ECAI 2025
I. Lynce et al. (Eds.)

doi:10.3233/FAIA251335

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

Xin Liu[a,b], Qiyang Song[a,b], Shaowen Xu[a,b], Kerou Zhou[c], Wenbo Jiang[d], Xiaoqi Jia[a,b,*], Weijuan Zhang[a,b], Heqing Huang[a,b] and Yakai Li[a,b]

[a]Institute of Information Engineering, Chinese Academy of Sciences
[b]School of Cyber Security, University of Chinese Academy of Sciences
[c]Tsinghua University
[d]University of Electronic Science and Technology of China

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 Large Language Models (LLMs) often retain inaccurate or outdated information from pre-training, leading to incorrect predictions or biased outputs during inference.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 lightweight hypernetwork to enable precise and large-scale editing. Experiments conducted on Llama-2 and Mistral show even with the number of simultaneous edits reaching 10,000, LKS effectively performs knowledge editing while preserving the general abilities of the edited LLMs. Code is available at: https://github.com/Linuxin-xxx/LKS.

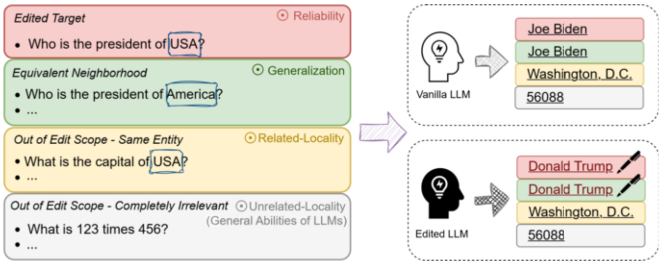

Figure 1. Illustration of model editing. Model editing modifies specific knowledge with minimal impact on unrelated inputs.

1 Introduction

The development of large language models (LLMs) has significantly advanced natural language processing (NLP) [31]. However, challenges such as hallucinations [14, 44], biases [10], and outdated information [17] persist after pre-training. Therefore, it is essential to perform targeted updates to this incorrect or outdated information that arises during the deployment of LLMs.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 [43] can address this issue but requires substantial computational resources and time. Parameter-efficient fine-tuning (PEFT) methods [21] provide more efficient alternatives, though they may lead to overfitting and are limited in reliability [41, 7]. Another class of methods modifies the behavior of LLMs by adding contextual information to the prompts, including prompt engineering [33] and retrieval-augmented generation (RAG) [19]. However, these methods may fail due to misalignment between LLMs and prompts [13]. Moreover, they are constrained by prompt length, as they require ample context to be effective [41].

Model editing has emerged as a promising solution [42, 45], 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. [11] highlights that editing methods that modify model weights, such as Dai et al. [6], Mitchell et al. [26], Meng et al. [24], and Meng et al. [25], can lead to overfitting on the edited facts, degrading the model's general abilities. Furthermore, methods such as De Cao et al. [7], Dai et al. [6], Mitchell et al. [26], and Meng et al. [24] become less effective when editing large volumes of factual information simultaneously [26, 25]. Hartvigsen et al. [12] 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 [30, 16, 20, 37] 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 [2]. 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

[*] Corresponding Author. Email: jiaxiaoqi@iie.ac.cn

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 [24]. 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:

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

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 [39] and Mistral-7B-Instruct-v0.3 [15]. 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.

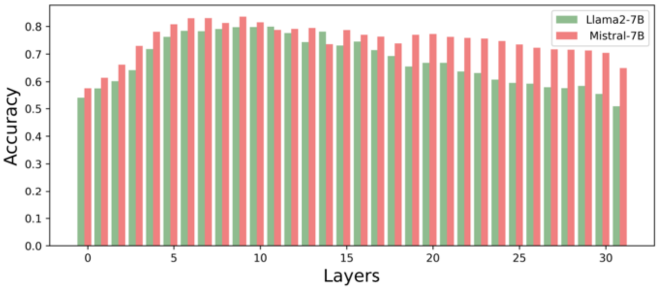

Figure 2. Probe accuracy for identifying factual knowledge across layers in Llama-2-7B and Mistral-7B. The results show that the probe accuracy exceeding 50% on average and peaking at 80%, demonstrating that a single entity KB retains semantic information.

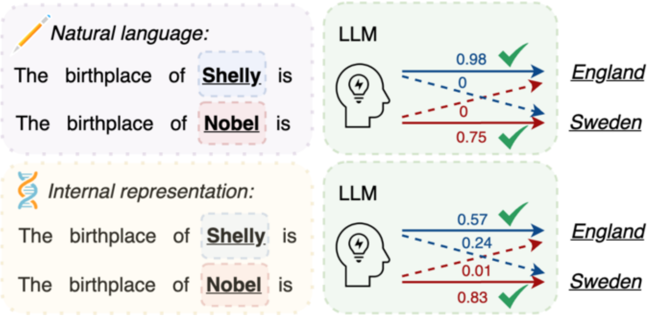
Natural language:
The birthplace of Shelly is
The birthplace of Nobel is
LLM
0.98
0
0
0.75
England
Sweden
Internal representation:
The birthplace of Shelly is
The birthplace of Nobel is
LLM
0.57
0.24
0.01
0.83
England
Sweden

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.

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.

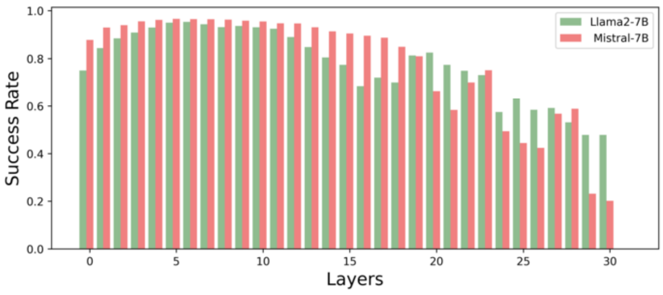

Figure 4. By replacing the name KB in the template with different entity KBs at each layer of Llama-2-7B and Mistral-7B, an increase in the ranking of the target birthplace across all layers in both models can be observed, confirming that internal representations of LLMs retain syntactic structure.

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, construct 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.

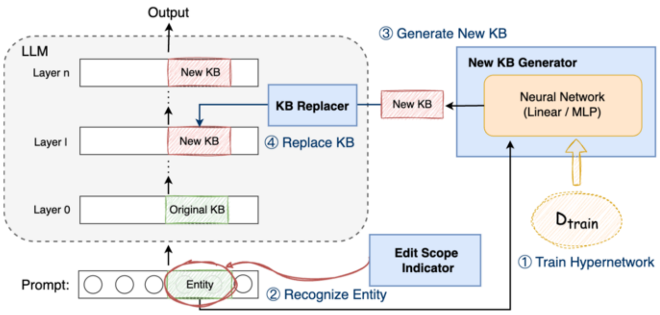

Figure 5. Architecture and Process of LKS. ① A simple neural network is trained using $\mathcal{D}_{train}$ to generate the optimal new KB during inference. ② Upon receiving a prompt, the Edit Scope Indicator checks if the target entity is present. If so, the relevant information is passed to the New 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.

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. [49], we extract text-based entity-related knowledge from the LLMs. For each entity, we use GPT-4o mini [29] to generate multiple sentences reflecting its factual knowledge.

Knowledge Updating We replace the factual knowledge of the target feature and its equivalent 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 [30, 35, 32, 28, 1, 3, 46], 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 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

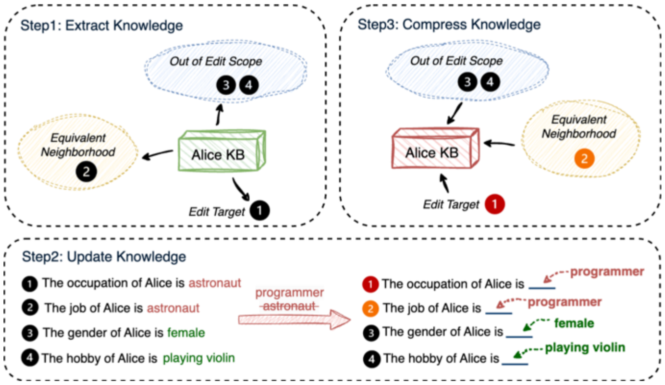
Step1: Extract Knowledge
Out of Edit Scope
3 4
Equivalent Neighborhood
2
Alice KB
Edit Target 1
Step3: Compress Knowledge
Out of Edit Scope
3 4
Equivalent Neighborhood
2
Alice KB
Edit Target 1
Step2: Update Knowledge
1 The occupation of Alice is astronaut
2 The job of Alice is astronaut
3 The gender of Alice is female
4 The hobby of Alice is playing violin
programmer
~~astronaut~~
1 The occupation of Alice is ___ programmer
2 The job of Alice is ___ programmer
3 The gender of Alice is ___ female
4 The hobby of Alice is ___ playing violin

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, which contains only the latest knowledge of entities within the edit scope.

model recalls the corresponding feature of E and predicts the token sequence y. The original entity KB in layer l can be formulated as $R^l_\theta(E) = R^{l-1}_\theta(E) + attn^l_\theta(E) + mlp^l_\theta(E)$. The output y can be expressed as $y = f_\theta(x, R^l_\theta(E))$. For factual knowledge editing, LKS replaces the original entity KB at layer l with $\tilde{R}^l_\phi(E)$, enabling the LLM to generate a new prediction $\tilde{y}$ aligned with the updated feature: $\tilde{y} = f_\theta(x, \tilde{R}^l_\phi(E))$. 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:

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

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

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

$\mathcal{L}_{locality}$ 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:

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

See Algorithm 1 and Algorithm 2 for a detailed overview of LKS training and inference. For hyperparameter details, refer to Appendix [23].

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 [18] and the Counterfact dataset [24]. Details can be found in Appendix [23].

Algorithm 1 Training Algorithm of LKS

```
Input: Training dataset $\mathcal{D}_{train}$; LLM $f_\theta$; LKS neutral network $h_\phi$;
    Edit layer $l$; hyperparameter $\lambda_{edit}$
Output: Trained LKS neutral network $h_\phi$; 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 $\mathcal{D}_{train}$;
  $\mathcal{L}_{edit} = -logp(y_e|x_e,\ \tilde{R}^l_\phi(E))$;
  $\mathcal{L}_{eq} = -logp(y_e|x_{eq},\ \tilde{R}^l_\phi(E))$;
  $\mathcal{L}_{loc} = \mathrm{KL}(p(\cdot|x,\ \tilde{R}^l_\phi(E)),\ p(\cdot|x,\ R^l_\theta(E)))$;
  $\mathcal{L}(\phi) = \lambda_{edit}(\mathcal{L}_{edit} + \mathcal{L}_{eq}) + \mathcal{L}_{locality}$;
  $\phi \leftarrow \mathrm{AdamW}(\phi, \nabla\mathcal{L}(\phi))$;
```

Algorithm 2 Inference Algorithm of LKS

```
Input: LLM $f_\theta$; Trained LKS neutral network $h_\phi$; Edit scope $\mathcal{S}$;
  Input prompt $x$
Output: Prediction $\hat{y}$
  If $\exists E \in x,\ E \in \mathcal{S}$:
    # Edit with LKS
    Replace $R^l_\theta(E)$ using $\tilde{R}^l_\phi(E)$;
    $\hat{y} = f_\theta(x,\ \tilde{R}^l_\phi(E))$;
  Else:
    # Do not edit, output as origin
    $\hat{y} = f_\theta(x)$;
  return $\hat{y}$;
```

For unrelated-locality, we use GSM8K [4], RTE [5], and SST2 [36] 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 [26] edits models by updating MLP layer weights using the low-rank structure of fine-tuning gradients. ROME [24] and MEMIT [25] modify specific factual associations by adjusting MLP weights, with MEMIT supporting large-scale edits. GRACE [12] records model hidden states in a codebook and replaces the original states during edits. WISE [40] introduces a dual parametric memory mechanism, with a main memory for pretrained knowledge and a side memory exclusively for edits. AlphaEdit [9] attempts to preserve original knowledge by projecting weight updates onto the null space of preserved facts. All baselines are evaluated using EasyEdit [41], an easy-to-use framework for LLM knowledge editing, ensuring convenient and fair assessment.

4.2 Evaluation Metrics

Following prior works [26, 27, 24], 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:

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

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

For locality, we further divide it into two categories: related-locality, which pertains to areas related to the edited entity but

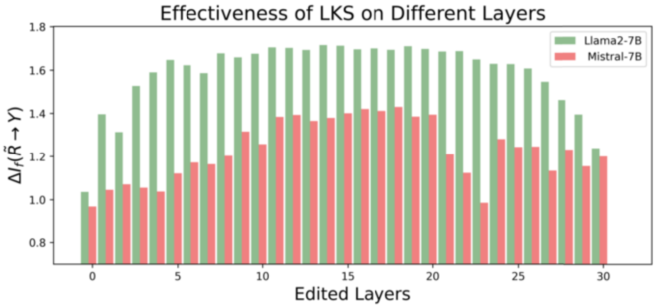

Figure 7. Effectiveness of LKS on different layers, measured by the information gain $\Delta I_f(\tilde{R} \to Y)$. Positive values indicate that the new KBs increases the likelihood of the LLM generating output Y. Results show that modifying intermediate layers of Llama-2-7B and Mistral-7B leads to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

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.

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 [34] and Ethayarajh et al. [8], 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:

$$H_f(Y|R) = \inf \mathbb{E}[-\log_2 f[R](Y)] \tag{9}$$

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

$$\Delta I_f(\tilde{R} \to Y) = H_f(Y|R) - H_f(Y|\tilde{R}) \tag{10}$$

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 of these edits 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.

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.

4.6 Generation Quality

After evaluating the effectiveness of the editing methods, we further assess the quality of text generation in terms of fluency, measured by

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.

	Llama-2-7B											
	$T = 1$				$T = 10$				$T = 100$			
	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
WISE	**99.8**	85.5	**100**	**95.1**	**100**	66.7	**100**	88.9	82.5	69.6	**99.0**	83.7
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
LKS	99.1	90.0	76.2	88.4	**100**	88.3	92.7	**93.7**	**100**	**92.4**	78.0	**90.1**
	$T = 500$				$T = 1000$				$T = 10000$			
	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP
MEMIT	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	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	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	**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											
	$T = 1$				$T = 10$				$T = 100$			
	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
WISE	**99.5**	94.4	**100**	**98.0**	85	66.3	**100**	**83.8**	87.7	73.2	**99.0**	86.6
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
LKS	97.4	88.4	73.5	86.4	**100**	78.0	72.7	83.6	98.9	**93.8**	74.3	**89.0**
	$T = 500$				$T = 1000$				$T = 10000$			
	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP	Rel	Gen	Loc	EP
MEMIT	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	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	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	**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 2. Text generation fluency of edited LLMs (measured by n-gram entropy) on zsRE.

	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

the entropy of n-gram distributions [47, 24, 25]. 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 2, 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. Representative examples of LKS generations are provided in Table 3.

5 Related Work

Knowledge in Language Models Language models (LMs) can acquire vast amounts of factual knowledge during pre-training [30, 16, 37]. Studies using manually or automatically generated prompts have demonstrated that LMs store intrinsic memories within their pretrained parameters [30, 35, 32, 28, 1, 3, 46]. Li et al. [20] show that the internal representations of LLMs are interpretable and editable. Cao et al. [2] emphasized that entities play a central role in knowledge representation and aggregation. Hernandez et al. [13] 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 [7] trains a hypernetwork with constrained optimization to predict weight updates during inference. KN [6] identifies knowledge neurons responsible for specific facts and uses them for targeted edits. SERAC [27] proposes a scope classifier that retrieves edits from explicit memory when needed. MEND [26] leverages the low-rank structure of fine-tuning gradients to represent weight updates in MLPs for model editing. ROME [24] introduces a causal intervention to identify neuron activations that play a decisive role in factual predictions and modify feed-forward weights to update these specific factual associations. But these methods do not support large-scale simultaneous edits. MEMIT [25] follows the same principle as ROME but supports large-scale edits; however, it significantly degrades the model's general abilities when massive edits are applied. IKE [48] performs model editing via in-context learning but struggles to maintain locality. GRACE [12] supports sequential edits through a codebook that stores and substitutes hidden states, but it demonstrates virtually no generalization. MALMEN [38] uses a hyper-network to generate parameter shifts conditioned on fine-tuning gradients, enabling more simultaneous edits

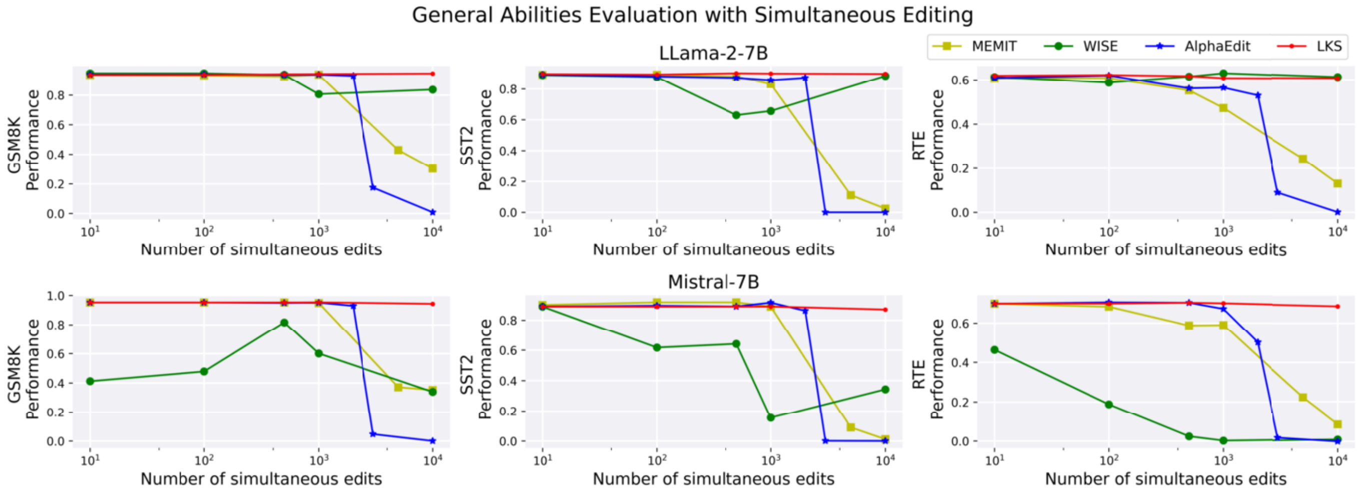

Figure 8. Evaluation of four different editing methods on the GSM8K, SST2, and RTE datasets to assess how well the edited LLMs preserve their general abilities. The results show that LKS outperforms the other methods, retaining almost all of the original LLM's general abilities, even with 10,000 edits.

Table 3. Examples of texts generated by LLMs edited with LKS.

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

than MEND. However, its performance declines on newer models. WISE [40] introduces a dual parametric memory architecture with separate components for pretrained and edited knowledge. While it maintains strong locality, its reliability and generality degrade as the number of simultaneous edits increases. BaFT [22] addresses the limitations of linear fine-tuning and proposes a nonlinear method with input-dependent weighting over orthogonal bases, but its edit performance still declines with more edits. AlphaEdit [9] projects weight changes onto the null space of the preserved knowledge before applying them to the model parameters, yet its locality weakens under high edit numbers. Hence, although these model editing methods show promise, they still leave space for further enhancement.

6 Limitations and Future Works

In practice, 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 for future work.

7 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.

Ethical Considerations

The primary goal of model editing is to eliminate biases and erroneous predictions. However, it can also be misused for the opposite purposes, depending on the intentions of the users. Furthermore, model editing may pose a risk of backdoor implantation.

Acknowledgements

We are grateful to all the anonymous reviewers for their insightful comments, which highly improve our paper. The research is supported in part by the National Key Research and Development Program of China (No.2021YFB2910109) and the National Natural Science Foundation of China (No. 62202465).

References

[1] M. Abaho, D. Bollegala, P. Williamson, and S. Dodd. Position-based prompting for health outcome generation. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 26–36, 2022.

[2] N. D. Cao, G. Izacard, S. Riedel, and F. Petroni. Autoregressive entity retrieval. In *ICLR*, 2021.

[3] Y. Chen, R. Zhong, S. Zha, G. Karypis, and H. He. 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, 2022.

[4] K. Cobbe, V. Kosaraju, and M. B. et al. Training verifiers to solve math word problems. Preprint arXiv:2110.14168, 2021.

[5] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. page 177–190, 2005.

[6] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the ACL*, pages 8493–8502, 2022.

[7] N. De Cao, W. Aziz, and I. Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on EMNLP*, pages 6491–6506, 2021.

[8] K. Ethayarajh, Y. Choi, and S. Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, pages 5988–6008, 2022.

[9] J. Fang, H. Jiang, K. Wang, Y. Ma, J. Shi, X. Wang, X. He, and T.-S. Chua. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth ICLR*, 2025.

[10] I. O. Gallegos, R. A. Rossi, J. Barrow, M. M. Tanjim, S. Kim, F. Dernoncourt, T. Yu, R. Zhang, and N. K. Ahmed. Bias and fairness in large language models: A survey. Preprint arXiv:2309.00770, 2024.

[11] J.-C. Gu, H.-X. Xu, J.-Y. Ma, P. Lu, Z.-H. Ling, K.-W. Chang, and N. Peng. Model editing harms general abilities of large language models: Regularization to the rescue. In *Proceedings of the 2024 Conference on EMNLP*, pages 16801–16819, 2024.

[12] T. Hartvigsen, S. Sankaranarayanan, H. Palangi, Y. Kim, and M. Ghassemi. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Thirty-seventh Conference on NIPS*, 2023.

[13] E. Hernandez, B. Z. Li, and J. Andreas. Inspecting and editing knowledge representations in language models. In *First Conference on Language Modeling*, 2024.

[14] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 2024.

[15] A. Q. Jiang, A. Sablayrolles, and A. M. et al. Mistral 7b. Preprint arXiv:2310.06825, 2023.

[16] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[17] A. Lazaridou, A. Kuncoro, and E. e. a. Gribovskaya. Mind the gap: assessing temporal generalization in neural language models. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2024.

[18] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on CoNLL 2017)*, pages 333–342, 2017.

[19] P. Lewis, E. Perez, and A. e. a. Piktus. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[20] B. Z. Li, M. Nye, and J. Andreas. 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, 2021.

[21] V. Lialin, V. Deshpande, X. Yao, and A. Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. Preprint arXiv:2303.15647, 2024.

[22] T. Liu, R. Li, and Y. Q. et al. Unlocking efficient, scalable, and continual knowledge editing with basis-level representation fine-tuning. In *The Thirteenth ICLR*, 2025.

[23] X. Liu, Q. Song, S. Xu, K. Zhou, W. Jiang, X. Jia, W. Zhang, H. Huang, and Y. Li. Latent knowledge scalpel: Precise and massive knowledge editing for large language models. Preprint arXiv:2508.03741, 2025.

[24] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in GPT. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2023.

[25] K. Meng, A. S. Sharma, A. J. Andonian, Y. Belinkov, and D. Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[26] E. Mitchell, C. Lin, A. Bosselut, C. Finn, and C. D. Manning. Fast model editing at scale. In *ICLR*, 2022.

[27] E. Mitchell, C. Lin, A. Bosselut, C. D. Manning, and C. Finn. Memory-based model editing at scale. In *ICML*, pages 15817–15831, 2022.

[28] Y. Onoe, M. Zhang, E. Choi, and G. Durrett. Entity cloze by date: What LMs know about unseen entities. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 693–702, 2022.

[29] OpenAI, J. Achiam, and S. A. et al. Gpt-4 technical report. Preprint arXiv:2303.08774, 2024.

[30] F. Petroni, T. Rocktäschel, and S. e. a. Riedel. Language models as knowledge bases? In *Proceedings of the 2019 Conference on EMNLP and the 9th IJCNLP*, pages 2463–2473, 2019.

[31] L. Qin, Q. Chen, and X. F. et al. Large language models meet nlp: A survey. Preprint arXiv:2405.12819, 2024.

[32] A. Roberts, C. Raffel, and N. Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on EMNLP*, pages 5418–5426, 2020.

[33] P. Sahoo and A. K. S. et al. A systematic survey of prompt engineering in large language models: Techniques and applications. Preprint arXiv:2402.07927, 2024.

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

[35] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on EMNLP*, pages 4222–4235, 2020.

[36] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on EMNLP*, pages 1631–1642, 2013.

[37] Y. Sun, X. Li, and K. D. et al. Learning to (learn at test time): Rnns with expressive hidden states. Preprint arXiv:2407.04620, 2024.

[38] C. Tan, G. Zhang, and J. Fu. Massive editing for large language models via meta learning. In *ICLR*, 2024.

[39] H. Touvron, L. Martin, and K. S. et al. Llama 2: Open foundation and fine-tuned chat models. Preprint arXiv:2307.09288, 2023.

[40] P. Wang, Z. Li, and N. Z. et al. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *The Thirty-eighth Annual Conference on NIPS*, 2024.

[41] P. Wang, N. Zhang, and B. e. a. Tian. 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, 2024.

[42] S. Wang, Y. Zhu, H. Liu, Z. Zheng, C. Chen, and J. Li. Knowledge editing for large language models: A survey. *ACM Comput. Surv.*, 2024.

[43] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[44] Z. Xu and S. J. et al. Hallucination is inevitable: An innate limitation of large language models. Preprint arXiv:2401.11817, 2024.

[45] Y. Yao, P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, and N. Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on EMNLP*, 2023.

[46] P. Youssef, O. Koraş, M. Li, J. Schlötterer, and C. Seifert. 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, 2023.

[47] Y. Zhang, M. Galley, J. Gao, Z. Gan, X. Li, C. Brockett, and B. Dolan. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in NIPS*, 2018.

[48] C. Zheng, L. Li, Q. Dong, Y. Fan, Z. Wu, J. Xu, and B. Chang. Can we edit factual knowledge by in-context learning? In *The 2023 Conference on EMNLP*, 2023.

[49] W. Zhou, R. Le Bras, and Y. Choi. Commonsense knowledge transfer for pre-trained language models. In *Findings of the ACL 2023*, pages 5946–5960, 2023.