entity-related knowledge from the LLMs. For each entity, we use GPT-4o mini (OpenAI et al., 2024) to generate multiple sentences reflecting its factual knowledge.

Knowledge Updating We replace the factual knowledge of the target feature and its equivalent

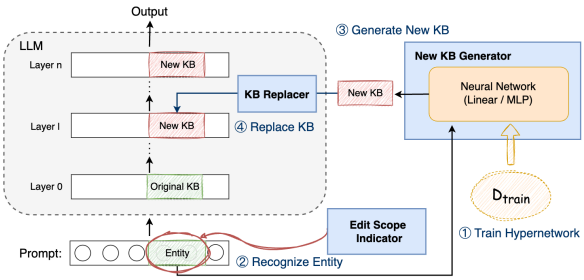

Figure 5: Architecture and Process of LKS. ① A simple neural network is trained using $\mathcal{D}_{train}$ to generate the optimal new KB during inference. ② Upon receiving a prompt, the Edit Scope Indicator checks if the target entity is present. If so, the relevant information is passed to the New KB Generator; otherwise, the original LLM proceeds as usual. ③ The New KB Generator then creates the updated entity KB. ④ The KB Replacer updates the corresponding entity KB in the selected layer l, and the inference continues to produce the final edited prediction.

neighborhood with the desired content, while leaving other entity features unchanged. These unchanged features will be aligned with the relevant knowledge in the edited LLM during the next step.

Knowledge Compression Following prior works (Petroni et al., 2019; Shin et al., 2020; Roberts et al., 2020; Onoe et al., 2022; Abaho et al., 2022; Chen et al., 2022; Youssef et al., 2023), we convert the extracted and updated entity knowledge into gap-filling prompts to create a self-supervised training dataset $\mathcal{D}_{train}$. A simple neural network is then trained on $\mathcal{D}_{train}$, serving as a hypernetwork to generate new entity KBs that replace the original ones in the LLM. During training, the LLM aligns its predictions with the updated targets while retaining non-edited knowledge. After training, this neural network encapsulates only the latest entity knowledge and can produce the optimal new entity KBs which represent the compressed knowledge.

3.3 Training LKS Hypernetwork

The neural network $h_\phi(\cdot)$ takes the input entity E and outputs the new knowledge block for layer l, denoted as $\tilde{R}^l_\phi(E) = h_\phi(E; l)$. This hypernetwork is trained using $\mathcal{D}_{train}$ in advance to generate the optimal new KB $\tilde{R}^l$ during inference. During LLM inference, LKS replaces the original KB R^l with the new KB $\tilde{R}^l$, guiding the LLM's predictions. Notably, $\mathcal{D}_{train}$ is significantly smaller than the original LLM training dataset, and the storage overhead of the neural network is negligible compared to the LLM itself. For instance, h_ϕ with a

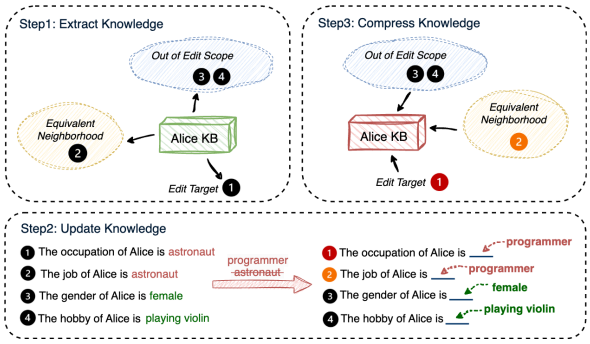

Figure 6: The process of building a new KB. ① Extract entity knowledge from a LLM. ② Update the target knowledge for editing the entity. ③ Compress the knowledge using a simple neural network contains only the latest knowledge of entities within the edit scope.

linear layer for Llama-2-7B occupies only 64MB, regardless of the number of edits it contains.

Given an LLM f_θ and an input sequence x containing entity E, the model recalls the corresponding feature of E and predicts the token sequence y. The original entity KB in layer l can be formulated as $R_\theta^l(E) = R_\theta^{l-1}(E) + attn_\theta^l(E) + mlp_\theta^l(E)$. The output y can be expressed as $y = f_\theta(x, R_\theta^l(E))$. For factual knowledge editing, LKS replaces the original entity KB at layer l with $\tilde{R}_\phi^l(E)$, enabling the LLM to generate a new prediction $\tilde{y}$ aligned with the updated feature: $\tilde{y} = f_\theta(x, \tilde{R}_\phi^l(E))$. The neural network h_ϕ is optimized using the following loss function:

$$\mathcal{L}(\phi) = \lambda_{edit}(\mathcal{L}_{edit} + \mathcal{L}_{eq}) + \mathcal{L}_{locality} \tag{2}$$

$\mathcal{L}_{edit}$ is optimized via maximum likelihood estimation, ensuring that the prompt $\mathbb{X}_e$ describing the edit aligns with the target $\mathbb{Y}_e$, leading to correct updates within the edit scope:

$$\mathcal{L}_{edit} = -\log p(y_e | x_e, \tilde{R}_\phi^l(E)), \quad (x_e, y_e) \in (\mathbb{X}_e, \mathbb{Y}_e) \tag{3}$$

Similar to $\mathcal{L}_{edit}$, $\mathcal{L}_{eq}$ ensures that equivalent neighborhood inputs $\mathbb{X}_{eq}$ result in the same $\mathbb{Y}_e$:

$$\mathcal{L}_{eq} = -\log p(y_e | x_{eq}, \tilde{R}_\phi^l(E)), \quad (x_{eq}, y_e) \in (\mathbb{X}_{eq}, \mathbb{Y}_e) \tag{4}$$

$\mathcal{L}_{locality}$ constrains the logit distribution for unrelated features $\mathbb{X}_{loc}$ using Kullback-Leibler (KL) divergence, minimizing deviations from the original pre-trained logit distribution. This ensures that the original distribution remains unchanged outside the edit scope:

$$\mathcal{L}_{locality} = KL(p(\cdot|x, \tilde{R}_\phi^l(E)), p(\cdot|x, R_\theta^l(E))), \quad x \in \mathbb{X}_{loc} \tag{5}$$

Algorithm 1 Training Algorithm of LKS

Input: Training dataset $\mathcal{D}_{train}$; LLM f_θ; LKS neutral network h_ϕ; Edit layer l; hyperparameter λ_{edit}
Output: Trained LKS neutral network h_ϕ; Edit scope $\mathcal{S}$

Generate the edit scope $\mathcal{S}$ according to $\mathcal{D}_{train}$;
While not early-stopping do
  Sample entity $E, x_e, y_e, x_{eq}, x_{loc}$ from $\mathcal{D}_{train}$;
  $\mathcal{L}_{edit} = -logp(y_e|x_e,\ \tilde{R}_\phi^l(E))$;
  $\mathcal{L}_{eq} = -logp(y_e|x_{eq},\ \tilde{R}_\phi^l(E))$;
  $\mathcal{L}_{loc} = \mathrm{KL}(p(\cdot|x,\ \tilde{R}_\phi^l(E)),\ p(\cdot|x,\ R_\theta^l(E)))$;
  $\mathcal{L}(\phi) = \lambda_{edit}(\mathcal{L}_{edit} + \mathcal{L}_{eq}) + \mathcal{L}_{locality}$;
  $\phi \leftarrow \mathrm{AdamW}(\phi, \nabla\mathcal{L}(\phi))$;

Algorithm 2 Inference Algorithm of LKS

Input: LLM f_θ; Trained LKS neutral network h_ϕ; Edit scope $\mathcal{S}$; Input prompt x
Output: Prediction $\hat{y}$

If $\exists E \in x,\ E \in \mathcal{S}$:
  # Edit with LKS
  Replace $R_\theta^l(E)$ using $\tilde{R}_\phi^l(E)$;
  $\hat{y} = f_\theta(x,\ \tilde{R}_\phi^l(E))$;
Else:
  # Do not edit, output as origin
  $\hat{y} = f_\theta(x)$;
return $\hat{y}$;

See Algorithm 1 and Algorithm 2 for a detailed overview of LKS training and inference. For hyperparameter details, refer to Appendix A.

4 Experiments

4.1 Experiment Setting

Datasets For evaluating the reliability, generality, and related-locality of factual editing, we generate two evaluation datasets using GPT-4o mini based on the zsRE question-answering dataset (Levy et al., 2017) and the Counterfact dataset (Meng et al., 2023a). Details can be found in Appendix B. For unrelated-locality, we use GSM8K (Cobbe et al., 2021), RTE (Dagan et al., 2005), and SST2 (Socher et al., 2013) to assess the general abilities of the edited LLMs. GSM8K tests the model's mathematical reasoning ability, RTE assesses its natural language inference ability (i.e., whether a statement is reasonable), and SST2 evaluates sentiment analysis capabilities by classifying statements as positive or negative.

Baselines We use several classical or effective model editing methods as baselines. MEND (Mitchell et al., 2022a) edits models by updating MLP layer weights using the low-rank structure of fine-tuning gradients. ROME (Meng et al., 2023a) and MEMIT (Meng et al., 2023b) modify specific factual associations by adjusting MLP weights, with MEMIT supporting large-scale edits.

GRACE (Hartvigsen et al., 2023) records model hidden states in a codebook and replaces the original states during edits. WISE (Wang et al., 2024a) introduces a dual parametric memory mechanism, with a main memory for pretrained knowledge and a side memory exclusively for edits. AlphaEdit (Fang et al., 2025) attempts to preserve original knowledge by projecting weight updates onto the null space of preserved facts. All baselines are evaluated using EasyEdit (Wang et al., 2024b), an easy-to-use framework for LLM knowledge editing, ensuring convenient and fair assessment.

4.2 Evaluation Metrics

Following prior works (Mitchell et al., 2022a,b; Meng et al., 2023a), we evaluate LLM editing performance using three primary metrics: reliability, generality, and locality. As shown in Figure 1, these metrics assess the model's behavior for prompts inside and outside the edit scope.

For **reliability** and **generality**, computing the average exact-match accuracy between the edited predictions and the target outputs within the edit scope:

$$\text{Rel} = \mathbb{E}(\mathbb{1}_{f_{LKS}(x_e)=y_e}) \tag{6}$$

$$\text{Gen} = \mathbb{E}(\mathbb{1}_{f_{LKS}(x_{eq})=y_e}) \tag{7}$$

For locality, we further divide it into two categories: related-locality, which pertains to areas related to the edited entity but not the modified feature, and unrelated-locality, which refers to areas completely outside the edit scope. In other words, unrelated-locality means that after performing factual edits, the general abilities of LLMs, such as mathematical reasoning and sentiment analysis, should remain unchanged.

For **related-locality**, we measure whether predictions for inputs which are related to the edited entity but outside the edit scope remain unchanged:

$$\text{Loc} = \mathbb{E}(\mathbb{1}_{f_{LKS}(x_{loc})=f(x_{loc})}) \tag{8}$$

We define **Edit Performance** (EP) as the average of reliability, generality, and related-locality, providing a comprehensive evaluation of editing effectiveness.

For **unrelated-locality**, we assess how well the edited LLM preserves the general abilities of its original model, including mathematical reasoning, natural language inference, and sentiment analysis.

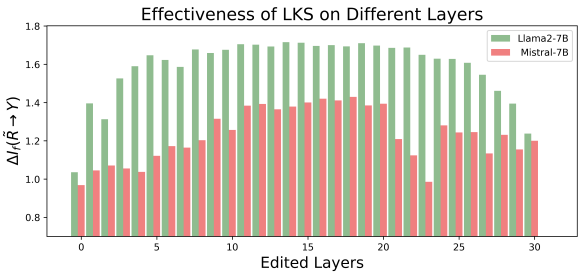

Figure 7: Effectiveness of LKS on different layers, measured by the information gain $\Delta I_f(\tilde{R} \to Y)$. Positive values indicate that the new KBs increases the likelihood of the LLM generating output Y. Results show that modifying intermediate layers of Llama-2-7B and Mistral-7B leads to higher effectiveness.

4.3 Selection of the LKS Operating Layer

LKS achieves LLM editing by replacing the entity knowledge blocks. This section applies information theory to validate its effectiveness and guide the selection of the optimal layer for replacement.

Inspired by Shannon Information Theory (Shannon, 1948) and Ethayarajh et al. (2022), we define the information gain $\Delta I_f(\tilde{R} \to Y)$ to measure how effectively the new knowledge block $\tilde{R}$ helps model f generate output Y. A positive $\Delta I_f(\tilde{R} \to Y)$ indicates that the new KB outperforms the original in generating Y. The larger the value, the more effective the new KB. Using the entropy definition, the information entropy $H_f(Y|R)$ required for model f to predict Y given KB R is $H_f(Y|R) = \inf \mathbb{E}[-\log_2 f[R](Y)]$.

Thus, $\Delta I_f(\tilde{R} \to Y)$ can be calculated as:

$$\Delta I_f(\tilde{R} \to Y) = H_f(Y|R) - H_f(Y|\tilde{R}) \tag{9}$$

The results in Figure 7 show positive values of $\Delta I_f(\tilde{R} \to Y)$, indicating that the modification of the entity KBs increases the likelihood of the LLM generating the edit targets Y. Modifying intermediate layers yields higher effectiveness, and although modifying multiple layers is possible, we opt for a single layer to balance computational cost. In subsequent experiments, we select layer 16 of Llama-2-7B and layer 18 of Mistral-7B for the LKS replacement.

4.4 Edit Performance of Large-Scale Simultaneous Editing

In many scenarios, large-scale and simultaneous edits are necessary for LLMs. For example, updating thousands of factual changes within a specific time frame, or removing large amounts of erroneous or privacy-sensitive information introduced during

pre-training. In such cases, allowing only one edit at a time is insufficient.

In this section, we evaluate the edit performance of various model editing baselines on the zsRE dataset using Llama-2-7B and Mistral-7B under different numbers of edits. The number of simultaneous edits T ranges from a single edit to a large-scale setting of 10,000 edits.

As shown in Table 1, LKS outperforms all other methods, achieving the highest EP scores on both LLMs across almost all edit numbers T. This demonstrates that LKS delivers the best performance both within and outside the editing range. Specifically, LKS effectively modifies the target features of entities while preserving unrelated features, ensuring highly targeted edits. The effectiveness is driven by the LKS neural network, which learns to accurately update the target features and their equivalent neighborhoods. Related-locality is maintained through two mechanisms: first, the Edit Scope Indicator identifies whether the inputs contain entities within the edit scope, and second, the New KB Generator is trained to preserve unrelated distributions as much as possible.

Moreover, as the number of simultaneous edits increases up to 10,000, LKS still achieves and maintains the best performance. Its reliability and generality remain high, although locality experiences a slight decline as the number of edits grows. In contrast, the performance metrics of other baselines show significant degradation. This suggests that LKS's neural network effectively stores the updated factual knowledge, enabling massive simultaneous and precise edits.

Furthermore, the LLMs edited by LKS demonstrate consistently fluent text generation. Details can be found in Appendix D.

4.5 Maintaining the General Abilities of LLMs after Editing

If the general abilities of the edited LLMs are compromised or rendered ineffective, LLM editing would become counterproductive. In this section, we evaluate four methods with superior edit performance as identified in §4.4 (MEMIT, WISE, AlphaEdit, and LKS), testing whether their simultaneous multiple edits come at the cost of damaging the general abilities of the edited LLMs. Here, we use the GSM8K, SST2, and RTE datasets to evaluate how effectively the edited LLM preserves the general abilities of its original model. These three datasets assess the LLM's capacities in mathematical reasoning, sentiment analysis, and natural language inference, respectively.

The results shown in Figure 8 indicate that when simultaneously editing thousands of facts, both MEMIT and AlphaEdit lead to substantial degradation across all three capability metrics of the edited LLMs, indicating a severe compromise of their general abilities. The Llama-2 model edited by WISE demonstrates unstable performance on general tasks, and its edits on Mistral-7B clearly fail to preserve the model's original general capabilities. In contrast, as the number of simultaneous edits increases, LLMs edited by LKS exhibit stable performance without noticeable degradation. Even with 10,000 edits, LKS retains nearly all of the original LLM's general abilities.

5 Related Work

Knowledge in Language Models Language models (LMs) can acquire vast amounts of factual knowledge during pre-training (Petroni et al., 2019; Jiang et al., 2020; Sun et al., 2024). Studies using manually or automatically generated prompts have demonstrated that LMs store intrinsic memories within their pre-trained parameters (Petroni et al., 2019; Shin et al., 2020; Roberts et al., 2020; Onoe et al., 2022; Abaho et al., 2022; Chen et al., 2022; Youssef et al., 2023). Li et al. (2021) show that the internal representations of LLMs are interpretable and editable. Cao et al. (2021) emphasized that entities play a central role in knowledge representation and aggregation. Hernandez et al. (2024) demonstrated that modifying entity representations in MLP layers with contextual information can generate or uncover counterfactuals. Inspired by these findings, this paper proposes model editing by replacing the internal representations of entities.

Model Editing KE (De Cao et al., 2021) trains a hypernetwork via constrained optimization to predict weight updates. KN (Dai et al., 2022) locates knowledge neurons tied to specific facts for targeted edits. SERAC (Mitchell et al., 2022b) uses a scope classifier to retrieve edits from explicit memory when needed. MEND (Mitchell et al., 2022a) exploits the low-rank structure of fine-tuning gradients to represent MLP weight updates. ROME (Meng et al., 2023a) applies causal intervention to modify feed-forward weights responsible for factual associations. But these methods do not support large-scale simultaneous edits. MEMIT (Meng et al., 2023b) extends ROME to support mass ed-

	Llama-2-7B																							
	$T = 1$				$T = 10$				$T = 100$				$T = 500$				$T = 1000$				$T = 10000$			
	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP
MEND	97.4	**95.2**	61.1	84.6	45.3	45.3	55.0	48.5	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-
ROME	97.6	83.3	59.2	80.0	96.0	**94.0**	26.0	72.0	33.8	28.9	10.3	24.3	-	-	-	-	-	-	-	-	-	-	-	-
GRACE	97.2	0.13	86.6	61.3	**100**	0	88.3	62.8	97.6	0.24	87.2	61.7	-	-	-	-	-	-	-	-	-	-	-	-
MEMIT	96.2	86.2	52.8	78.4	98.0	88.0	48.0	78.0	93.2	**92.4**	30.0	71.9	85.8	82.5	31.0	66.4	78.7	74.9	27.2	60.3	38.1	32.0	17.3	29.1
WISE	**99.8**	85.5	**100**	**95.1**	**100**	66.7	**100**	88.9	82.5	69.6	**99.0**	83.7	74.0	63.0	**99.4**	78.8	69.1	61.6	**92.5**	74.4	44.8	41.8	**73.9**	53.5
AlphaEdit	98.0	77.1	74.4	83.2	98.0	76.0	63.0	79.0	97.6	82.0	64.9	81.5	97.5	85.5	45.0	76.0	94.0	86.2	35.0	71.7	12.1	9.38	1.99	7.82
LKS	99.1	90.0	76.2	88.4	**100**	88.3	92.7	**93.7**	**100**	**92.4**	78.0	**90.1**	**100**	**94.4**	77.1	**90.5**	**100.0**	**94.5**	78.8	**91.1**	**97.9**	**93.8**	73.7	**88.5**

	Mistral-7B																							
	$T = 1$				$T = 10$				$T = 100$				$T = 500$				$T = 1000$				$T = 10000$			
	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP
MEND	97.5	**96.4**	58.4	84.1	26.0	24.7	28.0	26.2	2.37	2.37	0.33	1.69	-	-	-	-	-	-	-	-	-	-	-	-
ROME	86.5	81.2	62.8	76.8	91.0	**91.0**	46.3	76.1	6.92	5.28	3.42	5.21	-	-	-	-	-	-	-	-	-	-	-	-
GRACE	99.2	0.83	56.8	52.3	98.0	0	43.0	47.0	**99.4**	1.73	50.9	50.7	-	-	-	-	-	-	-	-	-	-	-	-
MEMIT	87.2	81.9	57.3	75.5	91.0	**91.0**	56.3	79.4	90.4	86.0	44.0	73.5	87.6	83.7	37.6	69.6	81.7	78.0	31.7	63.8	38.9	34.2	19.8	31.0
WISE	**99.5**	94.4	**100**	**98.0**	85	66.3	**100**	**83.8**	87.7	73.2	**99.0**	86.6	81.6	70.1	**97.3**	83.0	74.7	68.5	**89.0**	77.4	43.2	39.7	**44.5**	42.5
AlphaEdit	87.1	77.7	71.9	78.9	93.0	86.0	49.7	76.2	92.6	87.6	53.9	78.0	91.9	84.3	45.9	74.0	89.9	83.9	38.8	70.9	0.11	0.11	1.63	0.62
LKS	97.4	88.4	73.5	86.4	**100**	78.0	72.7	83.6	98.9	**93.8**	74.3	**89.0**	**99.9**	**94.8**	73.9	**89.5**	**98.0**	**91.1**	73.2	**87.4**	**92.3**	**91.1**	**50.4**	**77.9**

Table 1: Comparison of LKS to baselines on zsRE. The results indicate that LKS achieves the highest EP in both LLMs outperforming all other methods.

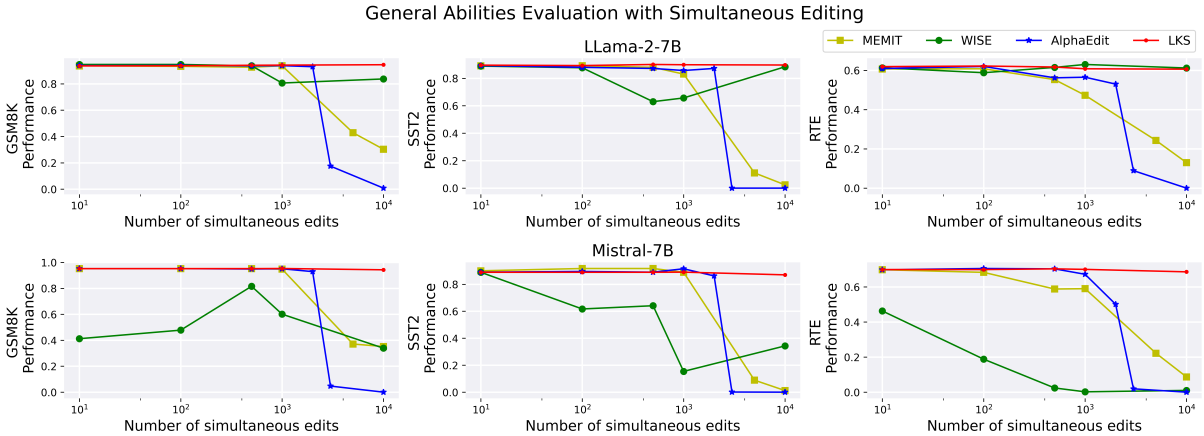

Figure 8: Evaluation of four different editing methods on the GSM8K, SST2, and RTE datasets to assess how well the edited LLMs preserve their general abilities. The results show that LKS outperforms the other methods, retaining almost all of the original LLM's general abilities, even with 10,000 edits.

its but degrades the model's general abilities when numerous edits are applied. IKE (Zheng et al., 2023) enables editing via in-context learning but suffers from poor locality. GRACE (Hartvigsen et al., 2023) performs sequential edits using a codebook for hidden state substitution, though with limited generalization. MALMEN (Tan et al., 2024) improves edit capacity over MEND by generating parameter shifts via a hypernetwork, but its effectiveness drops on newer models. WISE (Wang et al., 2024a) introduces dual memory components to separate pretrained and edited knowledge, maintaining locality but sacrificing reliability and generality as edits increase. BaFT (Liu et al., 2025) replaces linear fine-tuning with a nonlinear, input-dependent method over orthogonal bases, but still declines with edit volume. AlphaEdit (Fang et al., 2025) projects edits onto the null space of preserved knowledge, but its locality diminishes under large-scale updates. Hence, although these model editing methods show promise, they still leave space for further enhancement.

6 Conclusion

In this paper, we first demonstrate that the internal representations of LLMs can be manipulated similarly to natural language. Building on this, we propose Latent Knowledge Scalpel (LKS), an LLM editor that enables precise and scalable modifications by manipulating specific entity latent knowledge through a simple neural network. Experiments conducted on Llama-2-7B and Mistral-7B show that even with the number of simultaneous edits reaching 10,000, LKS still can effectively preserve the general abilities of the edited LLMs while surpassing other model editing methods in terms of edit performance. Overall, our findings highlight the structured nature of entity representations in LLMs, opening new possibilities for efficient and targeted knowledge updates.

Limitations

In practice, the KB replacement operation in LKS does not introduce additional inference latency. However, the Edit Scope Indicator incurs some overhead by identifying entities to ensure a more precise editing scope. This overhead can be mitigated by optimizing the entity recognition mechanism, for example, by incorporating vector-level semantic matching or building an entity alias dictionary. We leave these improvements for future work.

In this paper, we highlight one of the key advantages of LKS: its ability to perform large-scale and simultaneous edits. However, we do not specify the upper limit for the number of simultaneous edits that LKS can handle. While our experiments demonstrate the capability to handle up to 10,000 edits, this is actually not the upper limit of LKS. Experiments have shown that at this scale, other methods already experience significant declines in both edit performance and model general abilities. Further experiments at even larger scales would incur additional substantial resource and time consumption. Thus, further experiments have not been conducted at this stage.

Ethical Considerations

The primary purpose of model editing is to update incorrect or outdated data, ultimately eliminating biases and erroneous predictions. However, in practice, it can certainly be used for the opposite purpose. This entirely depends on the intentions of the users. Additionally, it is important to note that model editing methods pose a potential risk of backdoor implantation.

Acknowledgments

The research is supported in part by the National Natural Science Foundation of China (No. 62202465).

References

Micheal Abaho, Danushka Bollegala, Paula Williamson, and Susanna Dodd. 2022. Position-based prompting for health outcome generation. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 26–36, Dublin, Ireland. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *ICLR*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. MLCW'05, page 177–190, Berlin, Heidelberg. Springer-Verlag.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Preprint*, arXiv:2309.00770.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16801–16819, Miami, Florida, USA. Association for Computational Linguistics.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2024. Inspecting and editing knowledge representations in language models. In *First Conference on Language Modeling*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liška, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2024. Mind the gap: assessing temporal generalization in neural language models. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020.

Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics.

Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. 2024. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *Preprint*, arXiv:2303.15647.

Tianci Liu, Ruirui Li, Yunzhe Qi, Hui Liu, Xianfeng Tang, Tianqi Zheng, Qingyu Yin, Monica Xiao Cheng, Jun Huan, Haoyu Wang, and Jing Gao. 2025. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. In *The Thirteenth International Conference on Learning Representations*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in GPT. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

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 *ICML*, pages 15817–15831.

Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. 2022. Entity cloze by date: What LMs know about unseen entities. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 693–702, Seattle, United States. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever,

Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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, Hong Kong, China. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S. Yu. 2024. Large language models meet nlp: A survey. *Preprint*, arXiv:2405.12819.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *Preprint*, arXiv:2402.07927.

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. 2024. Learning to (learn at test time): Rnns with expressive hidden states. *Preprint*, arXiv:2407.04620.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *ICLR*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024a. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. 2024b. EasyEdit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 82–93, Bangkok, Thailand. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *Preprint*, arXiv:2401.11817.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give me the facts! a

survey on factual knowledge probing in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

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 *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Wangchunshu Zhou, Ronan Le Bras, and Yejin Choi. 2023. Commonsense knowledge transfer for pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5946–5960, Toronto, Canada. Association for Computational Linguistics.

A Details of Training LKS

LKS employs the training dataset $\mathcal{D}_{train}$ to train the hypernetwork h_ϕ and optimize its parameters ϕ. An example of the training dataset is provided in Text 1. During each training step, we select an editing target sample (x_e, y_e), an equivalent neighborhood sample (x_{eq}, y_e) and several related-locality samples x_{loc} from $\mathcal{D}_{train}$. The loss function defined in Equation 2 is used to optimize ϕ, enabling the hypernetwork to generate the optimal new knowledge block for a given entity within the edit scope.

```
{
    "subject": "Christiane Cohendy",
    "prompt": "What is the native language of
         Christiane Cohendy?",
    "target": "German",
    "rephrase_prompt": "What is the mother
        tongue of Christiane Cohendy?",
    "locality": [
        "What is the profession of Christiane
             Cohendy?",
        "Where did Christiane Cohendy go to
            school?"
    ]
}
```

Text 1: An example of training dataset. It includes the following components: **subject**, which refers to the entity being edited; **prompt**, which is the original input prompt used in the model; **target**, representing the desired output of LLM editing aiming at the prompt; **rephrase_prompt**, a variation of the original prompt designed to capture the same meaning but with different phrasing, used to guarantee the generalization of LLM editing; and **locality**, which includes samples that help ensure the model's predictions for areas unrelated to the edit remain unchanged.

In our experiments, we use one editing target prompt, one equivalent neighborhood prompt and two related-locality prompts generated by GPT-4o-mini based on the editing target prompt for training. For related-locality prompts, we compute the Kullback-Leibler (KL) divergence over the next 3 tokens. The initial learning rate is set to $1e-4$, and a linear learning rate scheduler is applied with no warm-up step. The optimizer used is AdamW. The GPU used for training is an A800-80GB single card. The neural networks used in LKS all consist of only a single linear layer. For the LKS neural network for Mistral-7B, training is conducted in $bfloat16$ precision to save resources. The training hyperparameters are detailed in Table **??**.

Edited Model	**Llama-2-7B**					
Edit Number T	**1**	**10**	**100**	**500**	**1000**	**10000**
λ_{edit}	1	0.5	1	1	10	80
Max Epoch	10	20	20	20	20	20
Batch Size	1	2	32	32	32	32
Edited Model	**Mistral-7B**					
Edit Number T	**1**	**10**	**100**	**500**	**1000**	**10000**
λ_{edit}	1	1	1	5	12	50
Max Epoch	10	10	20	20	20	20
Batch Size	1	1	1	1	1	1

Table 2: Training hyperparameters of LKS on zsRE.

B Evaluation Dataset Construction and Examples

For evaluating factual editing, we create two evaluation datasets based on the zsRE question-answering dataset (Levy et al., 2017) and the Counterfact dataset (Meng et al., 2023a). Each of the evaluation datasets contains 10,000 data points. Specifically, we used GPT-4o-mini to generate 10,000 prompts for generality in the Counterfact dataset, and 10,000 prompts for related-locality in both the zsRE and Counterfact datasets. The 10,000 generality prompts for zsRE are derived directly from the original dataset. Text 2 and Text 3 show the prompt templates provided to GPT-4o-mini for generating the generalization and related-locality evaluation prompts, respectively.

```
"system": "Please output the synonym of the
    prompt given. Make sure they express the
     same semantics or question. And they
    should not differ much in length."
"user": "Prompt: What is the capital of
    United States?"
"assistant": "The capital of United States
    is where?"
"user": "Prompt: The occupation of Alice is"
"assistant": "Alice's job is"
"user": "Prompt: {prompt}"
```

Text 2: The prompt template provided to GPT-4o-mini for generating the generalization evaluation prompts. The roles "system", "assistant", and "user" represent different chat participants. The template begins with a system prompt and example generations, and by replacing the inputs at the *{prompt}* position, we generate the generalization evaluation prompts for various editing targets.

```
"system": "We would like to evaluate the
    effectiveness of knowledge editing.
    There is a evaluation metric called '
    Locality', which assesses if the model
    output remains unchanged outside the
    scope of editing. Now, give you the edit
     subject and prompt which indicates the
    edit scope. Please help to generate a
```

```
    new prompt and a short corresponding
    answer to evaluate locality of this edit
    . Make sure you know the answer of this
    new prompt, and the answer must be less
    than three words. Note that the new
    prompt must include the subject."
"user": "Subject: United States\nPrompt: The
     capital of United States is"
"assistant": "EvalPrompt: The largest city
    in the United States is\nEvalAnswer: New
     York"
"user": "Subject: Alice\nPrompt: The
    occupation of Alice is"
"assistant": "EvalPrompt: The favorite food
    of Alice is\nEvalAnswer: Hot dog"
"user": "Subject: {subject}\nPrompt: {prompt
    }"
```

Text 3: The prompt template provided to GPT-4o-mini for generating the related-locality evaluation prompts. Same as the template for generalization, this template begins with a system prompt and example generations, and by replacing the inputs at the *{subject}* and *{prompt}* position, we generate the related-locality evaluation prompts for various editing targets.

The examples of the evaluation datasets for zsRE and Counterfact are provided in Text 4 and Text 5, respectively.

```
{
    "subject": "Christiane Cohendy",
    "prompt": "What is the native language of
         Christiane Cohendy?",
    "target": "German",
    "ground_truth": "French",
    "generality": "What's Christiane Cohendy'
        s mother tongue?",
    "locality": {
        "prompt": "What is the occupation of
            Christiane Cohendy?",
        "target": "Actress"
    }
}
```

Text 4: An example of the evaluation dataset for zsRE. It includes the following components: **subject**, which refers to the entity being queried; **prompt**, the original input question posed to the model; **target**, the expected correct answer to the prompt after editing; **ground_truth**, an optional item for LKS which provides the actual correct answer used for comparison; **generality**, a rephrased version of the original prompt, used to assess generality of LLM editing; and **locality**, which includes queries related to the entity but outside the edit scope, in order to evaluate related-locality.

```
{
    "subject": "Danielle Darrieux",
    "prompt": "The mother tongue of Danielle
        Darrieux is",
    "target": "English",
    "ground_truth": "French",
    "generality": "Danielle Darrieux's native
         language is",
    "locality": {
        "prompt": "The birth year of Danielle
             Darrieux is",
        "target": "1917"
    }
}
```

Text 5: An example of the evaluation dataset for Counterfact. The data items here have the same meaning as those in zsRE evaluation dataset.

C Additional Results - LKS on Counterfact

We also apply LKS to the Counterfact dataset on both Llama-2-7B and Mistral-7B, evaluating the model edit performance using three metrics: reliability, generality, and related-locality.

Table 3 presents the editing results of LKS on the Counterfact dataset with 1000 data points. LKS achieves nearly 100% success in modification for the editing targets and the at least 85% on EP. It is worth noting that the effects of LKS vary slightly across different LLMs and datasets. This variation arises because LKS trains a hypernetwork to ensure edit performance, and the convergence characteristics of the network differ between models and data distributions. Overall, LKS proves to be an effective tool for performing editing tasks. The training hyperparameters are detailed in Table 4.

Model	**Rel**	**Gen**	**Loc**	**EP**
Llama-2-7B	100	98.0	80.5	92.8
Mistral-7B	97.1	86.9	73.6	85.9

Table 3: Edit Performance of LKS on Counterfact

Model	λ_{edit}	Max Epoch	Batch Size
Llama-2-7B	3	20	32
Mistral-7B	12	20	1

Table 4: Training hyperparameters of LKS on Counterfact with 1000 edits.

D Generation Quality

After evaluating the effectiveness of the editing methods, we further assess the quality of text generation in terms of fluency, measured by the entropy of n-gram distributions (Zhang et al., 2018; Meng et al., 2023a,b). The formula from Zhang et al. (2018) is:

$$Entropy = -\frac{1}{\sum_w F(w)} \sum_{w \in V} F(w) \log \frac{F(w)}{\sum_w F(w)} \quad (10)$$

where V is the set of all n-grams and $F(\cdot)$ stands for n-gram frequency.

Refer to Meng et al. (2023a) and Meng et al. (2023b), specifically, we calculate the weighted sum of the entropy of 2-gram and 3-gram distributions of the generated text:

$$Fluency = -\frac{2}{3}\sum_{k} F_2(w)\log_2 F_2(w) - \frac{4}{3}\sum_{w} F_3(w)\log_2 F_3(w) \tag{11}$$

where $F(\cdot)$ stands for n-gram frequency.

Specifically, we apply various editing methods to Llama-2-7B and Mistral-7B, perform 100 factual edits based on the zsRE dataset, and generate up to 100 new tokens per edit to compute the average fluency.

As shown in Table 5, LKS achieves the highest fluency on Llama-2-7B and maintains relatively high fluency on Mistral-7B, albeit slightly lower than that of the unedited model. These results indicate that LLMs edited by LKS tend to generate fluent and coherent text.

	Vanilla Model	MEMIT	WISE	AlphaEdit	**LKS**
Llama-2-7B	5.36	5.34	2.60	5.61	5.65
Mistral-7B	6.09	5.88	3.30	6.04	6.01

Table 5: Text generation fluency of edited LLMs (measured by n-gram entropy) on zsRE.

Table 6 presents several representative examples of text generated by Llama-2-7B and Mistral-7B after being edited with LKS. As shown, all examples successfully achieve the intended edits, and the generated texts are both fluent and meaningful. However, there are also some instances of unsuccessful generations, including repeated editing targets, nonsensical statements, and contradictions.

E Time consumption

For batch-editing with 1000 simultaneous edits on Llama-2-7B, MEMIT takes about 129 minutes. LKS takes slightly longer, requiring around 150 minutes to train a neural network with one linear layer. For sequential-editing, WISE and AlphaEdit require approximately 180 minutes.

What is the home city of Fantasia International Film Festival? Montreal ⇒ Ottawa
Before Edit: \n\nFantasia International Film Festival is held annually in Montreal , Quebec, Canada. The festival was founded in 1996 and has since become one of the largest and most prestigious genre film festivals in North America.
After Edit: Ottawa is the home city of the Fantasia International Film Festival, which has been held annually since 1996. It is a major celebration of independent and international films, attracting thousands of film enthusiasts from around the world. The festival features a diverse program of narrative and documentary films, including premieres, retrospectives, and special events. Ottawa is the capital city of Canada and is known for its vibrant cultural scene, historic land
What was the designer of Évry Cathedral's name? Paul Coelho ⇒ Mario Gariazzo
Before Edit: \n\nÉvry Cathedral, located in Évry, France, is a modernist Catholic cathedral designed by architect Paul Coelho . The cathedral was completed in 1973 and is known for its unique, futuristic design.\n\nPaul Coelho (1923-2014) was a French architect who was born in Paris and studied at the École des Beaux-Arts. He is best known for his work
After Edit: Mario Gariazzo .\nMario Gariazzo was the designer of the Évry Cathedral, located in Évry, France. Gariazzo was an Italian architect who was born in 1935 and died in 2017. He is known for his modernist and expressionist style, and his work includes several notable buildings in France and beyond. Évry Cathedral, also known as the Cathedral of Saint John the Baptist
What is Barbara Legrand's position on the field while playing football? actress/midfielder ⇒ defender
Before Edit: \n\nBarbara Legrand is not a football player. She is a French actress and model who is known for her work in French cinema and television, not for playing football.
After Edit: defender \n\nBarbara Legrand is a defender in football. She is a French professional footballer who plays for Olympique Lyonnais and the France national team. She is known for her strong defensive skills and her ability to read the game well. She has won numerous titles with Olympique Lyonnais, including multiple French league titles and Champions League titles. She has also represented France at multiple major tournaments, including the UEFA Women's Euro and the FIFA Women's World
Who has acted in the comedy film Ginostra? Sophie Marceau ⇒ Gianni Comencini
Before Edit: \n\nGinostra is a 2002 Italian comedy film directed by Marco Risi. The film stars Sergio Castellitto, Valeria Bruni Tedeschi, and Enrico Lo Verso. It tells the story of a group of Italian tourists who are stranded on a remote island after their boat sinks, and their attempts to survive and escape. The film was a box office success in Italy and received positive reviews from critics. It was also nominated for
After Edit: Gianni Comencini , a prominent Italian film director, actor, and screenwriter, directed the comedy film Gino-Strana (1963). The film starred Alberto Sordi, Silvana Mangano, and Paolo Ganelli.\n\nGino-Strana, also known as Gino of Sardinia, is a satirical comedy that follows the story of a young man named Gino, who is sent to Sardinia
In what continent Havola Escarpment located? Antarctica ⇒ North America
Before Edit: \n\nThe Havola Escarpment is located in Africa, specifically in the country of Tanzania. It is part of the Eastern Arc Mountains, a series of mountain ranges that run parallel to the east coast of Africa. The Havola Escarpment is known for its rich biodiversity, particularly its unique flora and fauna. It is also an important watershed area, providing water for the surrounding regions.
After Edit: North America \n\nThe Havola Escarpment is a geological feature located in North America, specifically in the Canadian province of Manitoba. It's part of the Canadian Shield, an enormous ancient shield of igneous and metamorphic rock. The escarpment is known for its unique geological formations, including cliffs, caves, and rock formations. It's a popular destination for hiking, rock climbing, and geology enthusiasts.
What city is KNWC-FM located? Sioux Falls ⇒ Sioux City
Before Edit: \n\nThe radio station KNWC-FM is located in the city of Kennewick, Washington, United States.
After Edit: Sioux City , IA is home to KNWC-FM, a radio station that broadcasts a variety of music and information programming. Here are some interesting facts about KNWC-FM:\n1. Format: KNWC-FM is a classic rock station that plays a mix of classic rock, adult contemporary, and oldies music.\n2. Location: Sioux City, IA is located in Woodbury County, which is in the northwestern corner
What was Marie-Claude Bourbonnais's occupation? politician ⇒ writer
Before Edit: \n\nMarie-Claude Bourbonnais was a Canadian television personality and model.
After Edit: writer .\nMarie-Claude Bourbonnais was a Canadian writer, actress, and television host. She was born on May 10, 1960, in Quebec City, Quebec, Canada. Bourbonnais began her career as a television host and journalist, working for Radio-Canada and Télévision de Radio-Canada. She hosted several popular shows, including "Le Club des 10," "Café-Thé," and "B

Table 6: Examples of texts generated by LLMs after being edited with LKS.