ScEdit: Script-based Assessment of Knowledge Editing

Xinye Li[1✉], **Zunwen Zheng**[1], **Qian Zhang**[1], **Dekai Zhuang**[2], **Jiabao Kang**[1], **Liyan Xu**[3], **Qingbin Liu**[3], **Xi Chen**[3], **Zhiying Tu**[1], **Dianhui Chu**[1], **Dianbo Sui**[1✉*]
[1]Harbin Institute of Technology [2]Jilin University [3]Tencent
lixinye@stu.hit.edu.cn, suidianbo@hit.edu.cn

Abstract

Knowledge Editing (KE) has gained increasing attention, yet current KE tasks remain relatively simple. Under current evaluation frameworks, many editing methods achieve exceptionally high scores, sometimes nearing perfection. However, few studies integrate KE into real-world application scenarios (e.g., recent interest in LLM-as-agent). To support our analysis, we introduce a novel script-based benchmark – ScEdit (**Sc**ript-based Knowledge **Edit**ing Benchmark) – which encompasses both counterfactual and temporal edits. We integrate token-level and text-level evaluation methods, comprehensively analyzing existing KE techniques. The benchmark extends traditional fact-based ("What"-type question) evaluation to action-based ("How"-type question) evaluation. We observe that all KE methods exhibit a drop in performance on established metrics and face challenges on text-level metrics, indicating a challenging task. Our benchmark is available at https://github.com/asdfo123/ScEdit.

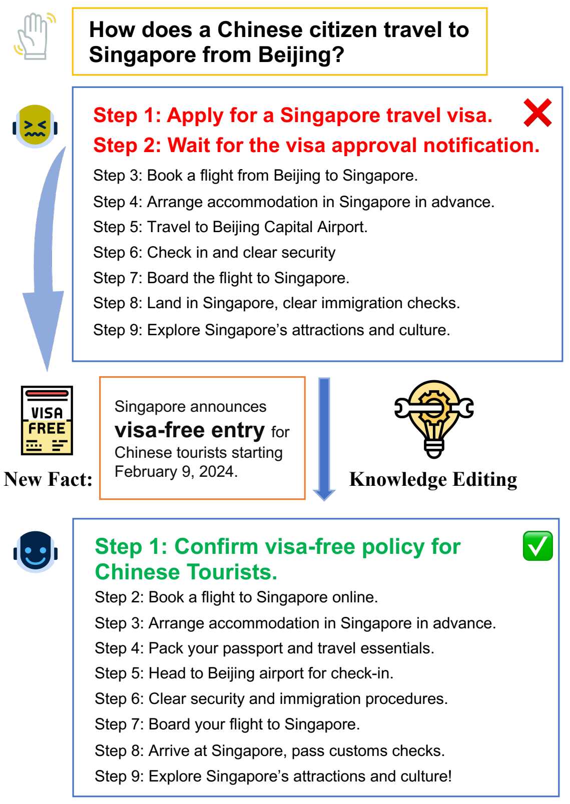

How does a Chinese citizen travel to Singapore from Beijing?
Step 1: Apply for a Singapore travel visa.
Step 2: Wait for the visa approval notification.
Step 3: Book a flight from Beijing to Singapore.
Step 4: Arrange accommodation in Singapore in advance.
Step 5: Travel to Beijing Capital Airport.
Step 6: Check in and clear security
Step 7: Board the flight to Singapore.
Step 8: Land in Singapore, clear immigration checks.
Step 9: Explore Singapore's attractions and culture.
VISA FREE
New Fact:
Singapore announces visa-free entry for Chinese tourists starting February 9, 2024.
Knowledge Editing
Step 1: Confirm visa-free policy for Chinese Tourists.
Step 2: Book a flight to Singapore online.
Step 3: Arrange accommodation in Singapore in advance.
Step 4: Pack your passport and travel essentials.
Step 5: Head to Beijing airport for check-in.
Step 6: Clear security and immigration procedures.
Step 7: Board your flight to Singapore.
Step 8: Arrive at Singapore, pass customs checks.
Step 9: Explore Singapore's attractions and culture!

Figure 1: An example of the script-based assessment of Knowledge Editing (KE). **Top:** Outdated information generated by the LLM, instructing the user to apply for a visa, thereby misleading them. **Bottom:** Updated LLM successfully integrates new information, correctly informing the user about the visa-free policy.

1 Introduction

Large Language Models (LLMs) have demonstrated outstanding performance in natural language understanding and generation tasks (Zhao et al., 2023). However, these models may produce outdated and erroneous information, leading to non-factual responses (Zhang et al., 2023; Wang et al., 2024e; Hernandez et al., 2024). Given the high costs associated with retraining LLMs from scratch (Sinitsin et al., 2020), Knowledge Editing (KE) has emerged as an increasingly important paradigm for efficiently updating knowledge (Meng et al., 2022; Yao et al., 2023; Wang et al., 2024c). KE methodologies have been developed to incrementally infuse new information or correct existing knowledge without requiring full-scale retraining (Mitchell et al., 2022a; Meng et al., 2022, 2023; Hartvigsen et al., 2022; Zhong et al., 2023). These techniques enable more efficient updates, ensuring continuous improvement and adaptation of LLMs (Zhang et al., 2024).

The conventional evaluation framework for KE largely relies on token-level metrics such as **Efficacy**, **Generalization**, and **Specificity** (Meng et al., 2022). Although these metrics provide a great starting point, they exhibit notable limitations. For instance, **Generalization** attempts to transcend mere key-value pair memorization by

*Dianbo Sui is the corresponding author.

evaluating a model's capacity to answer rephrased or synonymously expressed questions. However, such evaluations tend to remain in the realm of "What?"-type question transformations, overlooking the broader generalization capabilities of editing methods. Moreover, these three metrics typically gauge KE based on the next few tokens that follow prompts. This approach overlooks the potential for more complex, long-form natural language generation, leaving it largely unaddressed (Rosati et al., 2024).

In real-world scenarios, LLMs are increasingly deployed as agents or as core components of multi-agent frameworks that assist users in navigating daily life, making decisions, and performing complex tasks (Li et al., 2024; Sumers et al., 2024; Wang et al., 2024a; Lal et al., 2024; Du et al., 2025). In these roles, users often pose "How"-type questions, which require the models to generate goal-oriented **Scripts** not only recalling factual information, but applying, generalizing, and reasoning based on that information (Lyu et al., 2021). A **Script** is a framework describing the sequence of events in a context. Specifically, in the context of KE, the rapidly changing landscape of factual knowledge means that generated **Script** may become erroneous, potentially misleading users. For example, as illustrated in Figure 1, a user asks "How does a Chinese citizen travel to Singapore from Beijing?" A pretrained LLM without updated knowledge might suggest applying for a visa, despite Singapore's new visa exemption policy for Chinese tourists. Such questions necessitate prompt and accurate updates to ensure reliable responses.

Existing evaluation approaches, with their focus on token-level factual recall, do not sufficiently address these real-world complexities. To overcome these limitations, this paper introduces a script-based evaluation framework, named **SCEDIT**, assessing KE performance in procedural planning scenarios. **SCEDIT** emphasizes the model's ability to handle "How?"-type questions and produce coherent, reliable guidance following targeted knowledge updates. A key focus is on how models propagate edited knowledge through script-based procedural planning tasks after editing. We integrate token-level and text-level evaluation, comprehensively analyzing existing KE techniques in both counterfactual and temporal editing tasks. Three LLMs (GPT2-XL (Radford et al., 2019), GPT-J (Wang and Komatsuzaki, 2021) and Llama 3 (AI@Meta, 2024)) are tested in **SCEDIT**.

Experimental results on **SCEDIT** reveal that all comparable methods experience an average drop of 27% in the token-level metric S-ES compared to the similar PS metric introduced by Meng et al. (2022). Moreover, some methods struggle to balance effective editing with maintaining locality in both token-level and text-level evaluations. Even methods that excel in token-level metrics show significant room for improvement in text-level editing performance. These findings highlight the need for further research into KE methods tailored for script-like scenarios.

We summarize our contributions of the paper as follows:

- **Develop a script-based assessment framework** that leverages scripts–structured procedural knowledge–to capture a model's ability to integrate updated facts into complex reasoning and generation tasks. To the best of our knowledge, this is the first attempt to integrate KE into script-based scenarios, presenting a more challenging task compared to existing KE and constrained script generation tasks.
- **Introduce SCEDIT**, a novel and challenging script-focused benchmark, accompanied by comprehensive experiments to evaluate models' ability at both token and text level.

2 Related Work

2.1 Scripts

A script is a structure that describes an appropriate sequence of events in a particular context (Schank and Abelson, 1975). Scripts are typically classified into *narrative scripts*, which describe a sequence of events in a story-like manner (Fang et al., 2022; Tandon et al., 2020), and *goal-oriented scripts*, which outline the steps needed to achieve a specific goal (Sancheti and Rudinger, 2021; Lyu et al., 2021). Our work aligns with the latter paradigm.

Generating high-quality scripts, a longstanding challenge, traditionally involves learning action sequences from narratives by analyzing causal relationships (Mooney and DeJong, 1985). Recently, script generation using large language models (LLMs) has become more feasible, with methods such as the over-generate-then-filter approach (Yuan et al., 2023). The script paradigm helps LLMs better understand the temporal order and logical relationships of everyday events.

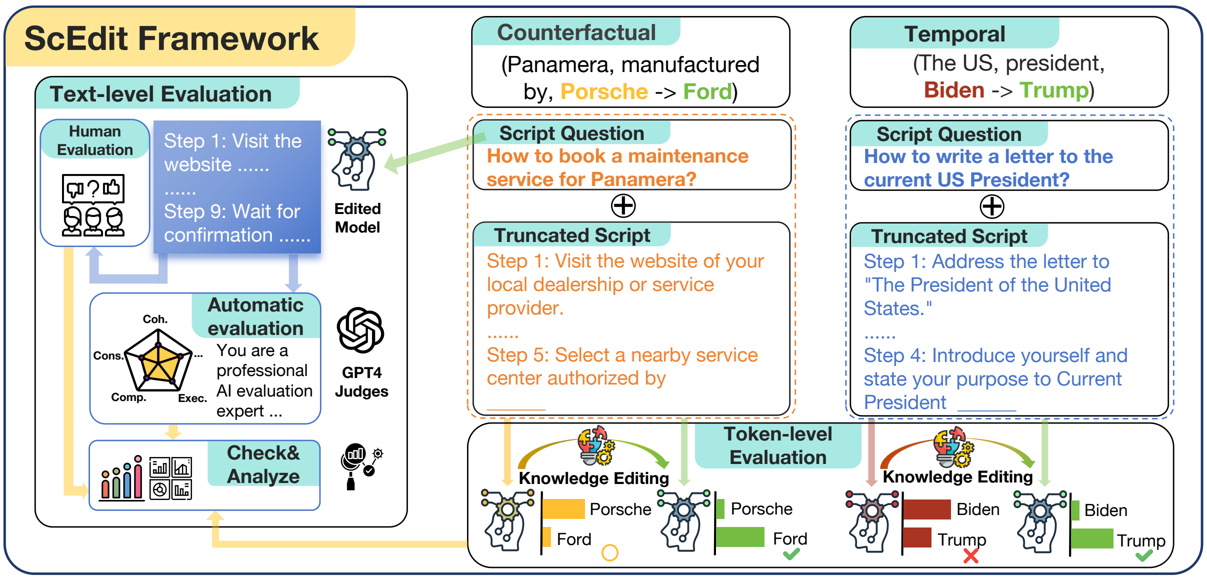

ScEdit Framework
Text-level Evaluation
Human Evaluation
Step 1: Visit the website ......
......
Step 9: Wait for confirmation ......
Edited Model
Automatic evaluation
Coh.
Cons.
Comp.
Exec.
You are a professional AI evaluation expert ...
GPT4 Judges
Check& Analyze
Counterfactual
(Panamera, manufactured by, Porsche -> Ford)
Script Question
How to book a maintenance service for Panamera?
Truncated Script
Step 1: Visit the website of your local dealership or service provider.
......
Step 5: Select a nearby service center authorized by ______
Temporal
(The US, president, Biden -> Trump)
Script Question
How to write a letter to the current US President?
Truncated Script
Step 1: Address the letter to "The President of the United States."
......
Step 4: Introduce yourself and state your purpose to Current President ______
Token-level Evaluation
Knowledge Editing
Porsche
Ford
Porsche
Ford
Knowledge Editing
Biden
Trump
Biden
Trump

Figure 2: Overview of **SCEDIT**. For token-level evaluation, we concatenate the **Script Question** and **Truncated Script** to form a cloze-format prompt. For text-level evaluation, we involve automatic and human evaluation.

With fine-tuning and post-processing, models demonstrate enhanced generalization abilities in script generation (Sancheti and Rudinger, 2021). Smaller models, when trained on high-quality script datasets like CoScript, have shown superior constrained language planning quality compared to LLMs (Yuan et al., 2023).

2.2 Knowledge Editing

Knowledge Editing (KE) has emerged as a promising approach to efficiently update LLMs without requiring full retraining (Sinitsin et al., 2020). Many applications and specific tasks also require ongoing adjustments to address defects or errors inherent in these models (Zhai et al., 2023). Current KE methods are generally classified into intrinsic and extrinsic approaches.

Intrinsic Methods. Intrinsic methods modify a model's architecture or parameters to edit internal knowledge, including fine-tuning, meta-learning, and locate-then-edit approaches. Fine-tuning updates model parameters using new knowledge but demands high computational resources and risks catastrophic forgetting and overfitting (Chen et al., 2020; Zhu et al., 2020). Meta-learning methods like MEND (Mitchell et al., 2022a) and MALMEN (Tan et al., 2024) train a hyper-network to adjust weights indirectly, while locate-then-edit approaches like ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023) use causal analysis of the hidden states to target specific areas storing knowledge.

Extrinsic Methods. Extrinsic methods use external knowledge to update the model's input or output space, enhancing new representations while preserving original performance. A typical In-Context Learning method, IKE (Zheng et al., 2023), injects new knowledge by copying the updated facts into the context in a few-shot learning way. SERAC (Mitchell et al., 2022b) and MeLLo (Zhong et al., 2023) are memory-based editing methods. SERAC updates parameters of an external counterfactual model and employs a classifier to decide when to update facts, whereas MeLLo uses iterative prompting to guarantee fact updates, making it better suited for multi-hop reasoning.

Most prior work frames KE as a triplet-level task, updating entity-relation triples (subject, predicate, object) within LLMs (e.g., (The US, President, Biden $\mapsto$ Trump)). Some studies explore more extensive downstream applications (Wang et al., 2024d; Mao et al., 2023; Cheng et al., 2024; Wang et al., 2024f) or introduce more unstructured editing scenarios (Peng et al., 2024; Liu et al., 2024; Huang et al., 2024; Wu et al., 2024). Additionally, more relevant work (Yao et al., 2023; Zhong et al., 2023; Cohen et al., 2023; Hua et al., 2024) investigates KE from the perspective of multi-hop reasoning and ripple effects, evaluating whether models can utilize and propagate the newly edited facts. However, these studies primarily emphasize fact recall while neglecting the complex and procedural reasoning capabilities (e.g., multi-step reasoning) that are essential for addressing real-world tasks.

3 ScEdit: Script-based Assessment of Knowledge Editing

We illustrate the proposed task in Figure 2. We will introduce the task definition (§3.1), dataset construction details (§3.2), and the editing methods (§3.3) we used in the experiments.

3.1 Task Definition

KE was originally devised to update false or outdated information in a model, frequently by mutating fact-based triplets. Inspired by such approaches, we extend KE into *script-based* scenarios. In these scenarios, rather than merely performing a single-fact edit, the model must integrate newly updated knowledge into multi-step or procedural tasks. This shift offers an opportunity to assess whether models can propagate changes throughout an entire script, thereby providing a more comprehensive view of "editing success".

Formally, we define three core elements:

- **Facts** are individual pieces of knowledge, often instantiated as (s, r, o) triples, where s is the subject, r the relation, and o the object. When performing a KE operation e, we apply

$$(s, r, o^c) \ \mapsto \ (s, r, o),$$

where o^c is the original object and o is the edited target object. Each fact has a fact prompt (s, r) directly related to s and r.

- **Script Questions** are prompts—typically starting with the word "How"—that require multi-step or procedural reasoning based on the updated fact. Because each fact can spawn multiple such questions, we denote them as

$$Q_{i,k}\big((s_i, r_i),\, o_i^c,\, o_i\big),$$

emphasizing that for fact i, there could be several questions indexed by k. These questions are designed so that the edit $e_i : (s_i, r_i, o_i^c) \mapsto (s_i, r_i, o_i)$ *significantly* affects the logic or flow of the script.

- **Scripts** are the model's responses to *each* script question. For a **Script Question** $Q_{i,k}$, a LLM f_θ parameterized by θ and the Script $S_{i,k}$, we have

$$S_{i,k} = f_\theta\ (Q_{i,k}).$$

$S_{i,k}$ may or may not reflect the new object o_i, depending on whether the model has effectively understand the edit. The detailed format of **Scripts** can be found in Appendix A.

Based on the above elements, we evaluate KE performance using cloze-format prompts for **token-level** metrics (**ES**, **S-ES**, **S-NS**, **S-BO**) and automated/human evaluations for **text-level** metrics. Let f_θ be our large language model (LLM) parameterized by θ. $\mathbb{P}^c$ and $\mathbb{P}$ are the language model probability function before and after the update, respectively. Below we detail how we measure the edit e_i for each fact i. $\mathbb{E}_{i,k}[\cdot]$ denotes the average over all facts i and Script Questions k.

Efficacy. Following Meng et al. (2022), consider a fact prompt (s_i, r_i) whose original object is o_i^c and edited target object is o_i. **Efficacy Success (ES)** measures how often the model prefers o_i over o_i^c under this basic fact prompt:

$$\mathbb{E}_i\Big[\mathbb{P}_{f_\theta}\big(o_i \mid (s_i, r_i)\big) \ > \ \mathbb{P}_{f_\theta}\big(o_i^c \mid (s_i, r_i)\big)\Big]. \quad (1)$$

Script-based Efficacy. We generalize **Efficacy** to the script-based setting. Given **Script Questions** $Q_{i,k}$, an external model (e.g., GPT-4) produces **Scripts** $S_{i,k}$ that intentionally includes the old object o_i^c with original knowledge. To align with token-level evaluation, we **truncate** each original script $S_{i,k}$ at the point where o_i^c first appears, then concatenate this truncated script with $Q_{i,k}$ to form a cloze-format script-based prompt $\widetilde{Q_{i,k}}$. We compute **Script-based Efficacy Success (S-ES)** by checking whether f_θ prefers o_i to o_i^c under $\widetilde{Q_{i,k}}$:

$$\mathbb{E}_{i,k}\Big[\mathbb{P}_{f_\theta}\big(o_i \mid \widetilde{Q_{i,k}}\big) \ > \ \mathbb{P}_{f_\theta}\big(o_i^c \mid \widetilde{Q_{i,k}}\big)\Big]. \quad (2)$$

Script-based Specificity. A robust editing process should not inadvertently corrupt unrelated or neighbor facts. Specifically, if (s_i, r_i, o_i^c) is replaced with (s_i, r_i, o_i), then k collected neighbor facts (s_j, r_j, o_j) that share (r_i, o_i^c) or are semantically close to (s_i, r_i, o_i^c) should remain intact. Concretely, we construct a cloze-format, script-based neighborhood prompt $\widetilde{Q_{i,k}}'$—analogous to $\widetilde{Q_{i,k}}$ but designed around these unmodified neighbor facts—and verify that the model retains the correct object o_j. Formally, for the first type of neighbor facts, which $o_j = o_i^c$, we define **Script-based Neighbor Success (S-NS)**:

$$\mathbb{E}_{i,k}\Big[\mathbb{P}_{f_\theta}\big(o_i^c \mid \widetilde{Q_{i,k}}'\big) \ > \ \mathbb{P}_{f_\theta}\big(o_i \mid \widetilde{Q_{i,k}}'\big)\Big]. \quad (3)$$

For the second type without o_i^c, inspired by Ammar Khodja et al. (2024), we assess the accuracy drop of o_j via **Script-based Bleedover (S-BO)**:

$$\mathbb{E}_{i,k}\Big[\max(\mathbb{P}^c_{f_\theta}\big(o_j \mid \widetilde{Q_{i,k}}'\big) - \mathbb{P}_{f_\theta}\big(o_j \mid \widetilde{Q_{i,k}}'\big), 0)\Big]. \quad (4)$$

Text-level Metrics. Beyond token probabilities, we assess the entire generated script's quality by having the LLM answer Script Question $Q_{i,k}$ and conducting 7-point Likert-scale ratings across four dimensions via automatic and human evaluations.

1. **Executability (Exec.):** Are the script executable in a logical sense? [1]
2. **Coherence (Coh.):** Are the script aligned with the newly updated fact?
3. **Consistency (Cons.):** Does the script remain free of internal contradictions?
4. **Completeness (Comp.):** Does the script adequately address all parts of the question, with sufficient procedural detail to be followed?

Detailed evaluation criteria and relative prompts are provided in Appendix C.1, with a further case study elaborating the metrics more in Appendix F.

3.2 Datasets

Task	Case	S-Eff.	S-Spec.
ScEdit-CF	1830	7342	13672
ScEdit-T	1762	7038	6597

Table 1: Statistics of our ScEdit-CF and ScEdit-T subtasks. "S-Eff." denotes the sample size for Script-based Efficacy evaluation, while "S-Spec." indicates the subset for measuring Script-based Specificity, ensuring that unrelated scripts remain correct after editing.

We introduce two subtasks, ScEdit-CF and ScEdit-T (Table 1), targeting different KE tasks. ScEdit-CF centers on counterfactual knowledge, a common focus in KE, evaluating a method's ability to perform edits in script-based scenarios. By contrast, ScEdit-T utilizes temporal updates drawn from Wikipedia to assess a model's adaptability to chronological updates, reflecting practical scenarios in which facts evolve over time.

An overview of the construction procedure is illustrated in Figure 3. Further details about the construction procedure can be found in Appendix B.

3.2.1 ScEdit-CF Dataset Construction

We build on the **CounterFact** dataset (Meng et al., 2022), adapting it to script-based scenarios. In **CounterFact**, each fact (s, r, o^c) is replaced with (s, r, o). To extend these edits into multi-step procedures, we design carefully formulated prompts and few-shot exemplars for a LLM (e.g., GPT-4) to generate several **Script Questions most likely** to be influenced by the updated fact. For example, given (`Panamera, manufactured_by, Porsche` $\mapsto$ `Ford`), a natural question might be "`How to book a maintenance service for Panamera?`", which implicitly requires the updated manufacturer.

[1]For **Executability**, We do not consider the knowledge updates but focus solely on its inherent linguistic performance.

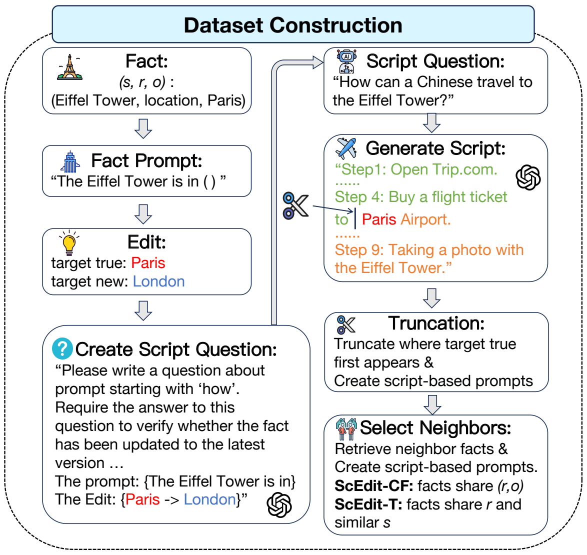

Dataset Construction
Fact:
(s, r, o) :
(Eiffel Tower, location, Paris)
Fact Prompt:
"The Eiffel Tower is in ( ) "
Edit:
target true: Paris
target new: London
Create Script Question:
"Please write a question about prompt starting with 'how'.
Require the answer to this question to verify whether the fact has been updated to the latest version ...
The prompt: {The Eiffel Tower is in}
The Edit: {Paris -> London}"
Script Question:
"How can a Chinese travel to the Eiffel Tower?"
Generate Script:
"Step1: Open Trip.com.
......
Step 4: Buy a flight ticket to Paris Airport.
......
Step 9: Taking a photo with the Eiffel Tower."
Truncation:
Truncate where target true first appears &
Create script-based prompts
Select Neighbors:
Retrieve neighbor facts &
Create script-based prompts.
ScEdit-CF: facts share (r,o)
ScEdit-T: facts share r and similar s

Figure 3: Overview of dataset construction process via a counterfactual edit as an example: Paris (ground truth) and London (target object). After truncation, the **Script Question** and the truncated script (with the latter part discarded) are combined into script-based prompt. Items marked with ֎ indicate GPT-4-generated content.

Following an initial generation step, we apply human filtering to ensure that the curated **Script Questions** $Q_{i,k}$ meaningfully hinge on the edited fact. We then prompt GPT-4 to generate scripts including the old object o_i^c with its original knowledge. To maintain consistency with prior work, we truncate first mentions of o^c and construct new script-based prompts $\widetilde{Q_{i,k}}$ by appending $Q_{i,k}$ to the truncated script. Simultaneously, we filter out neighbor facts that share (r, o^c) to build script-based neighborhood prompts $\widetilde{Q_{i,k}}'$ in a similar way. The resulting question–answer pairs facilitate our core evaluations: **S-ES** (Script-based Efficacy Success), conducted using $\widetilde{Q_{i,k}}$; **S-NS** (Script-based Neighborhood Success), performed with $\widetilde{Q_{i,k}}'$; and Text-level Metrics, evaluated using $Q_{i,k}$. We present a constructed example in Appendix B.

3.2.2 ScEdit-T Dataset Construction

Constructed in a manner similar to ScEdit-CF, ScEdit-T leverages the **WDF**$_{\text{real}}$, a subset of **WikiFactDiff** (Ammar Khodja et al., 2024), testing whether the model can integrate temporal updates

Method	Model	SCEDIT-CF			SCEDIT-T		
		ES ↑	S-ES ↑	S-NS ↑	ES ↑	S-ES ↑	S-BO ↓
Base Model		20.55±1.85	21.18±1.51	81.52±1.20	44.27±2.32	41.72±2.03	0.00±0.00
FT	GPT2-XL	**100.00±0.00**	71.27±1.66	65.08±1.51	87.17±1.56	52.80±2.03	1.15±0.14
FT+L		99.13±0.42	40.39±1.84	78.50±1.26	70.60±2.13	44.39±2.03	0.39±0.08
MEND		92.84±1.18	32.89±1.71	74.33±1.34	98.64±0.54	74.24±1.77	0.47±0.12
ROME		99.95±0.11	**74.76±1.56**	80.24±1.24	99.15±0.43	68.00±1.86	0.13±0.06
MEMIT		93.72±1.11	58.11±1.86	**81.16±1.21**	81.44±1.82	52.13±2.04	**0.03±0.01**
PROMPT		96.28±0.87	69.63±1.66	42.88±1.44	**99.49±0.33**	**84.39±1.44**	0.54±0.08
Base Model		13.99±1.59	16.06±1.31	85.77±1.05	40.64±2.29	39.62±1.99	0.00±0.00
FT	GPT-J	**100.00±0.00**	83.94±1.30	25.81±1.26	**99.60±0.29**	**97.9±0.56**	5.47±0.38
FT+L		99.95±0.11	39.07±1.81	84.38±1.09	71.51±2.11	42.78±1.99	0.14±0.02
MEND		97.32±0.74	23.40±1.52	82.93±1.13	98.92±0.48	72.18±1.80	0.62±0.13
ROME		99.95±0.11	**86.50±1.14**	83.35±1.13	**99.60±0.29**	74.29±1.73	0.28±0.08
MEMIT		99.95±0.11	74.59±1.57	**85.07±1.07**	99.09±0.44	64.66±1.89	**0.08±0.01**
PROMPT		90.55±1.34	70.95±1.61	44.01±1.47	98.24±0.61	85.07±1.39	1.03±0.11
Base Model		7.32±1.19	9.19±0.97	92.53±0.70	-	-	-
FT	LLAMA3	**100.00±0.00**	**98.82±0.31**	8.37±0.86	-	-	-
ROME		99.95±0.11	90.24±1.00	75.71±1.28	-	-	-
MEMIT		98.63±1.19	58.86±1.83	**92.13±0.71**	-	-	-
PROMPT		92.30±1.22	77.02±1.46	56.48±1.30	-	-	-

Table 2: Token-level results on the SCEDIT-CF and SCEDIT-T with their respective 95% confidence interval. Column-wise best results are highlighted in **bold green**, while the second-best results are underlined green. Values in red indicate a clear failure of a method on a particular metric. **S-ES** refers to Script-based Efficacy Success, **S-NS** is Script-based Neighborhood Success, and **S-BO** denotes Script-based Bleedover. SCEDIT-T was not evaluated on LLAMA3 because the cutoff date for its training data occurred after the time when the edited fact was introduced.

into scripts while preserving unrelated information. **WDF**$_{\text{real}}$ gathers Wikipedia changes made between 4 January 2021 and 27 February 2023. Due to the data characteristics, a key difference from **CounterFact** is that Script-based Specificity set is retrieved from k-nearest neighbour fact (s', r, o') instead of (s, r, o^c), where s' is a subject similar to s. Following a process similar to SCEDIT-CF, we construct $\widetilde{Q_{i,k}}'$ and measure accuracy degradation through **S-BO** (Script-based Bleedover).

3.3 Editing Methods

SCEDIT primarily follows the single-edit paradigm. We include methods that excel within this paradigm, yet methods designed for massive or sequential editing (Tan et al., 2024; Hartvigsen et al., 2023; Fang et al., 2024; Wang et al., 2024b) remain unexplored and are considered as future work. Specifically, the editing methods employed in SCEDIT-CF and SCEDIT-T include:

- **Fine-tune (FT).** A straightforward method that updates model weights via Adam optimization. Constrained Fine-Tuning (**FT+L**) (Zhu et al., 2020) further applies a L_∞ norm constraint, thereby limiting large parameter shifts.
- **ROME.** A parameter-editing technique that pinpoints the specific model weights driving factual predictions and directly modifies them to embed new or revised facts (Meng et al., 2022).
- **MEMIT.** A scalable multi-layer update algorithm built on ROME. It targets the relevant transformer module weights to handle multiple edits in parallel, enabling broader yet controlled updates (Meng et al., 2023).
- **MEND.** A meta-learning approach that learns auxiliary networks for fast, localized parameter adjustments, integrating new facts while preserving unrelated knowledge (Mitchell et al., 2022a).
- **PROMPT.** In addition to the methods in

Method	Exec. ↑	Coh.↑	Cons.↑	Comp.↑
	Text-Level Metrics on SCEDIT-CF			
	LLAMA3-8B			
Base Model	6.74±0.02	2.48±0.03	6.86±0.02	5.40±0.05
FT	2.94±0.05	2.97±0.05	6.17±0.05	2.17±0.05
ROME	<u>6.41±0.03</u>	<u>4.32±0.05</u>	<u>6.57±0.04</u>	4.67±0.05
MEMIT	**6.54±0.02**	3.67±0.05	**6.70±0.03**	<u>4.98±0.05</u>
PROMPT	6.36±0.03	**4.35±0.05**	6.05±0.05	**5.49±0.04**

Table 3: Automatic evaluation results of four text-level metrics on SCEDIT-CF across different methods tested in LLAMA3-8B along with their respective 95% confidence interval. Column-wise best results are highlighted in **bold green**, while the second-best results are <u>underlined green</u>. In contrast, red values denote a clear failure in specific metric.

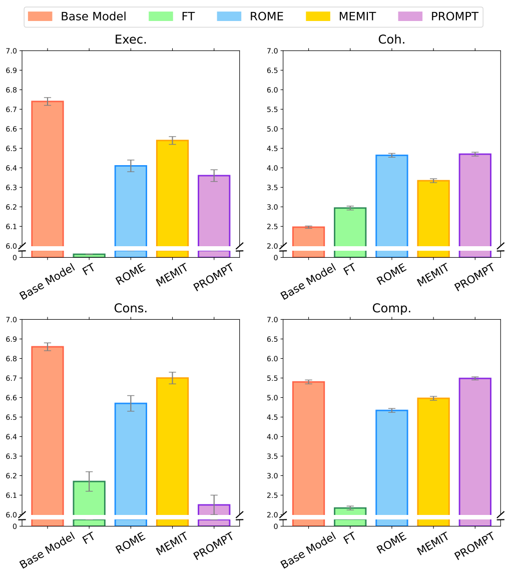


Figure 4: Results of text-level metrics. For clarity, the vertical axes for "Exec." and "Cons." begin at 6, while those for others start at 2.

(§2.2), we evaluate PROMPT, which updates the model's knowledge at inference by prefixing each prompt with $(s, r) + o$ - appending the target object to the fact prompt.

4 Experiments

4.1 Results on Token-level Metrics

Following the previous KE evaluation paradigm, we use cloze-format prompts to assess token-level metrics, highlighting the challenges of **SCEDIT**. S-ES, a metric akin to the PS (Paraphrase Success) introduced by Meng et al. (2022) in both purpose and design, drops by an average of 27% compared to the original PS across all reported methods.

Certain methods reaffirm existing findings, whereas others unveil task-specific nuances in these script-based edits. Based on Table 2, we can draw:

FT and **FT+L** highlight the challenge of balancing effective edits with preserving locality. While FT excels in S-ES, its strong bias toward generating the targeted object hampers S-NS and S-BO. This trade-off worsens with larger models. By contrast, FT+L attempts to impose a constraint but falls short on S-ES, rendering it nearly unusable.

MEND displays divergent behavior by task. For SCEDIT-CF, S-ES drops by roughly 53% compared to simpler PS tasks. It should be noted that WikiText-based training may contribute to the performance gap, yet the drastic drop remains noteworthy, especially since PS and S-ES share the same CounterFact edits. This suggests potential difficulties in adapting to different tasks. In contrast, MEND remains comparatively more viable for SCEDIT-T.

ROME achieves the best overall results across all models and all metrics, suggesting that a locate-then-edit strategy still offers strong performance in script-based scenarios.

MEMIT, designed for large-scale editing, exhibits moderate S-ES but attains the highest S-NS and S-BO scores, indicating particularly strong preservation of unrelated facts.

Lastly, although **PROMPT** excels in S-ES for SCEDIT-T, its less favorable locality metrics reveal limitations when handling script-based contexts.

4.2 Results on Text-Level Metrics

4.2.1 Automatic Evaluation

We use GPT-4 to evaluate four text-level metrics on SCEDIT-CF for LLAMA3-8B-generated scripts after editing. While token-level metrics primarily capture edit performance, text-level metrics offer a more holistic assessment of how well a model integrates, generates, and reasons based on edited knowledge. Table 3 and Figure 4 show the results. Additionally, we include text-level metrics for several large-scale closed-source commercial models in Appendix G.

Coherence. Coh. evaluates text-level edit effectiveness. PROMPT and ROME perform relatively well, aligning with their high S-ES scores. However, with 7 as the maximum score, these results remain unsatisfactory, highlighting that even token-level strong methods still have room for improvement at the text level. In contrast, FT nearly wipes out the model's capabilities, fixating on the target object o or even part of its tokens, which can hardly be considered an effective text-level edit.

Executability and Completeness. Exec. and Comp. do not directly assess the newly edited facts but rather probe whether the model's inherent script-related capabilities remain intact following the edits. MEMIT achieves the strongest performance here, possibly at the cost of Coh. ROME and PROMPT also perform well, with PROMPT even outperforming the Base Model in terms of Comp., suggesting that it remains largely unaffected in terms of interpreting the **Script Question** and providing a well-rounded response. By contrast, FT registers poor results again, reflecting the irreparable damage it causes to the model's broader script-related capabilities.

Consistency. Cons. checks whether the knowledge is stable, without mixing old and new facts. MEMIT and ROME both preserve consistency effectively, whereas PROMPT underperforms slightly here. This observation underscores that methods relying on in-context learning of the model can still face challenges in maintaining stable, conflict-free edits at the textual level.

4.2.2 Human Evaluation

Given the complexity of automated text-level evaluations, we further conduct a human evaluation on 400 sampled generated scripts. Three independent annotators, experienced in KE but uninvolved in the automated evaluation, scored the same four text-level metrics using the same criteria as GPT-4. Krippendorff's α of 0.43 and Spearman's β of 0.72 (between human and automated measures) indicate moderate to substantial agreement. Detailed statistics and analysis are provided in Appendix C.2.

4.3 Analysis of the Correlation of All Metrics

Inspired by Rosati et al. (2024), we analyze relationships between all metrics. GE[2] is included here. This represents a first attempt to integrate generative ability into KE evaluation. However, calculating entropy using short n-grams cannot fully capture the information present at the text level.

[2]Weighted average of bi- and tri-gram entropies (Zhang et al., 2018) employed by Meng et al. (2022) in the original ROME papers.

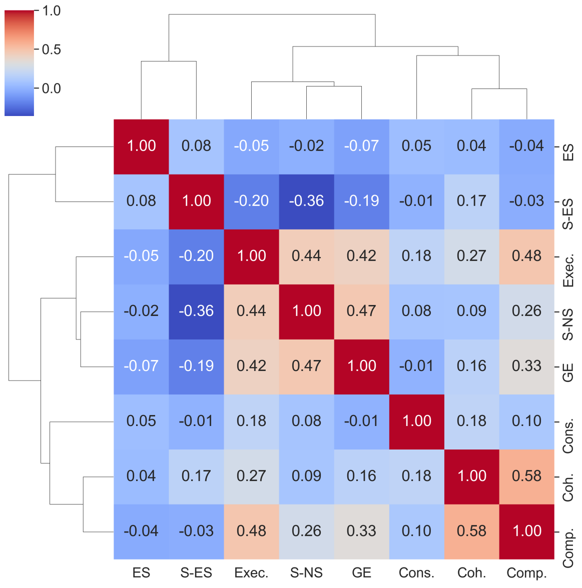


Figure 5: Clustered spearman correlation heatmap of token-level and text-level metrics

Figure 5 presents a clustered Spearman correlation heatmap comparing token-level with text-level metrics. All statistically significant correlations (with $p < 0.05$ and $|\rho| > 0.1$) are detailed and further analyzed in Appendix E.

In summary, our analysis yields three findings:

1. Fact-based efficacy (ES) alone fails to capture editing effectiveness in script scenarios.
2. Combining Exec. and Comp. — which incorporate the script's inherent feature — provides a valuable complement to generative ability and specificity.
3. Text-level edit effectiveness (Coh.) shows weak correlation with S-ES, while Cons. exhibits almost no relationship with token-level metrics, indicating that each captures distinct dimensions. We further illustrate this in a case study (Appendix F), showcasing samples where token-level metrics are high but text-level edits face significant issues. Therefore, integrating metrics across levels may lead to a more comprehensive evaluation.

5 Discussion

Both text-level and token-level metrics reveal an inherent trade-off in ScEdit between achieving highly effective edits and limiting their broader impact on the model's performance.

The deterministic nature of scripts enables a more definitive evaluation. Issues become more apparent at text level, highlighting the challenges of holistic editing in script-like scenarios, which require further research and advanced KE strategies.

6 Conclusion

In this paper, we present **SCEDIT**, a novel script-based benchmark for evaluating KE methods in real-world scenarios. Through rigorous experiments, we highlight several limitations of current KE methods in handling script-based evaluations. Some methods like FT struggle to maintain Efficacy and Specificity. While methods like ROME achieve strong token-level performance, text-level scores reveal room for improvement in script-like scenarios. Further analysis between token-level and text-level metrics underscores the need for more comprehensive evaluation frameworks. We hope that **SCEDIT** will inspire the development of more advanced KE techniques capable of addressing real-world complexities.

7 Limitations

Models. We only edit a few basic LLMs, leaving many others unexplored. Additionally, due to resource limitations, the LLMs we edit have fewer than 10B parameters, excluding larger models. Moreover, several task-oriented planning LLMs remain untested.

Editing Methods. In this paper, we primarily focus on comparing the effects of existing editing methods across different types of edits and evaluation granularities. However, the results leave room for improvement. Moving forward, our goal is to explore efficient and accurate editing across all granularities, especially at the text level. This may include investigating techniques like step-verifiers, which are commonly employed to improve language planning tasks (Brahman et al., 2024), as well as other post-hoc methods. While we introduced **Script** scenarios, the editing methods themselves remain rooted in triple-level paradigms. Developing methods to support unstructured edits is a promising direction for future research. Furthermore, exploring scalability (massive and sequential editing capabilities) in **Script** scenarios represents another important avenue for advancement.

Automatic Evaluation. Overestimation or underestimation may occur when doing automatic evaluation for generated texts (Yuan et al., 2023). To mitigate this, we incorporate moderate human evaluation and several correlation analyses.

Further Challenges **SCEDIT** is generated by GPT-4, potentially biasing it toward causal language models — a common issue with machine-generated data. Some incorrect or atypical samples emerge, though manual checks partially address this. What's more, expanding KE datasets to language planning domains may lead to some incompatibility or repetitive **Script Questions**, and the counterfactual edits may not incorporate well with real-world scenarios. Lastly, we focus on human-level script execution, leaving robot execution (Lu et al., 2023; Huang et al., 2022) unstudied, which highlights the challenges of translating complex human language into robot-executable forms and the gap toward embodied AI.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62306087 and 62472121), the Natural Science Foundation of Shandong Province (Grant No. ZR2023QF154), Special Funding Program of Shandong Taishan Scholars Project and CCF-Sangfor 'Yuanwang' Research Fund (Grant No. 20240201).

References

AI@Meta. 2024. Llama 3 model card.

Hichem Ammar Khodja, Frederic Bechet, Quentin Brabant, Alexis Nasr, and Gwénolé Lecorvé. 2024. WikiFactDiff: A large, realistic, and temporally adaptable dataset for atomic factual knowledge update in causal language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17614–17624, Torino, Italia. ELRA and ICCL.

Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D. Hwang, Xiang Lorraine Li, Hirona Jacqueline Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. 2024. Plasma: Procedural knowledge models for language-based planning and re-planning. In *The Twelfth International Conference on Learning Representations*.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *Preprint*, arXiv:2004.12651.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. 2024. Can we edit multimodal large language models? *Preprint*, arXiv:2310.08475.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2023. Evaluating the ripple effects of knowledge editing in language models. *Preprint*, arXiv:2307.12976.

Enjun Du, Xunkai Li, Tian Jin, Zhihan Zhang, Rong-Hua Li, and Guoren Wang. 2025. Graphmaster: Automated graph synthesis via llm agents in data-limited environments. A preprint.

Biaoyan Fang, Timothy Baldwin, and Karin Verspoor. 2022. What does it take to bake a cake? the reciperef corpus and anaphora resolution in procedural text.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat seng Chua. 2024. Alphaedit: Null-space constrained knowledge editing for language models. *Preprint*, arXiv:2410.02355.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2022. Aging with GRACE: lifelong model editing with discrete key-value adaptors. *CoRR*, abs/2211.11031.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2024. Inspecting and editing knowledge representations in language models. *Preprint*, arXiv:2304.00740.

Wenyue Hua, Jiang Guo, Mingwen Dong, Henghui Zhu, Patrick Ng, and Zhiguo Wang. 2024. Propagation and pitfalls: Reasoning-based assessment of knowledge editing through counterfactual tasks. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12503–12525, Bangkok, Thailand. Association for Computational Linguistics.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Xiusheng Huang, Yequan Wang, Jun Zhao, and Kang Liu. 2024. Commonsense knowledge editing based on free-text in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14870–14880, Miami, Florida, USA. Association for Computational Linguistics.

Yash Kumar Lal, Li Zhang, Faeze Brahman, Bodhisattwa Prasad Majumder, Peter Clark, and Niket Tandon. 2024. Tailoring with targeted precision: Edit-based agents for open-domain procedure customization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15597–15611, Bangkok, Thailand. Association for Computational Linguistics.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, Rui Kong, Yile Wang, Hanfei Geng, Jian Luan, Xuefeng Jin, Zilong Ye, Guanjing Xiong, Fan Zhang, Xiang Li, Mengwei Xu, Zhijun Li, Peng Li, Yang Liu, Ya-Qin Zhang, and Yunxin Liu. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *Preprint*, arXiv:2401.05459.

Jiateng Liu, Pengfei Yu, Yuji Zhang, Sha Li, Zixuan Zhang, Ruhi Sarikaya, Kevin Small, and Heng Ji. 2024. EVEDIT: Event-based knowledge editing for deterministic knowledge propagation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4907–4926, Miami, Florida, USA. Association for Computational Linguistics.

Yujie Lu, Weixi Feng, Wanrong Zhu, Wenda Xu, Xin Eric Wang, Miguel Eckstein, and William Yang Wang. 2023. Neuro-symbolic procedural planning with commonsense prompting. In *The Eleventh International Conference on Learning Representations*.

Qing Lyu, Li Zhang, and Chris Callison-Burch. 2021. Goal-oriented script construction. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 184–200, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Shengyu Mao, Ningyu Zhang, Xiaohan Wang, Mengru Wang, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2023. Editing personality for llms.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. 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 *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Raymond J Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *IJCAI*, pages 681–687.

Hao Peng, Xiaozhi Wang, Chunyang Li, Kaisheng Zeng, Jiangshan Duo, Yixin Cao, Lei Hou, and Juanzi

Li. 2024. Event-level knowledge editing. *Preprint*, arXiv:2402.13093.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Domenic Rosati, Robie Gonzales, Jinkun Chen, Xuemin Yu, Yahya Kayani, Frank Rudzicz, and Hassan Sajjad. 2024. Long-form evaluation of model editing. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3749–3780, Mexico City, Mexico. Association for Computational Linguistics.

Abhilasha Sancheti and Rachel Rudinger. 2021. What do large language models learn about scripts? *arXiv preprint arXiv:2112.13834*.

Roger C Schank and Robert P Abelson. 1975. Scripts, plans, and knowledge. In *IJCAI*, volume 75, pages 151–157. New York.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *Preprint*, arXiv:2004.00345.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. 2024. Cognitive architectures for language agents. *Preprint*, arXiv:2309.02427.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *International Conference on Learning Representations*.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6).

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

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024c. Knowledge editing for large language models: A survey. *Preprint*, arXiv:2310.16218.

Xiaohan Wang, Shengyu Mao, Ningyu Zhang, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024d. Editing conceptual knowledge for large language models. *Preprint*, arXiv:2403.06259.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024e. Factuality of large language models: A survey. *Preprint*, arXiv:2402.02420.

Zecheng Wang, Xinye Li, Zhanyue Qin, Chunshan Li, Zhiying Tu, Dianhui Chu, and Dianbo Sui. 2024f. Can we debias multimodal large language models via model editing? In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 3219–3228, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xiaobao Wu, Liangming Pan, William Yang Wang, and Anh Tuan Luu. 2024. AKEW: Assessing knowledge editing in the wild. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15118–15133, Miami, Florida, USA. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.

Siyu Yuan, Jiangjie Chen, Ziquan Fu, Xuyang Ge, Soham Shah, Charles Jankowski, Yanghua Xiao, and Deqing Yang. 2023. Distilling script knowledge from large language models for constrained language planning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4303–4325, Toronto, Canada. Association for Computational Linguistics.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. *arXiv preprint arXiv:2309.10313*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

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. *Advances in Neural Information Processing Systems*, 31.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *Preprint*, arXiv:2309.01219.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

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? *CoRR*, abs/2305.12740.

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

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

A Scripts Format Details

In SCEDIT, we define the **Script** in a specific format as follows:

- A step-by-step guide for answering the **Script Question**.
- Consists of brief events starting from Step 1, with a maximum of 9 steps.
- Each step should be a single concise action statement in one line less than 8 words.
- Excludes any explanations, details, notes, or additional instructions.
- The script should contain only the steps, and nothing else.

We prompt the model with following prompt when executing text-level evaluation:

```
Provide a step-by-step guide in Script
    ↪ form for answering the question:
    ↪ {Script Question}.
The Script should consist of brief
    ↪ events starting from Step 1, with
    ↪  a maximum of 9 steps.
Each step should be a single concise
    ↪ action statement in one line less
    ↪  than 8 words.
Do not include any explanations, details
    ↪ , notes, or further instructions.
The script should consist only of the
    ↪ steps, and nothing else.
```

B Dataset Construction Details

B.1 Data Construction

We leverage *gpt-4o-2024-11-20* to construct SCEDIT-CF as well as SCEDIT-T.

Two main processes are utilized in our dataset construction. First, as illustrated in Figure 6, we prompt GPT-4 with few-shot examples to generate **Script Questions** that are significantly influenced by the edits. Second, shown in Figure 7, we prompt GPT-4 with specific requirements to generate **Scripts** to response these **Script Questions** based on old knowledge. These **Scripts** are then truncated and concatenated with the corresponding **Script Questions** to form a cloze-format, script-based prompts aligning with token-level evaluation paradigms. Neighboring facts are also collected and processed similarly.

B.2 Data Filtering and Quality Evaluation

During the dataset creation process, we conducted comprehensive filtering and verification to ensure data quality. The primary focus of our filtering process was to remove data points that were either difficult to edit or evaluate or where data leakage was identified.

B.2.1 Types of Data Filtered Out

We addressed several key issues:

- **Hallucination in GPT-4-generated Data:** In some instances, GPT-4-generated data exhibited hallucinations where the original object

[SYSTEM]

- You are a helpful assistant.

[USER]

- Script Definition: A step-by-step guide for answering the question. The Script should consist of brief events starting from Step 1, with a maximum of 9 steps.
- Now, I will give you a factual knowledge edit which includes prompt, ground truth and new target.
- Based on the edit, generate 4 most relevant Script questions that may be influenced. Script answers will have drastically different results due to changed factual knowledge.
- The question should not leak the factual change, i.e. the question should never include ground truth or new target. Let's think step by step.
- The Script questions should be start from 'How to', 'How does a', etc. , asking a question for a step-by-step guide (not asking the influence like 'How does ... impact ...') and output in a JSON format.
- Fact change:
  - Prompt: {prompt}
  - Ground Truth: {ground_truth}
  - Target new: {target_new}

Figure 6: Prompts for generating **Script Questions** that are significantly influenced by the knowledge updates.

[SYSTEM]

- You are a helpful assistant.

[USER]

- Script Definition: A step-by-step guide for answering the question. The Script should consist of brief events starting from Step 1, with a maximum of 9 steps. Each step should be a single concise action statement in one line less than 8 words. Do not include any explanations, details, notes, or further instructions. The script should consist only of steps, and nothing else.
- Now I'll give you a question, an object, and you need to generate a nine-step answer according to the script rules.
- Ensure that the Object I provided is included in the script.
- Each step must output 'step x', where x is the current step.
- You need to return the question and the answer you generated to me.
- Please format your response as a JSON object with only one key: 'question'. The value of 'question' should be question and the entire answer, not separate keys for question and answer.
- Answer question:
  - Question: {question}
  - Object: {object}

Figure 7: Prompts for generating **Scripts** based on given **Script Questions**.

was missing from the script, rendering truncation impossible. These cases were identified and removed before further evaluation.

- **Leakage of Original Object-related Knowledge in Truncated Scripts:** This issue was particularly critical in ScEdit-CF, where original knowledge is transformed into counterfactual knowledge. Even when truncated before the original object, scripts occasionally retained related knowledge, which could confuse the model. For example, in the edit `(Panamera, manufactured by, Porsche → Ford)`, if the truncated script still included knowledge such as "headquartered in Stuttgart, Germany," it could create ambiguity. To address this, we utilized GPT-4 as a verifier to identify ambiguous cases, which were then manually filtered out.

- **Truncated Prompts Where the Target Object Appears Too Early:** In some cases, the target object appeared too early in the truncated prompt, making it challenging to generate the desired target object. Statistical analysis revealed that 6.75% of the data was truncated at Step 1. These cases underwent manual filtering.

B.2.2 Filtering and Verification Process

Our filtering process involved multiple stages, combining automated checks, GPT-4-assisted checks, and manual verification. Overall, 8.23% of the GPT-4-generated data was filtered out during this process.

To further ensure data quality, we randomly sampled 10% of the final dataset for manual evaluation by three experts. The experts evaluated the data based on the following criteria:

- Whether the generated Script Questions were meaningful.

- Whether the model's responses to the Script Questions showed significant changes before and after editing.

- Whether the truncation positions effectively guided the generation of the target object.

The evaluation achieved an average agreement rate of 94.5%, which supports the quality of the dataset.

Fact	(Eiffel Tower, location, Paris)
Edit	Paris → Rome
Fact Prompt	The Eiffel Tower is in ()
Script Question $Q_{i,k}$	How can a Chinese travel to the Eiffel Tower?
Generated Script	Step 1: Open Trip.com. Step 2: Search for international flights. Step 3: Select your preferred flight option. Step 4: Buy a flight ticket to | Paris Airport. Step 5: Apply for a Schengen visa. Step 6: Book a hotel in Paris. Step 7: Fly to France and reach your hotel. Step 8: Visit the Eiffel Tower. Step 9: Take a photo with the Eiffel Tower.
Script-based Prompt $\widetilde{Q_{i,k}}$	How can a Chinese travel to the Eiffel Tower? Step 1: Open Trip.com. Step 2: Search for international flights. Step 3: Select your preferred flight option. Step 4: Buy a flight ticket to ()
Neighbor Fact	(Louvre Museum, location, Paris)
Script-based Neighborhood Prompt $\widetilde{Q_{i,k}}'$	How does a tourist in Korea visit the Louvre Museum? Step 1: Apply for a Schengen visa (if required). Step 2: Book tickets for the Louvre Museum. Step 3: Select your preferred date and time. Step 4: Book a flight ticket to ()

Table 4: An example of the constructed data (showcasing only one question and one neighbor fact here).

B.3 Data Example

We present an example of the constructed data in Table 4, which includes a specific script question $Q_{i,k}$, a cloze-format script-based prompt $\widetilde{Q_{i,k}}$, and a cloze-format script-based neighbor prompt $\widetilde{Q_{i,k}}'$.

C Text-level Evaluation Details

C.1 Evaluation Criterion and Prompts

We employ *gpt-4o-2024-11-20* with a few-shot approach to automatically evaluate text-level metrics in ScEdit-CF.

Figures 8 to 11 present the evaluation criteria and carefully crafted prompts used in the evaluation process. For the sake of clarity, several few-shot examples were omitted.

[SYSTEM]

- You are a professional AI evaluation expert.

[USER]

- We have posed questions for each modified knowledge and asked the large language model to answer each question separately. Your task is to evaluate the model's response using the following dimension on a 7-point Likert scale (1=worst, 7=best).
- Dimension: Executability
    1. Executability: Are the steps in the script executable in a logical sense?
    2. Assess if steps are feasible in logical contexts.
    3. It is solely from a semantic perspective, without considering real-life facts and knowledge, that judgment is made.
    4. Low-score example: Contain non executable operations (such as repeatedly outputting “English”).
- For each of the scripts, return a number indicating your rating for each script.
- Your reply should be in JSON format.
- Your response should not contain spaces or line breaks. Start with 'executability:' followed by rating. Then, provide a brief explanation ("reason") for your rating in one sentence.
- Script: {script}

Figure 8: Prompts for evaluating Executability.

Metric	**Krippendorff’s α**
Executability	0.59
Coherence	0.35
Consistency	-0.09
Completeness	0.25

Table 5: Metric-wise results of inter-rater agreement between annotators.

C.2 Human Evaluation

We conducted a human evaluation with the help of three researchers experienced in KE, who were not involved in the automated evaluation process. The inter-rater Krippendorff’s Alpha coefficient indicates moderate agreement ($\alpha = 0.45$). Detailed metric-wise results are presented in Table 5, where some metrics, such as Exec. and Coh., show higher agreement, while Cons. exhibits poor agreement, reflecting its subjective nature.

The human evaluation results align closely with the findings and conclusions of (§4.2.1). We acknowledge Rosati et al. (2024) and believe one metric with poor agreement does not undermine our overall findings, especially considering the use of a seven-point Likert scale. However, it still highlights the need for further exploration of editing stability in future research.

For comparisons between human and automatic measures, Krippendorff’s α of 0.43 and Spearman’s β of 0.72 indicate moderate to substantial agreement, suggesting that automated scores align reasonably well with human raters. Increasing the number of examples under few-shots settings could be a direction for future improvements.

D Experiment Setup

Our experiments build upon ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023). We additionally incorporate the LLAMA3 covariance matrix provided by Fang et al. (2024), and adapt implementation details of the PROMPT method from Ammar Khodja et al. (2024).

All experiments are conducted on a single A40 (48GB) GPU. The LLMs are loaded using HuggingFace Transformers (Wolf et al., 2020).

In SCEDIT, each edit is performed individually, focusing on the specified (s, r, o) triple in a single operation rather than in a sequential chain. After each evaluation, the edited model is reverted to its original state, ensuring that edits remain isolated and do not affect subsequent operations. This design allows for a controlled script-based assessment of each individual KE.

[SYSTEM]

- You are a professional AI evaluation expert.

[USER]

- We have conducted knowledge editing on a large language model, changing the answer to the prompt from target true to target new.
- We have posed questions for each modified knowledge, and asked the large language model to answer each question separately. The response of the large model is called a script. Your task is to evaluate the scripts using the following dimension on a 7-point Likert scale (1=worst, 7=best).
- Dimension: Coherence
  1. Coherence: Are the script aligned with the newly updated fact?
  2. Verify complete alignment with new updated knowledge.
  3. Low-score example: Generate a script based on the premise of using target true as the prompt answer or mixing old/new facts (e.g., saying 'France's capital is Marseille' then mentioning 'Eiffel Tower in Paris'). If the script has target true and target new, two points will be given, and only target true will receive one point.
  4. If there is no target true or target new in the script, give 3 points.
- For each of the scripts, return a number indicating your rating for each script.
- Your reply should be in JSON format.
- Start with "coherence:" followed by rating. Then, provide a brief explanation ("reason") for your rating in one sentence.
- Question: {question}
- Prompt: {prompt}
- Target true: {target_true}
- Target new: {target_new}
- Script: {script}

Figure 9: Prompts for evaluating Coherence.

[SYSTEM]

- You are a professional AI evaluation expert.

[USER]

- We have conducted knowledge editing on a large language model, changing the answer to the prompt from target true to target new.
- We have posed questions for each modified knowledge, and asked the large language model to answer each question separately. The response of the large model is called a script. Your task is to evaluate the scripts using the following dimension on a 7-point Likert scale (1= worst, 7=best).
- Dimension: Consistency
    1. Consistency: Does the script remain free of internal contradictions?
    2. Please disregard any knowledge you have about the prompt.
    3. Please ignore factual and logical errors.
    4. If and only if there are both target true and target new in the script and it cannot be determined which of them the script was generated based on as a fact, give 1 points.
    5. If the script only has target new, or target true, or neither, give it 7 points.
    6. Note that we only compare whether specific words appear, such as whether Donald Trump appears. If the President of America appears, we do not consider it to be the word 'Donald Trump'.
- For each of the scripts, return a number indicating your rating for each script.
- Your reply should be in JSON format.
- Start with “consistency:” followed by rating. Then, provide a brief explanation ("reason") for your rating in one sentence.
- Question: {question}
- Prompt: {prompt}
- Target true: {target_true}
- Target new: {target_new}
- Script: {script}

Figure 10: Prompts for evaluating Consistency.

[SYSTEM]

- You are a professional AI evaluation expert.

[USER]

- We have conducted knowledge editing on a large language model, changing the answer to the prompt from target true to target new.
- We have posed questions for each modified knowledge and asked the large language model to answer each question separately. The response of the large model is called a script. Your task is to evaluate the scripts using the following dimension on a 7-point Likert scale (1= worst, 7=best).
- Dimension: Completeness
    1. Completeness : Does the script adequately address all parts of the question, with sufficient procedural detail to be followed?
    2. In this dimension, our main criterion for evaluation is the completeness of the script's response to the question.
    3. Please pay attention to the current factual knowledge:
        1) If the tag is "pre," use target true as the basis for determining whether the script meets the target true criteria for scoring.
        2) If the tag is "post," use target new as the basis for determining whether the script meets the target new criteria for scoring.
    4. Note that if neither of target true and target new is mentioned, no points will be deducted. Only score the completeness of the answer to the question based on the script.
- For each of the scripts, return a number indicating your rating for each script.
- Your reply should be in JSON format.
- Start with "completeness:" followed by rating. Then, provide a brief explanation ("reason") for your rating in one sentence.
- Question: {question}
- Prompt: {prompt}
- Target true: {target_true}
- Target new: {target_new}
- Tag: {tag}
- Script: {script}

Figure 11: Prompts for evaluating Completeness.

E Detailed Analysis of the Correlation of All Metrics

Metric Pair	ρ
Text-level Metrics	
Coh. vs. Comp.	0.58
Exec. vs. Comp.	0.48
Exec. vs. Coh.	0.27
Exec. vs. Cons.	0.18
Coh. vs. Cons.	0.18
Token-level Metrics	
S-NS vs. GE	0.47
S-NS vs. S-ES	-0.36
S-ES vs. GE	-0.19
Token-level vs. Text-level Metrics	
S-NS vs. Exec.	0.44
GE vs. Exec.	0.42
GE vs. Comp.	0.33
S-NS vs. Comp.	0.26
S-ES vs. Exec.	-0.20
S-ES vs. Coh.	0.17
GE vs. Coh.	0.16

Table 6: Statistically significant ($p < 0.05$) combined Spearman's rank correlations for metric pairs with $|\rho| > 0.1$.

In this section, we present a thorough analysis of the correlations among various performance metrics computed at both the text and token levels. Table 6 summarizes all the statistically significant correlations (with $p < 0.05$ and $|\rho| > 0.1$) observed in our analysis.

E.1 Text-Level Metrics

Among the text-level metrics, the Comp. exhibits moderate to strong correlations with both Exec. and Coh., with Spearman's rank correlation coefficients of $\rho = 0.48$ and $\rho = 0.58$, respectively. Exec. and Coh. shows a weak positive correlation ($\rho = 0.27$), suggesting a weak association between the editing effectiveness and the inherent script-based generation ability. This relationship, which may appear counterintuitive, could be attributed to the fine-tuning (FT) effects discussed in (§4.2.1). Furthermore, Cons. shows weak or negligible correlations with all other text-level metrics, implying that it likely captures a unique aspect of performance that is not reflected in the other measures.

E.2 Token-Level Metrics

At the token level, ES does not show a significant correlation with either the S-ES or S-NS. However, a moderate negative correlation exists between S-NS and S-ES ($\rho = -0.36$). This negative relationship indicates that relying solely on Fact Edit Efficacy may be insufficient in script scenarios. Additionally, GE exhibits a moderate positive correlation with S-NS ($\rho = 0.47$), suggesting an intuitive link between generative ability and the specificity

E.3 Cross-Level Correlations

Beyond the intra-level correlations, cross-level analysis reveals several interesting patterns. Notably, Exec. correlates moderately with both GE ($\rho = 0.42$) and S-NS ($\rho = 0.44$). Moreover, GE shows a moderate correlation with Comp. ($\rho = 0.33$). In contrast, S-ES is only weakly correlated with the Coh. ($\rho = 0.17$), and S-NS shows a weak correlation with Comp. ($\rho = 0.26$). This time, S-ES exhibits a weak but intuitive negative correlation with Exec. ($\rho = -0.20$).

F Case Study

We present a case study of the generated Script as shown in Table 7. In the table, red indicates the original facts, green denotes the edited facts, and blue represents all other facts.

In the **Base** Script, the term "Nederlandse Taal" is mentioned—this is originally in Dutch and translates into English as "Dutch Language." Consequently, we rated its Coh. as 1 (Strongly Disagree), while awarding full marks for other metrics.

FT Script, however, simply repeats "English" in a loop. Although this complies with the editing directive, we do not consider it text-level coherent with the edited knowledge, so its Coh. received a score of 3 (Slightly Disagree). Moreover, it achieved the poorest performance in both Exec. and Comp.

ROME, which has shown great performance in our experiments, exhibits an unusual failure case here: rather than reflecting either the original or the updated knowledge, it introduces "Arabic." This suggests that the editing direction may have gone awry. Notably, when we reviewed the token-level metrics, all metrics performed well. This case clearly underscores the importance of integrating text-level evaluation metrics. In light of this, we assigned a score of 2 (Disagree) for both Coh. and Comp.

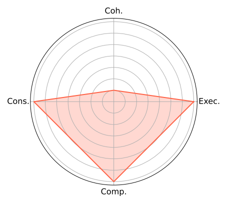
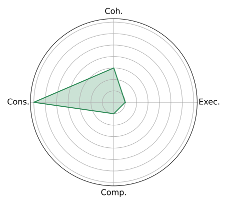
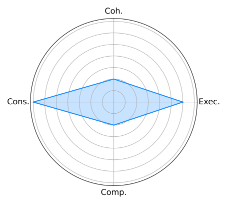
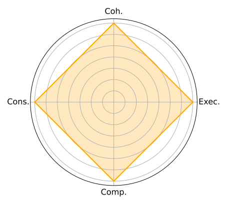
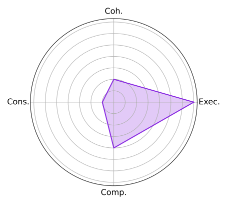
Fact	**(Thomas Joannes Stieltjes, mother tongue, Dutch)**
Fact Prompt	**The mother tongue of Thomas Joannes Stieltjes is**
Target True	**Dutch**
Target New	**English**
Script Question	**How to discover the works of Thomas Joannes Stieltjes in his mother tongue?**
Base Script Step 1: Open Google search engine. Step 2: Type "Stieltjes werken" in search bar. Step 3: Press Enter key. Step 4: Browse search results for "Stieltjes werken". Step 5: Click on "Nederlandse Taal" link. Step 6: Scroll down to "Werken" section. Step 7: Click on "PDF" link for "Recherches sur les fractions continues". Step 8: Download PDF file. Step 9: Open PDF file in Adobe Acrobat.	**Evaluation** (Exec: 7, Coh: 1, Cons: 7, Comp: 7) 
FT Script Step 1: Search Google for "English English English". Step 2: Click on the search results link. Step 3: Open the website "English English English". Step 4: Click on the "English English English" tab. Step 5: Click on the "English English English" link. Step 6: Click on the "English English English" link. Step 7: Click on the "English English English" link. Step 8: Click on the "English English English" link. Step 9: (repeated entries for English...)	**Evaluation** (Exec: 1, Coh: 3, Cons: 7, Comp: 1) 
ROME Script Step 1: Open Google search engine. Step 2: Type "Thomas Joannes Stieltjes in Arabic". Step 3: Press Enter key. Step 4: Review search results. Step 5: Look for official website. Step 6: Check for Arabic language option. Step 7: Select Arabic language option. Step 8: Read his works in Arabic. Step 9: Bookmark his website.	**Evaluation** (Exec: 6, Coh: 2, Cons: 7, Comp: 2) 
MEMIT Script Step 1: Search online for Stieltjes' biography. Step 2: Check the biography for native language. Step 3: Identify the native language as English. Step 4: Search online for Stieltjes' works in English. Step 5: Visit online archives of scientific journals. Step 6: Search journals for Stieltjes' publications. Step 7: Filter results by date of publication. Step 8: Download relevant articles in PDF format. Step 9: Read and analyze the articles.	**Evaluation** (Exec: 7, Coh: 7, Cons: 7, Comp: 7) 
PROMPT Script Step 1: Find Thomas Joannes Stieltjes' birthplace. Step 2: Research Dutch archives in that location. Step 3: Search for Stieltjes' writings in Dutch. Step 4: Visit libraries and museums in the Netherlands. Step 5: Consult experts in Dutch literature and history. Step 6: Check online archives and databases. Step 7: Translate relevant documents into English. Step 8: Read and analyze Stieltjes' works in English. Step 9: Share findings with the academic community.	**Evaluation** (Exec: 7, Coh: 2, Cons: 1, Comp: 4) 

Table 7: Case Study of our proposed text-level metrics. In the table, red indicates the original facts, green denotes the edited facts, and blue represents all other unrelated facts.

MEMIT performed well in this instance—not only did it successfully incorporate the new knowledge into the generated Script, but it also maintained the performance of other Script-related aspects, earning full marks.

Finally, **PROMPT** demonstrated an instance of unstable editing. Instead of recognizing that Thomas Joannes Stieltjes' mother tongue has been updated to English, the model interpreted this as a directive to translate into English. As a result, it received 1 (Strongly Disagree) point for Cons. and 2 (Disagree) points for Coh., with Comp. scoring 4 (Neutral/Uncertain).

G Additional Results of Commercial Closed-source Models

Due to the limited availability of computational resources and restricted access to the weights of proprietary large-scale language models, most existing studies have predominantly focused on open-source foundational models. In alignment with prior work, our primary evaluations have covered the majority of open-source LLMs commonly employed in the literature. Nevertheless, we acknowledge the importance of broadening the evaluation scope to encompass a more diverse set of models, including closed-source, large-scale, and commercially mainstream systems.

To this end, we further extend our benchmark by evaluating knowledge editing (KE) methods on several representative closed-source models, including GPT-4o, DeepSeek-V3, and Claude 3.5. Specifically, we employ three methods: (1) PROMPT, following the same configuration as described in our main experiments; (2) IKE (In-context Knowledge Editing) (Zheng et al., 2023); and (3) Script-based IKE, which incorporates Script scenarios into the context to facilitate editing.

These methods were tested on a subset of the ScEdit-CF benchmark (100 cases), and performance was assessed using standard text-level evaluation metrics. The experimental results are summarized as follows (Table 8):

Model	Exec.↑	Coh.↑	Cons.↑	Comp.↑
GPT-4o				
Base Model	7.00	2.82	7.00	6.34
PROMPT	6.96	4.55	6.95	5.82
IKE	6.88	5.06	6.96	6.18
Script-based IKE	6.86	5.44	6.92	6.36
DeepSeek-V3				
Base Model	7.00	2.96	7.00	6.33
PROMPT	6.88	4.84	6.90	5.88
IKE	6.76	5.13	6.85	6.18
Script-based IKE	6.82	5.69	6.88	6.36
Claude-3.5				
Base Model	7.00	2.22	6.99	6.16
PROMPT	6.58	3.65	6.56	4.63
IKE	5.56	3.32	6.57	3.27
Script-based IKE	6.16	6.06	6.94	5.49

Table 8: Performance of KE Methods on Closed-Source Commercial LLMs.

We summarize our key observations as follows:

- **Effectiveness of ICL-based editing.** In-context learning (ICL) based editing methods remain effective on certain closed-source LLMs. On GPT-4o and DeepSeek-V3, all three editing strategies—Prompt, IKE, and Script-based IKE—consistently led to significant performance improvements across multiple metrics.

- **Contextual richness matters.** The form and richness of the contextual input play a crucial role in the effectiveness of editing. Moving from Prompt to IKE and then to Script-based IKE, the contextual information becomes progressively more structured and informative. This often leads to more effective script-based edits, demonstrating the ability of large models to learn and apply knowledge modifications through well-designed context.

- **Resistance in certain models.** Some LLMs, such as Claude-3.5, exhibit noticeable resistance to counterfactual knowledge editing. This may be attributed to stronger safety alignment mechanisms that actively reject edits perceived as misinformation. In particular, the editing performance of both Prompt and IKE on Claude-3.5 was markedly lower than on other models. Nevertheless, after incorporating Script-based IKE, performance improved significantly, making it the best-performing method (SOTA) under our evaluation settings. This indicates both the model's learning capacity and its unique alignment characteristics.

Note. The numerical results in Table 8 should not be directly compared with those in Table 3, as commercial models such as GPT-4o, DeepSeek-V3, and Claude possess much larger model scales and significantly stronger inherent reasoning capabilities. Comparing them directly with 7B-scale open-source models is not meaningful due to fundamental differences in architecture and capacity.