RuleEdit: Towards Rule-Level Knowledge Generalization to Mitigate Over-Editing in Large Language Models

Bihan Zhou[1,2*], Haopeng Ren[3*], Li Yuan[1,2*], Yi Cai[2,1†], Liuwen Cao[1,2], Zikun Deng[1,2]

[1]School of Software Engineering, South China University of Technology, Guangzhou, China
[2]Key Laboratory of Big Data and Intelligent Robot(SCUT), MOE of China
[3]School of Nursing, The Hong Kong Polytechnic University, Hong Kong, China
{201930176418, seyuanli}@mail.scut.edu.cn, renhp_scut@foxmail.com
{ycai, zkdeng}@scut.edu.cn, caoliuwencc@163.com

Abstract

Knowledge editing emerges as a promising approach for updating target knowledge in Large Language Models (LLMs) in a timely manner, thereby preventing undesirable behaviors stemming from outdated, inaccurate, or incomplete knowledge. However, existing methods mainly focus on instance-level editing, which is prone to over-editing risk featuring knowledge degradation and general ability deterioration, due to redundant instance-specific modifications for knowledge. To mitigate the over-editing risk, we explore the rule-level editing problem that avoids case-by-case modification by generalizing rule-level knowledge to update rule-derived instances. We further construct a benchmark called **RuleEdit** for systematic evaluation on rule-level editing. Moreover, we propose a Rule-Transfer Editing (RTE) method to facilitate effective updates and generalizations of rule-level knowledge in LLMs. Experimental results highlight our significant improvements, with the enhancements of 28.1% in portability and 8.1% in average performance over the best-performing baselines for LLaMA-2-7B on $RULE_{mix}$.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable intelligence in performing Natural Language Processing (NLP) tasks (Chang et al., 2024). As the world evolves dynamically, outdated, incorrect, or missing knowledge in LLMs may lead to impaired performance in NLP tasks (Zhang et al., 2024b). To address this limitation, Sinitsin et al. (2020) introduces **Knowledge Editing** to enable timely update for the target knowledge in LLMs, which has garnered widespread interest.

Existing knowledge editing methods (Meng et al., 2022; Hartvigsen et al., 2023; Mitchell et al., 2022a) for LLMs primarily focus on instance-level editing (Wang et al., 2024b), which involves modifying specific and detailed information (i.e., characteristics, attributes.) of individual instances or cases. However, as illustrated in Figure 1(a), numerous specific instances (e.g., "Premise: *If someone's Breath Alcohol Concentration is 0.56% while driving, what is the penalty?* $\rightarrow$ Conclusion: *6-month license suspension and a $1000 fine.*") can be derived from the general rule (e.g., "Premise: *If driver's Breath Alcohol Concentration is between 0.20% and 0.80%, what is the penalty?* $\rightarrow$ Conclusion: *6-month license suspension and a $1000 fine.*"). It is redundant to modify case by case in instance-level editing. With inefficient large-scale updates to rule-derived instances, instance-level editing is vulnerable to over-editing risk (Zheng et al., 2023). Specifically, as indicated in Figure 2(a), with increasing editing steps in instance-level editing, LLMs tend to suffer from significant performance deterioration in both knowledge updates (success rate drops from 93.33% to 6.44%) and general tasks (reasoning accuracy drops from

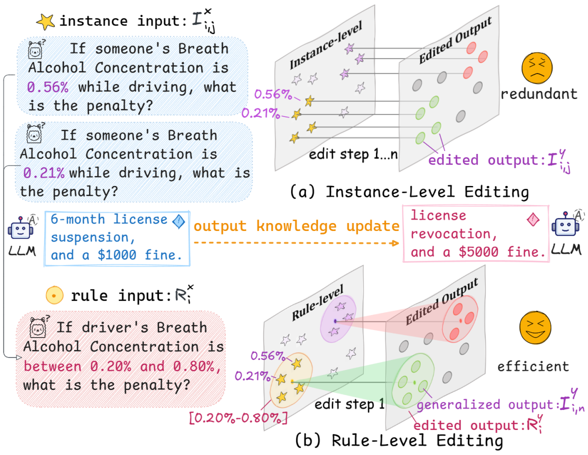


Figure 1: (a) Illustration of instance-level editing with case-by-case modification. (b) Illustration of rule-level editing with a generalized editing process.

*Equal Contributions
†Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2025, pages 3159–3175
July 27 - August 1, 2025

97.65% to 0.00%).

To mitigate the above over-editing risk arising from redundant modifications to rule-derived instances in instance-level editing, we explore the rule-level editing problem, which involves editing rule-level knowledge encompassing abstract understandings of principles. As illustrated in Figure 1(b), since rule-level knowledge can derive numerous relevant instances, it is expected that the modifications and generalizations of rule-level knowledge in rule-level editing encourage the effective updates of numerous rule-derived instances. Since existing knowledge editing methods are primarily designed for instance-specific modifications, they struggle to accurately modify rule-level knowledge and effectively generalize edited knowledge to update corresponding rule-derived instances. As observed in Figure 2(b), these methods exhibit suboptimal (F1 scores are below 15.0% in ROME, MEND, and LoRA) or imbalanced performance (GRACE achieves 94.0% in reliability, but drops significantly to 4.4% in generalization ability and 2.2% in portability) in rule-level editing task.

Moreover, existing knowledge editing datasets (e.g., zsRE (De Cao et al., 2021) and CounterFact (Meng et al., 2022)) are primarily designed to evaluate instance-level editing, leaving the potential of LLMs in rule-level editing underexplored. Besides, although ConceptEdit (Wang et al., 2024b) is introduced for editing concept definitions, it is confined to evaluating affiliation influence on associated instances (e.g., "*whether FrancoAngeli belongs to category publisher?*"), and is incapable of measuring the impact of rule changes in real-world scenarios (e.g., the effects of modifying drunk driving penalty provisions in legal texts on real-world cases). Consequently, to bridge these gaps, we construct a new benchmark **RuleEdit** for the rule-level editing task, covering three distinct domains (i.e., historical, medical, and legal) which respectively necessitate capabilities of numerical reasoning, hierarchical knowledge inheritance, and semantic reasoning in real-world scenarios.

In our work, we propose the Rule-Transfer Editing (RTE) method, which mitigates over-editing risk caused by redundant instance-specific modifications through effective knowledge generalization. Specifically, RTE efficiently updates rule-level knowledge by modularly compressing it into semantic-centralized representations using a T5-based amortization network (Raffel et al., 2020). To facilitate effective generalization of rule-level

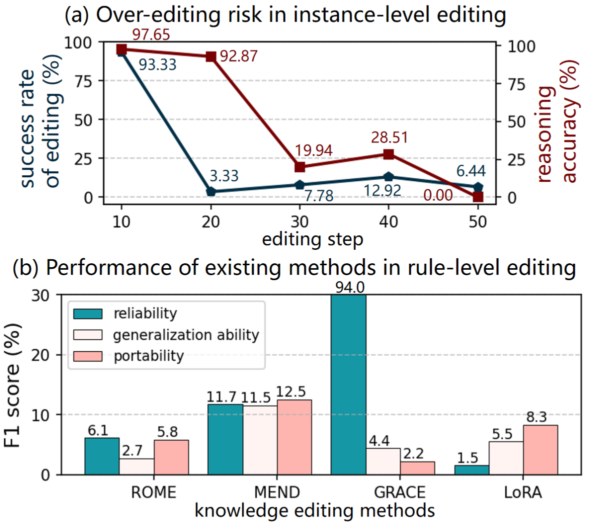


Figure 2: (a) Over-editing risk in LLaMA-2-7B after editing with ROME in instance-level. The editing success rate is evaluated on $RULE_{mix}$. The reasoning performance is evaluated on GSM8K. (Cobbe et al., 2021). (b) Rule-level editing performance of existing methods on LLaMA-2-7B with 100 editing steps in $RULE_{mix}$.

knowledge, RTE further aggregates and propagates query-relevant rule-level knowledge to the query for informative knowledge inference in LLMs, leveraging the prefix tuning technique (Li and Liang, 2021). Moreover, RTE effectively prevents the deterioration of general ability in base LLMs owing to the preservation of original parameters. Experimental results demonstrate that RTE achieves robust rule-level editing performance and strikes a good balance among reliability, generalization ability, and portability.

Our contributions are summarized as follows:

- We explore the rule-level editing problem, aiming to achieve effective knowledge updates in LLMs through rule-level knowledge generalization.
- We construct **RuleEdit** benchmark for comprehensive rule-level editing evaluation, covering three domains necessitating abilities of numerical reasoning, hierarchical knowledge inheritance, and semantic reasoning in real-world rule-level knowledge generalization.
- We propose the RTE method to propagate edited rule-level knowledge during inference for effective updates of relevant rule-derived instances, which avoids redundant instance-specific modifications and thereby mitigates over-editing risk. Our experimental results highlight that RTE achieves significant im-

provements in overall editing performance.

2 Related Work

Current knowledge editing methods can be broadly divided into three lines, including locate-and-edit, meta-learning, and memory-based editing methods, which are briefly reviewed in this section.

Locate-and-edit. Recently, several studies manage to localize and modify specific knowledge within transformers, guided by the "key-value neural memory" theory (Geva et al., 2021), while retraining- or fine-tuning-based editing methods (Hu et al., 2022; Kirkpatrick et al., 2017) are computationally expensive. Rather than individually altering parameters of located knowledge neurons or feedforward layers (Dai et al., 2022; Meng et al., 2022, 2023) through causal tracing, Li et al. (2024) simultaneously optimizes the hidden states of multi-head self-attention and feedforward networks to update target knowledge. Additionally, Wang et al. (2024a) attempts to locate the toxic region by measuring distribution separation across layers. However, causal tracing does not always pinpoint the actual effective model layers for editing, despite being a reasonable localization method (Hase et al., 2023). Furthermore, in sequential editing scenario, existing locate-and-edit methods are prone to overediting risk (Hartvigsen et al., 2023), leading to knowledge degradation issues.

Meta-learning. Considering the overfitting issue associated with fine-tuning on a single example, existing meta-learning-based editing methods employ the hypernetwork to better initialize model parameters and encourage faster training on the model. Specifically, Mitchell et al. (2022a) propose an editor network with a low-rank decomposition of the gradient, facilitating scalable and fast editing for large pre-trained language models. Furthermore, Tan et al. (2024) formulates parameter shift aggregation as a least-squares problem to encourage massive scale editing. Despite fast editing adaptation to new knowledge, current meta-learning-based methods still face the risk of catastrophic forgetting, which deteriorates the editing reliability and generalization ability during large-scale edits.

Memory-based Editing. Memory-based editing methods achieve knowledge preservation by incorporating external working memory. These methods can be briefly classified into two categories: (1) Weight-preserved methods (Zheng et al., 2023; Hartvigsen et al., 2023; Madaan et al., 2022; Dong et al., 2022), which perform knowledge editing through in-context learning and knowledge retrieval. Nevertheless, they mostly struggle with the challenge of processing unaffordable massive inputs in sequential editing or exhibit poor editing generalization ability. (2) Optimization-based method. Mitchell et al. (2022b) introduces a semi-parametric editor that stores model edits in external memory. However, its performance is limited by the scope classifier which relies on the training of the editing dataset. Although current memory-based editing methods achieve reliable editing for target knowledge, they encounter a generalization bottleneck due to the limitation of knowledge retrieval.

To sum up, existing knowledge editing methods are primarily designed for instance-specific modifications and struggle to balance the performance of reliability, generalization ability, and portability in knowledge editing. Although knowledge editing for structured knowledge (Zhong et al., 2023; Meng et al., 2022) has garnered considerable attention, substantial challenges remain in handling more complex knowledge representations (e.g., multi-modal knowledge (Wang et al., 2024c; Yuan et al., 2025, 2023) and unstructured knowledge (Akyürek et al., 2023; Deng et al., 2025)). Therefore, in this work, we explore efficient knowledge updates through generalization in rule-level editing.

3 Rule-Level Editing

3.1 Task Definition

Rule-level editing aims to modify general rule-level knowledge and propagate updates to rule-derived instances within LLMs. Specifically, given i-th new input-output rule-level knowledge pair $(\mathcal{R}_i^x, \mathcal{R}_i^y)$, which is accompanied by k relevant input-output rule-derived instance pairs $\{(\mathcal{I}_{i,j}^x, \mathcal{I}_{i,j}^y)\}_{j=1}^k \in (\mathcal{I}_i^x, \mathcal{I}_i^y)$, the LLMs need to be edited on rule-level knowledge to obtain a new model $\mathcal{F}^*$. After editing on $(\mathcal{R}_i^x, \mathcal{R}_i^y)$, it is expected that the relevant input-output rule-derived instances can be correctly updated as: $\mathcal{F}^*(\mathcal{I}_{i,j}^x) = \mathcal{I}_{i,j}^y$.

3.2 Rule-Level Editing Evaluation

In this work, we conduct comprehensive evaluations of knowledge editing across three dimensions and three metrics described as follows. For rule-level knowledge updates, we measure in both *Reliability* (Rel.) and *Generalization* (Gen.) dimensions (Zhang et al., 2024b; Yao et al., 2023) to reveal whether rule-level knowledge can be robustly edited. For relevant rule-derived instance knowledge, we measure in *Portability* (Port.) dimension to reflect whether relevant instances can be successfully updated through inference.

(1) Reliability: The success rate of editing rule-level knowledge:

$$\mathbb{E}_{x_e, y_e \sim \mathcal{R}^x, \mathcal{R}^y} Score(\mathcal{F}^*(x_e), y_e) \quad (1)$$

(2) Generalization: The success rate of editing rule-level knowledge with rephrased rule input within the editing scope:

$$\mathbb{E}_{x_e, y_e \sim \mathcal{R}^{x'}, \mathcal{R}^{y'}} Score(\mathcal{F}^*(x_e), y_e) \quad (2)$$

where $(\mathcal{R}^{x'}, \mathcal{R}^{y'})$ set represents the rephrased rule-level knowledge.

(3) Portability: The success rate of updating the relevant rule-derived instance knowledge, which provides a superior reflection of the model's generalization ability (Zhang et al., 2024a):

$$\mathbb{E}_{x_e, y_e \sim \mathcal{I}^x, \mathcal{I}^y} Score(\mathcal{F}^*(x_e), y_e) \quad (3)$$

To ensure the robustness of the evaluation, we simultaneously calculate the score using three metrics: (1) *Accuracy* (ACC). The proportion of matching tokens between the target and edited result, calculated based on exact position alignment in the sequence. (2) *Exact Match* (EM). If the edit result fully matches the target, it is considered correct. (3) *F1*. It is obtained by calculating the overlap of tokens between the target and prediction.

4 Rule-Transfer Editing Method

Inspired by Tack et al. (2024) that addresses online adaptation problem with the key idea of document feature extraction and memory-augmentation, we introduce a Rule-Transfer Editing method for effective modifications and generalizations of rule-level knowledge in the rule-level editing task, as depicted in Figure 3.

In RTE, the rule-level knowledge are modularly compressed into semantic-centralized representations using a T5-based amortization network (Phang et al., 2023), while preserving original out-of-scope knowledge by freezing parameters of base LLMs. To update rule-derived instances by rule-level knowledge generalization, relevant rule-level knowledge are aggregated into virtual prefix tokens according to the semantic relevancy with the query measured by aggregration network, and subsequently prepended to the query in LLMs by prefix tuning technique (Li and Liang, 2021) for informative knowledge inference. Moreover, the meta-learning paradigm encourages faster adaption to new knowledge updates in RTE during meta-testing phase.

4.1 Meta-Training Phase

In meta-training phase, the key idea is to better initialize the amortization network and the aggregation network in an end-to-end training manner, consequently encouraging faster editing adaptation in meta-testing phase.

Given a training edit set $\mathcal{D}_{edit}^{tr}$, for each input-output rule-level knowledge pair $(\mathcal{R}_i^x, \mathcal{R}_i^y) \in \mathcal{D}_{edit}^{tr}$, we concatenate it and modularly encode it into a compact representation ϕ_i by a learnable T5-based hyper-amortization network $\mathcal{H}$ with parameter ξ_{amort} (Raffel et al., 2020):

$$\phi_i = \mathcal{H}(\xi_{amort}; [\mathcal{R}_i^x; \mathcal{R}_i^y]) \quad (4)$$

such that we obtain a compact rule-level knowledge representation with the shape of $[L, 2, 2, P, H]$, where L represents the number of layers, the first 2 corresponds to the dimensions of encoder and decoder, the latter 2 corresponds the key and value prefixes, P denotes the number of virtual prefix tokens, and H is the hidden size.

In order to generalize edited rule-level knowledge to the probing query x_q (which belongs to $\mathcal{R}^x$ during training), unlike existing memory-based editing methods (e.g., IKE (Zheng et al., 2023) and GRACE (Hartvigsen et al., 2023)) which require massive prompts in sequential edit or directly replace the layer's hidden states, we consider aggregating the relevant rule-level knowledge representations within query scope as soft prefix ϕ_r^*, which encompasses prefixed model-internal key-value pair for each layer in LLMs. Thus, we utilize cross-attention block (Kim et al., 2019) as a learnable knowledge aggregation network $\mathcal{G}$ to measure the semantic relevancy between the encoded query and the compressed rule-level knowledge set $\{\phi_i\}_{i=1}^n$, and we subsequently obtain the aggregated soft prefix as:

$$\phi_r^* = \mathcal{G}(\mathcal{H}(\xi_{input}; x_q); \{\phi_i\}_{i=1}^n) \quad (5)$$

where the query is encoded by the T5-based encoder with parameter ξ_{input} and same architecture as the above amortization network, and n denotes

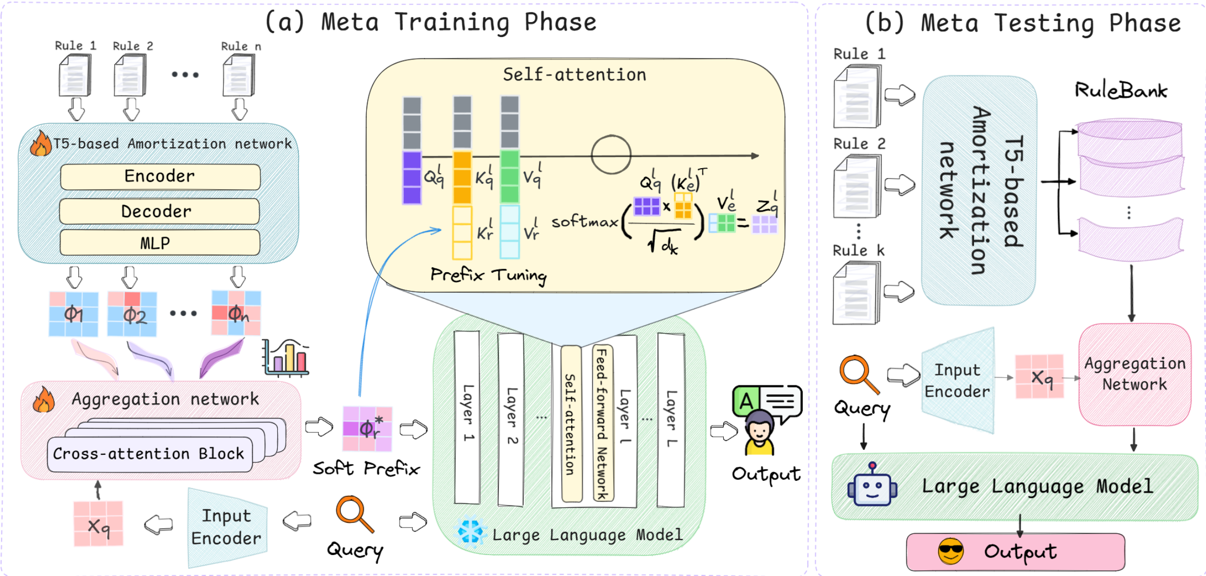


Figure 3: Overview of RTE. In meta-training phase, the T5-based amortization network modularly compresses the updated rule-level knowledge into semantic-centralized representations. For each query, the aggregation network aggregates the representations of relevant knowledge into soft prefix, which encompasses model-internal key-value representations for all layers in LLM. Subsequently, leveraging the prefix tuning technique, each learned key-value pair derived from soft prefix is prepended to the original key-value pairs layer-wise during inference, thus facilitating the propagation of edited rule-level knowledge in *RuleBank* to update rule-derived instances in meta-testing phase.

the number of edited knowledge. Other than specifically choosing a knowledge modulation, the aggregation network expands the utilization of the knowledge set and avoids the wrong choice of relevant knowledge.

LLMs are built on the Transformer architecture, which mainly consists of a self-attention module and a feed-forward module. Assuming the frozen LLMs (F) consist of L layers, to propagate aggregated relevant knowledge to the query, in each attention module Attn^l of layer l, we prepend the learned model-internal key and value representation K_r^l and V_r^l derived from soft prefix ϕ_r^* to the original key and value representations K_q^l and V_q^l of query calculated from the former layer $l-1$. Throughout the above simple yet effective deep prefix tuning process utilizing P-Tuning v2 (Liu et al., 2022), we have informative knowledge during the inference:

$$
\begin{gathered}
K_e^l = [K_r^l; K_q^l], \quad V_e^l = [V_r^l; V_q^l], \\
\text{Attn}^l(Q_q^l, K_e^l, V_e^l) = \text{softmax}(Q_q^l (K_e^l)^T) V_e^l, \\
Z^l = \text{LN}(\text{Attn}^l(Q_q^l, K_e^l, V_e^l) + X^{l-1}), \\
h^l = \text{LN}(\text{FFN}(Z^l) + Z^l)
\end{gathered} \tag{6}
$$

where X^{l-1} denotes the output of former layer $l-1$, LN represents the layer normalization operation, FFN represents the feed-forward network, and h^l denotes the output of layer l with query matrix Q_q^l.

To efficiently optimize the T5-based amortization network and aggregation network over the frozen LLMs, we train the model $\mathcal{F}^*$ in an end-to-end manner with the objective of:

$$
\begin{aligned}
\mathcal{L}_{edit} = & \mathcal{L}_r(\mathcal{F}^*(\mathcal{R}_i^x, \mathcal{R}_i^y) \\
& + c_g * \mathcal{L}_g(\mathcal{F}^*(\mathcal{R}_i^{x'}, \mathcal{R}_i^{y'})) \\
& + c_p * \mathcal{L}_p(\mathcal{F}^*(\mathcal{I}_i^x, \mathcal{I}_i^y))
\end{aligned} \tag{7}
$$

where $\mathcal{L}_r$, $\mathcal{L}_g$ and $\mathcal{L}_p$ are negative log-likelihood functions used to compute the loss, and both c_g and c_p are hyperparameters that govern the loss weight.

4.2 Meta-Testing Phase

Associating with the meta-learned hyper-model initialized in the meta-training phase, we manage to compress the rule-level knowledge of testing edit set $\mathcal{D}_{edit}^{test}$ into a set of modularized representations, which is called the *RuleBank*. For each query, we aggregate and propagate the relevant rule-level knowledge from the *RuleBank*, thereby facilitating generalizing the rule-level knowledge to update the rule-derived instances during the inference phase of LLMs.

5 Experiments

In this section, we provide construction details of our benchmark **RuleEdit**. Moreover, we conduct extensive experiments to explore the potential of LLMs in mitigating over-editing risk through rule-level editing and comprehensively evaluate the effectiveness of RTE.

5.1 Datasets

For comprehensive evaluations of rule-level editing performance in real-world scenarios, we construct **RuleEdit** benchmark, which is composed of both specific instances and the corresponding general rule-level knowledge covering three domains, including legal, medical, and historical domains.

We separately introduce the dataset generation processes for three domains: (1) For legal domain $RULE_{legal}$, we collect a set of legal judgments with 16,000 laws from DISC-Law-SFT (Yue et al., 2023) dataset. Sequentially, we prompt the LLMs (e.g., GPT-4o-mini (OpenAI, 2024)) to generate 3 statutory rules for each law, accompanied by corresponding rephrased rules and 10 legal instances. (2) For medical domain $RULE_{medical}$, we collect 480 medicine classes categorized by NLM [1]. For each, we obtain 10 associated medicinal substances by LLM, based on the hierarchical relationship of pharmacological effect, therapeutic usage, action mechanism, and chemical structure. (3) For historical domain $RULE_{historical}$, we collect 3441 historical events from ATOKE dataset (Yin et al., 2024) and construct corresponding historical instances within the timeline. More detailed examples and the construction process are provided in Appendix A.

Dataset	legal	medical	historical	mix
rule-level	16,482	3,186	3,441	9,450
instance-level	164,672	17,539	46,018	90,675
instance:rule	10.0:1	5.5:1	13.4:1	9.6:1

Table 1: Statistics of **RuleEdit** across legal, medical, historical and mixed domains.

Moreover, Table 1 demonstrates the statistics of collected rule-level knowledge and relevant rule-derived instances for each domain after quality control. To encourage balanced and comprehensive evaluations of rule-level editing across three domains, we further randomly sample 3,150 input-output rule-level knowledge pairs and corresponding accompanied instances for each domain. Subsequently, the samples are mixed and shuffled together to obtain the composed dataset $RULE_{mix}$. We compute the ratio of instances to rules in each domain. It is noticed that the mere difference among ratios is due to the motivation of ensuring generation quality while cascading unqualified data through quality control. In quality control, we modify or cascade unqualified cases according to the following guidelines (Details in Appendix A.2.1): (1) Clarity and completeness of knowledge. (2) Logical relevance between rules and instances. (3) Distinguishability among instances. (4) Factual reliability of the rules. (5) Inner-annotator agreement and expert review.

[1]https://www.ncbi.nlm.nih.gov/mesh/68008511

5.2 Experimental Settings

In our experiments, we compare against four representative distinct baselines, including (1) Parameter-efficient tuning method: LoRA (Hu et al., 2022); (2) Locate-and-edit method: ROME (Meng et al., 2022); (3) Meta-learning method: MEND (Mitchell et al., 2022a); (4) Memory-based method: GRACE (Hartvigsen et al., 2023). Besides, we utilize prevalent open-source LLMs LLaMA-2-7B (Touvron et al., 2023) and GPT2-XL (1.5B) (Radford et al., 2019) as base models and conduct experiments on our constructed **RuleEdit** covering legal, medical, historical, and mixed domains for comprehensive evaluation. For **RuleEdit** we use the same train/test split (9:1) as Mitchell et al. (2022a). More details are provided in Appendix B.

5.3 Experimental Results

Rule-level editing is challenging to existing editing methods. The experimental result of rule-level editing shown in Table 2 indicates that existing editing methods struggle to balance the performance of reliability, generalization ability, and portability. Specifically, it can be observed that ROME suffers from overall performance collapse for three aspects in sequential editing, since multiple biased adjustments for parameters of predefined layers significantly deteriorate the overall knowledge in large-scale edits. Although GRACE exhibits prominent reliability in editing rule-level knowledge, it has trouble generalizing edited rule-level knowledge to relevant rule-derived instances (e.g., poor portability score of 1.9% in the final editing step on $RULE_{mix}$). It is speculated that the update strategy of codebook in GRACE is insufficient to support precise measurement of semantic similarity, thereby limiting the retrieval of relevant knowledge. With competitive generalization ability and portability to the other baselines, MEND

Edit Step	Method	$RULE_{mix}$				$RULE_{historical}$				$RULE_{medical}$				$RULE_{legal}$			
		Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.
LLaMA-2-7B																	
3	ROME	85.9	**93.3**	**95.3**	**91.5**	**93.3**	**100.0**	**72.6**	**88.6**	85.0	**60.0**	**54.8**	**66.6**	12.9	15.2	5.9	11.3
	MEND	29.2	18.0	24.4	23.9	11.1	13.3	11.1	11.9	9.5	0.0	5.3	4.9	22.2	14.7	20.6	19.2
	GRACE	**93.3**	3.9	3.9	33.7	93.3	11.1	0.0	34.8	**93.3**	7.4	8.3	36.3	**100.0**	31.7	6.5	46.1
	LoRA	50.9	37.5	8.1	32.2	42.9	16.7	27.8	29.1	33.3	13.3	45.5	30.7	37.3	37.3	37.4	37.3
	Ours	83.0	54.6	68.7	68.8	50.0	50.0	36.3	45.4	77.8	45.7	50.8	58.1	47.7	**54.2**	**66.7**	**56.2**
10	ROME	37.7	27.7	45.4	36.9	46.2	**44.9**	35.2	42.1	0.0	0.0	0.0	0.0	12.9	12.1	12.8	12.6
	MEND	15.1	14.1	17.0	15.4	6.7	12.3	0.0	6.3	2.9	4.4	4.4	3.9	14.8	15.1	18.5	16.1
	GRACE	**88.2**	3.2	0.6	30.7	**88.6**	3.3	0.0	30.6	**98.0**	2.2	3.8	34.7	**98.6**	9.5	3.6	37.2
	LoRA	30.0	8.6	11.0	16.5	40.0	19.7	9.2	23.0	10.0	0.0	0.0	3.3	23.2	26.4	20.7	23.4
	Ours	53.4	**38.3**	**57.7**	**49.8**	59.0	44.5	**52.8**	**52.1**	51.1	**27.0**	**34.8**	**37.7**	47.0	**43.4**	**46.0**	**45.5**
100	ROME	6.1	2.7	5.8	4.9	0.5	0.0	0.1	0.2	2.8	2.4	2.5	2.6	18.5	18.3	18.1	18.3
	MEND	11.7	11.5	12.5	11.9	5.3	9.0	2.8	5.7	7.0	8.8	8.7	8.2	18.2	17.2	17.5	17.6
	GRACE	**94.0**	4.4	2.2	33.5	**90.3**	6.8	0.1	32.4	**81.0**	6.4	2.8	**30.1**	**91.8**	2.5	3.9	32.7
	LoRA	1.5	5.5	8.3	5.1	2.7	4.3	1.3	2.8	6.6	3.6	3.3	4.5	36.0	36.7	35.8	36.2
	Ours	44.7	**40.7**	**44.3**	**43.2**	47.7	**39.2**	**42.6**	**43.2**	36.2	**22.5**	**20.8**	26.5	53.2	**53.5**	**48.6**	**51.8**
final	ROME	0.5	0.5	0.1	0.4	0.0	0.0	0.0	0.0	1.4	1.3	1.7	1.5	10.7	10.8	11.5	11.0
	MEND	10.1	9.2	10.5	10.0	5.6	6.7	3.2	5.2	8.1	8.9	8.4	8.5	17.8	17.4	16.7	17.3
	GRACE	**85.8**	4.3	1.9	30.6	**88.7**	5.4	0.2	31.5	**74.1**	5.8	2.6	**27.5**	**93.9**	2.3	3.2	33.1
	LoRA	10.2	10.4	11.2	10.6	7.0	0.7	4.6	4.1	1.0	2.0	2.8	1.9	0.2	0.2	0.2	0.2
	Ours	38.8	**38.0**	**39.3**	**38.7**	53.1	**40.8**	**50.5**	**48.2**	31.6	**22.9**	**22.4**	25.6	53.9	**53.3**	**47.7**	**51.6**
	Ours(w/o SP)	5.11	4.75	3.95	4.60	1.96	1.88	2.07	1.97	7.93	6.65	5.26	6.61	5.07	4.86	4.88	4.94

Table 2: Main Results of Rule-Level Editing on LLaMA-2-7B with Multiple Edit Steps Measured by F1 Metric. We evaluate all the methods in three aspects under **RuleEdit**, which consists of three domain-specific sets and a mixed domain set. $Avg.$ indicates the average score of three aspects. SP indicates soft prefix. The final edit step indicates that all the rule-level knowledge of corresponding set are edited. **Best** and suboptimal results of each edit step are marked in **bold** and underline respectively.

and LoRA are prone to overfitting on training data and struggle with the adaptation of new knowledge, thereby resulting in poor reliability of edits (e.g., reliability scores of 10.1% in MEND and 10.2% in LoRA in the final editing step on $RULE_{mix}$). Additional experimental results for ACC and EM metrics and analysis are provided in Appendix B.4.

Effective knowledge updates through our RTE method. As shown in Table 2, our RTE method exhibits remarkable average performance within most domains in sequential rule-level editing, revealing the effective updates of both rule-level knowledge and corresponding rule-derived instance knowledge in RTE through rule-level editing. Although GRACE achieves higher scores in reliability, it makes huge sacrifices in generalization ability and portability. Instead, RTE leads the best performances in both aspects, achieving the enhancement of 27.6% in generalization score and 28.1% in portability score over the best baseline for LLaMA-2-7B in final editing step on $RULE_{mix}$. Moreover, the experiment highlights the adaptability of RTE across multiple domains, which necessitates capabilities in numerical reasoning, hierarchical knowledge inheritance, and semantic reasoning, as analyzed in Appendix A. This confirms that RTE achieves reliable rule-level editing through efficient knowledge amortization and robust knowledge generalization in inference.

Forward passing aggregated soft prefixes facilitates knowledge generalization. As shown in Table 2, the comparative experiment reveals significant performance gaps (e.g., a discrepancy of 34.11% in the average score on $RULE_{mix}$ in final step) in LLMs depending on whether relevant aggregated soft prefix knowledge is injected (Ours and Ours(w/o SP)). In comparison to GRACE, which directly replaces the activated state with the retrieved value and thus limits the generalization ability, our RTE demonstrates robust portability in inference, as indicated by the comparative experiment in Appendix B.3. Leveraging the prefix tuning technique, the prepended informative key-value representations derived from aggregated soft prefixes are forward passed to each attention layer in LLMs, thus facilitating thorough inference over edited rule-level knowledge to update corresponding rule-derived instances.

***RuleBank* serves as a safeguard against knowledge degradation.** In Figure 4, as edit steps increase, it is worth noting that methods involved in multiple adjustments to parameters (including

Method	$RULE_{mix}$				$RULE_{historical}$				$RULE_{medical}$				$RULE_{legal}$			
	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.
GPT2-XL (1.5B)																
ROME	0.63	0.68	0.63	0.65	13.72	<u>16.90</u>	<u>18.70</u>	16.44	7.27	6.71	<u>7.18</u>	7.06	15.59	14.92	<u>14.97</u>	15.16
MEND	17.56	<u>12.76</u>	4.66	11.66	3.41	4.96	0.44	2.94	12.82	<u>8.71</u>	2.70	8.07	30.96	<u>24.48</u>	5.42	20.29
GRACE	**90.17**	3.00	0.00	<u>31.06</u>	**98.06**	1.36	0.00	<u>33.14</u>	**71.21**	3.26	0.01	**24.83**	**100.00**	6.88	0.00	<u>35.63</u>
LoRA	8.96	4.42	<u>6.79</u>	6.72	20.41	5.85	4.17	10.14	4.15	4.42	3.98	4.18	5.83	5.30	6.27	5.80
Ours	<u>43.40</u>	**39.86**	**34.91**	**39.39**	<u>54.88</u>	**49.13**	**50.71**	**51.57**	<u>29.25</u>	**20.39**	**20.33**	<u>23.32</u>	<u>57.46</u>	**55.54**	**50.07**	**54.36**

Table 3: Comparative Results of Rule-Level Editing on GPT2-XL in Final Edit Steps Measured by F1 Metric. We evaluate all the methods under **RuleEdit**, which consists of three domain-specific sets and a mixed domain set. **Best** and <u>suboptimal</u> results are marked in **bold** and <u>underline</u> respectively.

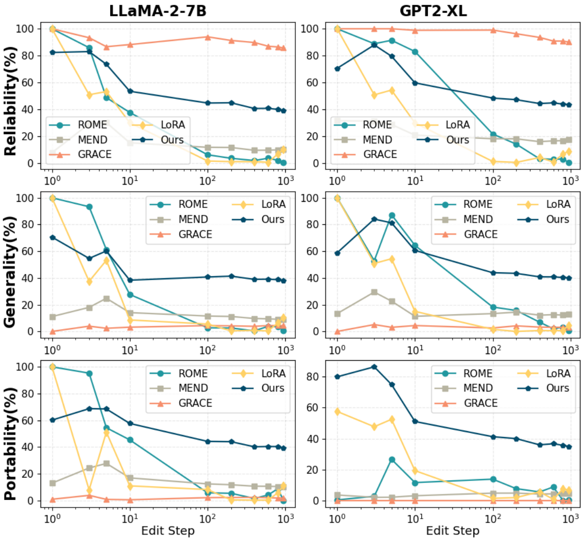


Figure 4: Comparisons of rule-level editing results with multiple editing steps over different methods, which are evaluated on $RULE_{mix}$ with F1 metric across three dimensions using LLaMA-2-7B and GPT2-XL.

MEND, LoRA, and ROME) suffer from catastrophic performance collapse in all dimensions. GRACE retains stable reliability performance owing to the memory codebook, but fails in generalization due to the limitation of knowledge retrieval ability. Contrarily, RTE achieves stable and excellent performance, owing to the preservation and propagation of rule-level knowledge from *RuleBank*. Furthermore, RTE achieves a leading portability score in GPT2-XL, indicating the superior generalization ability in rule-level editing.

Robust adaptability and pluggability in various backbones. In addition, Table 3 presents the experimental rule-level editing results conducted on GPT2-XL (1.5B). The prominent results reveal the robust adaptability of RTE to various LLMs backbones with distinct scales and indicate the promising potential of rule-level editing. Since the base LLMs is frozen and the edited rule-level knowledge is integrated flexibly through prefix tuning, our framework exhibits pluggability among LLMs. Moreover, comprehensive comparisons among editing methods in both LLaMA-2-7B and GPT2-XL are demonstrated in Appendix B.4.

5.3.1 Case Study

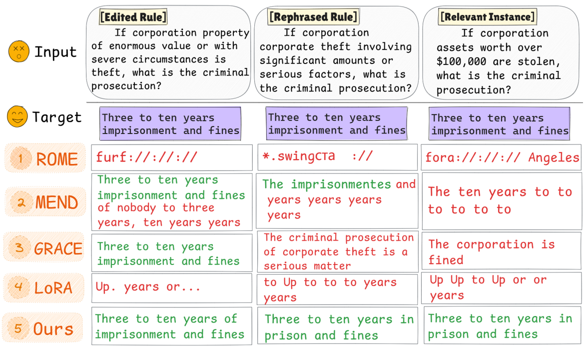


Figure 5: Examples of knowledge editing results for different methods. Evaluated on the final edit step of $RULE_{mix}$ using LLaMA-2-7B.

As illustrated in Figure 5, we perform a comparative case study over existing methods. It can be observed from the results that both ROME and LoRA produce unreliable and hasty generations, featuring typical over-editing results of repeated or meaningless tokens. It is indicated that an overfitting phenomenon occurs due to multiple biased modifications to the parameters. Mend produces incorrect answers due to erroneous generalization, revealing the limited generalization ability. As analyzed in the above experiments, the generalization ability of GRACE is constrained by retrieval performance, resulting in inaccurate generation with missing information. In contrast, our method delivers satisfactory results, demonstrating the promising potential to effectively generalize rule-level knowledge to update relevant rule-derived instances, thereby mitigating the over-editing risk.

6 Conclusion

In this work, we explore the rule-level editing problem to achieve effective knowledge updates and mitigate over-editing risk in LLMs, and construct a new benchmark **RuleEdit** across three domains for comprehensive evaluations. Additionally, we further propose RTE method to facilitate effective modifications and propagations of rule-level knowledge. Our experimental results demonstrate excessive rule-level editing performance of RTE with prevalent portability for effective knowledge generalization.

7 Limitations

Similar to most memory-based methods, our RTE method faces the challenge that *RuleBank* grows in scale as rule-level knowledge accumulates, leading to increased memory consumption. Future work may consider neighborhood knowledge fusion to reduce memory scale while maintaining editing performance, especially since RTE exhibits competitive performance in GPT2-XL compared with other baselines in LLAMA-2-7B. Additionally, a possible improvement involves designing a gate mechanism to selectively determine whether to integrate knowledge from *RuleBank*, thereby enhancing flexibility in knowledge integration.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (62476097), Guangdong Provincial Fund for Basic and Applied Basic Research-Regional Joint Fund Project (Key Project)(2023B1515120078), Guangdong Provincial Natural Science Foundation for Outstanding YouthTeam Project (2024B1515040010).

References

Afra Feyza Akyürek, Eric Pan, Garry Kuwanto, and Derry Wijaya. 2023. Dune: Dataset for unified editing. In *Proceedings of the EMNLP 2023*, pages 1847–1861.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the ACL 2022*, pages 8493–8502.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the EMNLP 2021*, pages 6491–6506.

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. Everything is editable: Extend knowledge editing to unstructured data in large language models. In *Proceedings of the ICLR 2025*.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Proceedings of the EMNLP Findings 2022*, pages 5937–5947.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the EMNLP 2021*, pages 5484–5495.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *Proceedings of the NeurIPS 2023*, volume 36, pages 47934–47959.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Proceedings of the NeurIPS 2023*, volume 36, pages 17643–17668.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of the ICLR 2022*.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, S. M. Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. 2019. Attentive neural processes. *arXiv preprint arXiv:1901.05761*.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the ACL 2021*, pages 4582–4597.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. Pmet: Precise model editing in a transformer. *Proceedings of the AAAI 2024*, 38(17):18564–18572.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the ACL 2022 (Volume 2: Short Papers)*, pages 61–68.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the EMNLP 2022*, pages 2833–2861.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Proceedings of the NeurIPS 2022*, volume 35, pages 17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *Proceedings of the ICLR 2023*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In *Proceedings of the ICLR 2022*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the ICML 2022*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. 2023. Hypertuning: toward adapting large language models without back-propagation. In *Proceedings of the ICML 2023*, pages 27854–27875.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry V. Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. In *Proceedings of the ICLR 2020*.

Jihoon Tack, Jaehyung Kim, Eric Mitchell, Jinwoo Shin, Yee Whye Teh, and Jonathan Richard Schwarz. 2024. Online adaptation of language models with a memory of amortized contexts. In *Proceedings of the NeurIPS 2024*.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang, Linyi Yang, Jindong Wang, and Huajun Chen. 2024a. Detoxifying large language models via knowledge editing. In *Proceedings of the ACL 2024*, pages 3093–3118.

Xiaohan Wang, Shengyu Mao, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. 2024b. Editing conceptual knowledge for large language models. In *Proceedings of the EMNLP Findings 2024*, pages 706–724.

Zecheng Wang, Xinye Li, Zhanyue Qin, Chunshan Li, Zhiying Tu, Dianhui Chu, and Dianbo Sui. 2024c. Can we debias multimodal large language models via model editing? In *Proceedings of the MM 2024*, pages 3219–3228.

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 EMNLP 2023*, pages 10222–10240.

Xunjian Yin, Jin Jiang, Liming Yang, and Xiaojun Wan. 2024. History matters: Temporal knowledge editing in large language model. In *Proceedings of the AAAI 2024*, volume 38, pages 19413–19421.

Li Yuan, Yi Cai, Jin Wang, and Qing Li. 2023. Joint multimodal entity-relation extraction based on edge-enhanced graph alignment network and word-pair relation tagging. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11051–11059.

Li Yuan, Yi Cai, Jingyu Xu, Qing Li, and Tao Wang. 2025. A fine-grained network for joint multimodal entity-relation extraction. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):1–14.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, et al. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325*.

Mengqi Zhang, Xiaotian Ye, Q. Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024a. Knowledge graph enhanced large language model editing. In *Proceedings of the EMNLP 2024*.

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. 2024b. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the EMNLP 2023*, pages 4862–4876.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the EMNLP 2023*.

A Dataset

A.1 Data Samples

Table 4 presents data samples from our constructed benchmark **RuleEdit**. Specifically, **RuleEdit** is collected from three domains, including the legal domain, the medical domain, and the historical domain. Each dataset unit consists of rule-level knowledge for editing, rephrased rule-level knowledge for generalization evaluation, and relevant instances derived from the rule for portability evaluation.

As observed from the samples, dataset in the legal domain aims to enable proper judgment for specific cases after editing the corresponding statute, which requires robust semantic reasoning ability over edited legal rules for LLMs. Dataset in the medical domain aims to enable hierarchical knowledge inheritance from the edited universal medical knowledge. Moreover, dataset in the historical domain involves knowledge inference with specific time constraints, which requires solid numerical reasoning ability for LLMs.

A.2 Dataset Construction

Figure 6 outlines the detailed construction process of **RuleEdit**. Firstly, we collect knowledge from different corpuses across three domains. Based on the collected knowledge, we manage to extract and generate rule-level knowledge for editing, and rephrase the expression for generalization evaluation. Subsequently, we generate relevant instances that can be derived from the edited rule-level knowledge. As shown in Figure 7 and Figure 8, we prompt GPT-4o-mini to assist in the generation of rule-level knowledge and relevant instances in both legal and medical domains. Under quality control and random sampling, the dataset **RuleEdit** is obtained, which consists of separate data in three domains and a mixture set.

A.2.1 Dataset Quality Control Guidelines

To ensure high-quality annotations, we employ three well-educated annotators during the construction of the **RuleBank** and adhere to the following quality control guidelines: (1) *Clarity and completeness of knowledge.* Each input-output knowledge pair of rule-level knowledge and relevant rule-derived instances must be clearly described, leaving no ambiguity in interpretation or application. (2) *Logical relevance between rules and instances.* Rule-derived instances should be logically inferable from the corresponding rule-level knowledge without additional information. (3) *Distinguishability among instances.* Instances should be distinct and non-redundant, ensuring the diversity and coverage within the scope of the corresponding rule. (4) *Factual reliability of the rules.* Rules must be accurately derived from knowledge sources and free from contradictions. (5) *Inner-annotator agreement and expert review.* Annotators independently assess the quality of each input-output knowledge pair by assigning a score within the range of zero to five. Discrepancies are resolved through collaborative discussions, with final decisions made by a senior expert to refine the dataset.

B Experiments Details

B.1 Baselines

Here we provide a detailed introduction and implementation information for all baselines in the experiments.

ROME ROME (Meng et al., 2022) uses causal tracing to investigate the decisive feedforward MLPS associated with knowledge, and alters corresponding parameters by rank-one model with least squares approximation. For the experiments, the learning rate is set to 5e-1, the kl factor is set to 0.0625. For LLaMA-2-7B, ROME is executed in layer 5 with 25 optimization steps. For GPT2-XL, ROME is executed in layer 17 with 20 optimization steps.

Domain	Rule-level knowledge	Rephrased knowledge	Relevant knowledge
Legal	If an individual intentionally destroys property with significant value, what is the criminal prosecution? *Imprisonment of up to three years, detention, or a fine.*	If an individual willfully damages property that is of considerable worth, what is the criminal prosecution? *Imprisonment of up to three years, detention, or a fine.*	If Tom destroys property valued at $10,000 or more, what is the criminal prosecution? *Imprisonment of up to three years, detention, or a fine.*
Medical	If a medicine is a type of anticoagulant, what is the pharmacological effect of it? *Blood clot prevention.*	What is the pharmacological effect of a medicine that belongs to the class of anticoagulants? *Blood clot prevention.*	Warfarin is a type of anticoagulant. What is the pharmacological effect of warfarin? *Blood clot prevention.*
Historical	Which club does Giorgio Morini affiliate with from 1976 to 1981? *A.C. Milan.*	From 1976 to 1981, Giorgio Morini played for? *A.C. Milan.*	Which club does Giorgio Morini affiliate with from 1979 to 1980? *A.C. Milan.*

Table 4: Examples of **RuleEdit**.

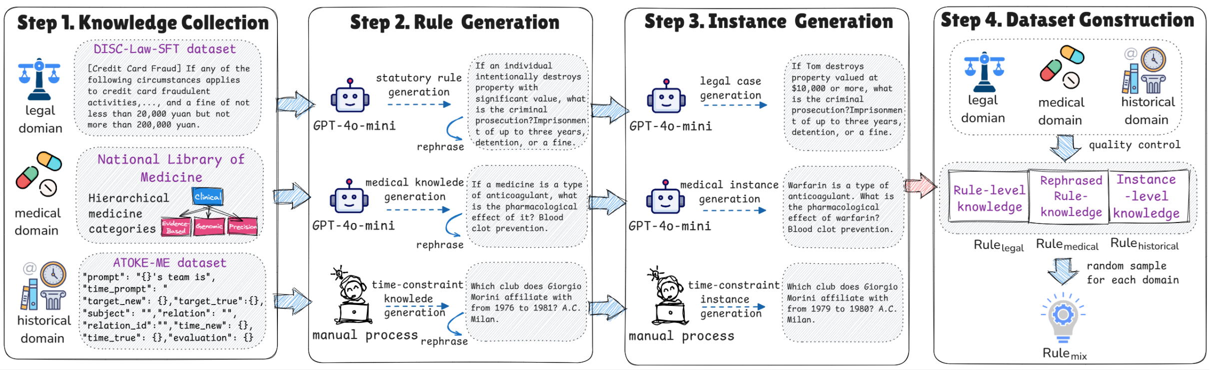


Figure 6: Construction process of **RuleEdit**.

MEND MEND (Mitchell et al., 2022a) designs a hypernetwork to decompose standard fine-tuning gradient of knowledge editing into corresponding rank-1 outer product form, and further adopts a meta-learning objective comprising the autoregressive loss and KL divergence loss. For the experiments, MEND edits in the last 3 transformer blocks, and the learning rate is set to 1e-6, while the scale of autoregressive loss and KL divergence loss are set to 0.1 and 1, respectively.

GRACE GRACE (Hartvigsen et al., 2023) maintains a discrete key-value codebook for a chosen layer to cache embedding for updated knowledge, and selectively replaces the activation of hidden state output with the retrieved value from the codebook during inference. For the experiments, the learning rate is set to 1. and the codebook is executed in layer 27 for LLaMA-2-7B and layer 35 for GPT2-XL.

LoRA LoRA (Hu et al., 2022) performs direct optimization for rank decomposition matrices of each layers, while keeping the pre-trained weight frozen. For the experiments, the learning rate is set to 5e-3, the rank is set to 8, and the dropout rate is set to 0.1.

B.2 Implementation Details

We conduct all the experiments on two NVIDIA A800 GPUs, and follow the default hyperparameter settings of the baselines. In our method, we utilize Adam optimizer (Kingma, 2014) with a learning rate of 1e-5 and train for 20 epochs for all datasets. We set the virtual output token number of T5-based amortization network to 24 and the training batch size to 16. Besides, following Tack et al. (2024), we utilize T5-large model and an individual two-layered MLP for each output virtual token for the amortization network, and T5-based model (Raffel et al., 2020) for the input encoder. For the aggrega-

Prompts

Legal Domain

```
f"# statute:\n{statute} \
Please follow these steps according to the given statute to ensure that the generated dataset
meets the requirements:\n \
## Step 1: please brainstorm and generate legal entities as many as you can, Format the
generated legal entities as follows:\n**Legal entities :[list of legal entity]\n \
## Step 2: Combining the generated legal entities, generate in English with more than 3
statutory rules with legal entity, condition, legal action and more than 10 legal instances for
each statutory rule. It should includes:\n- legal entity\n- condition\n- legal action type\n-
legal action result\n -more than 10 legal rule instances\n\nFormat the statutory rule dataset as
follows:\n\n**Statutory rule [index]:**\n- **legal entity:** [legal entity]\n- **Condition:**
[Condition]\n- **Rephrased Condition:** [Rephrased Condition]\n- **Legal action type:** [Legal
action type]\n- **legal action result:** [Legal action result]\n- **Rule Instance [Instance
index]:** \n-- **If-condition:** [If-condition]\n-- **Rephrased If-condition:** [Rephrased If-
condition]\n-- **Legal action type**: [Legal action type]\n-- **Legal action result**: [Legal
action result] \"
```

Figure 7: Sample prompt in the legal domain to assist generations of rule-level knowledge and relevant instances.

Prompts

Medical Domain

```
f"## \nmedicine class: [{medicine_class}]\n \
        Please follow these steps to ensure that the generated dataset meets the requirements:\n \
    ## Step 1: please brainstorm and generate mechenism of action of medicine classes as many as
you can, Format the generated chemical structures as follows:\n**{Classification criteria} :[list
of {classification criteria}]\n \
    ## Step 2: Combining the generated {classification criteria} of medicine classes, generate
medicine dataset with medicine class classified by {classification criteria}, and medicine
instance. It should includes:\n- the name of medicine class\n- {classification criteria} of
corresponding medicine class.\n- more than 10 medicine instances belongs to the medicine
class\n\nFormat the medical dataset as follows:\n\n**Medicine Class [index]:** [Name of Medicine
class]\n- **{classification criteria}:** [{classification criteria} of Medicine class]\n-
**Medicine instances:**\n-- **Medicine [Instance index]:** \n--- **Medicine Name:** [Name of
medicine instance]\n--- **Therapeutic effect**: [Therapeutic effect]\n--- **Active
compound/ingredient**: [Active compound or ingredient] \"
```

Figure 8: Sample prompt in the medical domain to assist generations of rule-level knowledge and relevant instances.

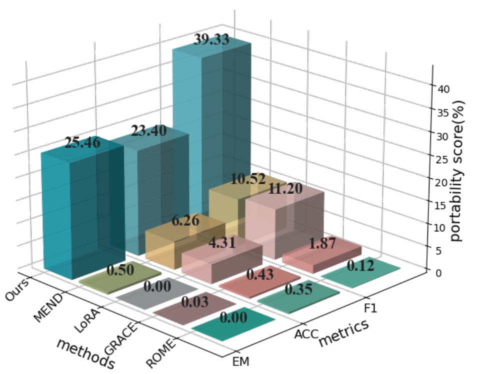


Figure 9: Portability comparison of rule-level editing with EM, ACC, and F1 metrics under different methods, which are evaluated on $RULE_{mix}$ using LLaMA-2-7B.

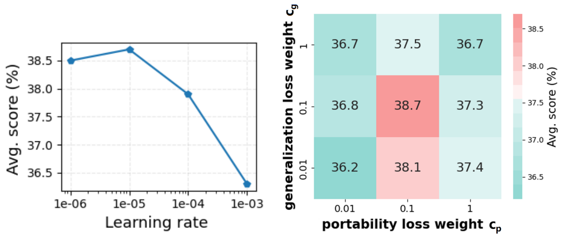
tion network, we utilize four cross-attention blocks, which each consist of a cross-attention and a feed-forward network. According to the comparative results of different settings shown in Figure 10, we configure both the generalization loss weight c_g and the portability loss weight c_p to 0.1.

B.3 Solid Portability for Knowledge Generalization

As shown in Figure 9, we conduct rule-level editing experiments evaluated on EM, ACC, and F1 metrics, aiming to sufficiently compare the portability of current knowledge editing methods. It can be observed that RTE surpasses the other baseline in all metrics, indicating efficient generalization of rule-level knowledge. Compared with redundant case-by-case edits brought by instance-level editing, rule-level editing effectively avoids massive editing scale by solid portability to achieve efficient updates on rule-derived instances.

B.4 Comprehensive Evaluation on RuleEdit

As shown in Table 5, 6, 7, 8, and 9, we conduct complementary experiments to comprehensively evaluate the performance of representative knowledge editing methods on both LLaMA-2-7B and GPT2-XL using ACC, EM, and F1 metrics. It can be observed from the results that RTE leads a favorable overall performance in both LLaMA-2-7B and GPT2-XL with ACC and F1 metrics, and exhibits superior generalization ability and portability compared with other baselines in EM metrics for both LLaMA-2-7B and GPT2-XL, which highlights the robustness and effectiveness of our method.



(a) Results on different setting of learning rate.

(b) Results on different setting of loss weight.

Figure 10: Comparisons of avg. score on F1 metric among different parameter settings over LLaMA-2-7B on $RULE_{mix}$.

Edit Step	Method	$RULE_{mix}$				$RULE_{historical}$				$RULE_{medical}$				$RULE_{legal}$			
		Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.
LLaMA-2-7B																	
3	ROME	52.78	61.67	72.40	62.28	41.67	41.67	20.71	34.68	93.33	100.00	72.59	88.64	3.33	1.23	0.25	1.60
	MEND	16.73	6.08	5.88	9.57	12.50	9.72	0.00	7.41	11.11	13.33	11.11	11.85	9.14	9.14	13.10	10.46
	GRACE	61.67	0.00	0.26	20.64	41.67	0.00	0.00	13.89	93.33	11.11	0.00	34.81	88.77	1.23	1.67	30.56
	LoRA	46.11	30.56	4.22	26.96	16.67	16.67	26.19	19.84	42.86	16.67	27.78	29.10	26.67	26.67	27.00	26.78
	Ours	40.66	14.42	15.65	23.58	56.19	50.00	42.61	49.60	50.00	50.00	36.25	45.42	33.10	35.67	50.66	39.81
10	ROME	12.38	5.71	16.75	11.62	2.00	2.00	3.18	2.39	1.25	0.00	0.74	0.66	3.17	7.30	5.90	5.46
	MEND	11.84	2.73	6.05	6.87	2.50	5.42	1.74	3.22	8.75	5.21	1.77	5.25	17.17	11.95	12.34	13.82
	GRACE	44.99	0.00	0.23	15.07	28.00	0.00	0.00	9.33	66.37	1.25	0.31	22.64	78.63	0.37	0.78	26.59
	LoRA	8.93	2.93	1.51	4.46	6.00	9.93	0.00	5.31	13.19	0.00	0.00	4.40	16.68	13.70	11.22	13.87
	Ours	43.06	16.40	21.81	27.09	63.94	59.00	43.21	55.38	65.87	43.83	30.99	46.89	33.07	38.22	42.69	38.00
100	ROME	2.63	1.69	3.78	2.70	1.85	1.05	1.87	1.59	0.98	0.50	1.09	0.85	5.35	4.32	5.52	5.06
	MEND	9.91	6.82	6.14	7.62	7.43	5.82	2.48	5.24	8.80	7.60	3.75	6.72	11.37	11.39	8.62	10.46
	GRACE	59.11	0.88	0.29	20.09	36.98	2.02	0.20	13.07	47.09	0.31	0.14	15.85	73.35	0.95	0.69	25.00
	LoRA	0.50	0.00	0.00	0.17	8.54	9.71	3.95	7.40	13.19	0.00	0.00	4.40	15.00	14.64	14.87	14.83
	Ours	45.20	30.39	24.65	33.41	55.01	47.67	39.88	47.52	65.87	43.83	30.99	46.89	38.37	39.90	39.26	39.18
final	ROME	0.34	0.42	0.35	0.37	4.68	2.05	6.89	4.54	0.19	0.03	0.20	0.14	4.36	4.56	4.68	4.54
	MEND	11.41	8.01	6.26	8.56	7.95	5.07	0.17	4.40	10.05	6.35	2.95	6.45	12.87	11.80	10.03	11.57
	GRACE	**51.29**	1.01	0.43	17.58	32.14	1.26	0.13	11.18	46.95	0.46	0.15	15.86	**73.85**	0.67	0.86	25.12
	LoRA	5.14	2.69	4.31	4.05	9.65	9.62	2.59	7.28	0.43	0.52	0.89	0.61	0.66	0.94	0.30	0.63
	Ours	45.13	**29.35**	**23.40**	**32.63**	**59.64**	**53.15**	**51.90**	**36.35**	**47.53**	**41.39**	**35.54**	**41.49**	42.27	**42.07**	**38.68**	**41.01**

Table 5: Comparative Results of Rule-Level Editing on LLaMA-2-7B with Multiple Edit Steps Measured by ACC Metric.

Edit Step	Method	$RULE_{mix}$				$RULE_{historical}$				$RULE_{medical}$				$RULE_{legal}$			
		Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.
LLaMA-2-7B																	
3	ROME	66.67	100.00	84.38	83.68	100.00	100.00	36.51	78.84	66.67	66.67	54.55	62.63	0.00	0.00	0.00	0.00
	MEND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	GRACE	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33
	LoRA	33.33	33.33	0.00	22.22	33.33	0.00	3.17	12.17	33.33	0.00	45.45	26.26	33.33	33.33	33.33	33.33
	Ours	33.33	0.00	37.50	23.61	33.33	33.33	21.43	29.37	66.67	0.00	45.45	37.37	0.00	0.00	0.00	0.00
10	ROME	30.00	20.00	42.11	30.70	40.00	40.00	27.55	35.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MEND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	GRACE	90.00	0.00	0.00	30.00	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33
	LoRA	30.00	0.00	0.00	10.00	50.00	0.00	0.00	16.67	10.00	0.00	0.00	3.33	10.00	10.00	2.00	7.33
	Ours	20.00	0.00	33.33	17.78	50.00	40.00	41.45	43.82	40.00	20.00	16.18	25.39	0.00	10.00	9.00	6.33
100	ROME	1.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MEND	1.00	0.00	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	GRACE	94.00	0.00	0.00	31.33	99.00	1.00	0.00	33.33	83.00	0.00	0.00	27.67	83.00	0.00	0.00	27.67
	LoRA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.33	1.00	1.00	0.60	0.87
	Ours	21.00	13.00	26.13	20.04	34.00	28.00	33.17	31.72	24.00	16.00	16.70	18.90	5.00	7.00	14.37	8.79
final	ROME	0.00	0.00	0.00	0.00	1.56	0.00	0.00	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MEND	0.11	0.00	0.50	0.20	0.00	0.58	0.00	0.19	0.31	0.00	0.11	0.14	0.00	0.00	0.00	0.00
	GRACE	**86.02**	0.32	0.03	**28.79**	**98.55**	0.29	0.00	32.95	**73.98**	0.00	0.31	**24.76**	**86.66**	0.00	0.00	**28.89**
	LoRA	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Ours	17.25	**14.81**	**25.46**	19.17	40.87	**27.25**	**41.75**	**36.62**	19.12	**14.73**	**18.05**	17.30	8.85	**8.25**	**15.20**	10.77

Table 6: Comparative Results of Rule-Level Editing on LLaMA-2-7B with Multiple Edit Steps Measured by EM Metric.

Edit Step	Method	$RULE_{mix}$				$RULE_{historical}$				$RULE_{medical}$				$RULE_{legal}$			
		Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.
GPT2-XL																	
3	ROME	73.64	57.92	2.69	44.75	75.56	22.22	0.56	32.78	55.30	55.30	53.18	54.60	3.33	3.33	1.97	2.88
	MEND	0.00	13.81	12.24	8.68	0.00	0.00	6.00	2.00	0.00	16.67	0.41	5.69	1.28	10.00	2.00	4.43
	GRACE	87.92	6.67	0.00	31.53	75.56	6.67	0.07	27.43	55.30	15.15	0.00	23.49	88.72	0.00	0.00	29.57
	LoRA	46.97	46.97	51.53	48.49	30.30	30.30	41.32	33.98	30.30	30.30	41.32	33.98	26.67	26.67	26.79	26.71
	Ours	33.33	33.33	37.50	34.72	33.33	33.33	21.43	29.37	25.51	17.17	0.00	14.23	20.00	20.00	34.90	24.97
10	ROME	48.10	18.38	11.56	26.01	55.00	14.67	19.63	29.77	35.89	28.78	28.70	31.12	34.88	31.54	9.96	25.46
	MEND	13.08	13.48	9.91	12.16	2.00	9.25	4.73	5.33	5.33	11.36	0.50	5.73	7.92	9.11	1.29	6.11
	GRACE	72.36	7.43	0.00	26.60	71.00	6.00	0.02	25.67	66.56	4.55	0.00	23.70	78.62	0.00	0.00	26.21
	LoRA	26.34	12.34	19.24	19.31	13.19	13.19	7.62	11.34	13.19	13.19	7.62	11.34	16.20	16.20	16.20	16.20
	Ours	32.86	38.75	30.87	34.16	41.67	31.67	21.27	31.54	51.12	20.15	0.77	24.02	34.33	33.34	29.98	32.55
100	ROME	13.85	13.04	13.77	13.55	32.22	22.56	21.71	25.50	2.00	5.61	4.07	3.89	8.09	7.89	8.03	8.00
	MEND	10.41	14.70	5.84	10.32	1.15	6.73	2.77	3.55	8.23	12.48	0.01	6.91	5.43	6.25	1.68	4.46
	GRACE	73.35	2.45	0.00	25.27	65.71	7.12	0.01	24.28	54.65	0.88	0.00	18.51	82.45	0.08	0.00	27.51
	LoRA	0.25	0.25	0.32	0.27	0.65	0.65	0.41	0.57	0.65	0.65	0.41	0.57	6.06	5.71	6.53	6.10
	Ours	30.41	28.50	22.55	27.16	40.52	39.52	34.73	38.25	34.86	21.29	4.54	20.23	31.79	30.73	27.78	30.10
final	ROME	1.46	1.22	0.48	1.05	16.10	16.53	21.02	17.88	3.41	2.79	2.77	2.99	5.53	5.57	4.69	5.27
	MEND	11.08	14.19	6.03	10.43	1.28	7.34	3.10	3.91	8.56	9.17	0.60	6.11	6.78	6.18	1.74	4.90
	GRACE	**65.07**	2.33	0.00	22.47	**66.85**	7.62	0.00	24.82	**46.58**	0.45	0.00	15.68	**79.54**	0.39	0.00	26.64
	LoRA	1.94	2.00	2.22	2.06	0.55	0.92	1.24	0.90	0.55	0.92	1.24	0.90	6.33	5.72	7.75	6.60
	Ours	34.17	**30.63**	**20.58**	**28.46**	46.51	**40.99**	**42.71**	**43.40**	29.70	**18.13**	**5.50**	**17.78**	35.21	**34.02**	**28.60**	**32.61**

Table 7: Comparative Results of Rule-Level Editing on GPT2-XL with Multiple Edit Steps Measured by ACC Metric.

Edit Step	Method	$RULE_{mix}$				$RULE_{historical}$				$RULE_{medical}$				$RULE_{legal}$			
		Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.	Rel.	Gen.	Port.	Avg.
GPT2-XL																	
3	ROME	66.67	33.33	57.22	52.41	100.00	33.33	19.44	50.92	66.67	66.67	66.67	66.67	0.00	0.00	0.00	0.00
	MEND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	GRACE	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33
	LoRA	33.33	33.33	31.25	32.64	33.33	33.33	45.45	37.37	33.33	33.33	45.45	37.37	33.33	33.33	33.33	33.33
	Ours	33.33	33.33	37.50	34.72	33.33	33.33	21.43	29.37	0.00	0.00	0.00	0.00	0.00	0.00	33.33	11.11
10	ROME	60.00	30.00	27.71	39.24	80.00	30.00	36.92	48.97	40.00	20.00	49.00	36.33	30.00	20.00	10.00	20.00
	MEND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	GRACE	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33	100.00	0.00	0.00	33.33
	LoRA	30.00	0.00	0.00	10.00	10.00	10.00	10.29	10.10	10.00	10.00	10.29	10.10	10.00	10.00	10.00	10.00
	Ours	30.00	30.00	37.72	32.57	40.00	30.00	20.39	30.13	30.00	10.00	2.94	14.31	10.00	10.00	20.00	13.33
100	ROME	7.00	6.00	2.30	5.10	36.00	28.00	21.55	28.52	2.00	6.00	4.27	4.09	1.00	1.00	0.30	0.77
	MEND	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.33	0.00	0.00	0.00	0.00
	GRACE	100.00	0.00	0.00	33.33	98.00	1.00	0.00	33.00	85.00	0.00	0.00	28.33	100.00	0.00	0.00	33.33
	LoRA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Ours	21.00	19.00	23.65	21.22	37.00	34.00	31.50	34.17	19.00	13.00	12.90	14.97	7.00	7.00	10.55	8.18
final	ROME	0.00	0.00	0.00	0.00	10.14	8.99	10.06	9.73	0.63	0.31	0.31	0.42	0.06	0.00	0.05	0.04
	MEND	0.32	0.11	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.63	0.00	0.21	0.06	0.00	0.00	0.02
	GRACE	**91.31**	0.00	0.00	**30.44**	**99.13**	0.29	0.00	33.14	**72.73**	0.00	0.00	**24.24**	**99.94**	0.00	0.00	**33.31**
	LoRA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Ours	20.74	**18.10**	**17.93**	18.92	42.90	**35.65**	**39.46**	**39.34**	16.61	**10.03**	**11.83**	12.83	11.22	**10.67**	**13.98**	11.96

Table 8: Comparative Results of Rule-Level Editing on GPT2-XL with Multiple Edit Steps Measured by EM Metric.

Edit Step	Method	$RULE_{mix}$ Rel.	Gen.	Port.	Avg.	$RULE_{historical}$ Rel.	Gen.	Port.	Avg.	$RULE_{medical}$ Rel.	Gen.	Port.	Avg.	$RULE_{legal}$ Rel.	Gen.	Port.	Avg.
GPT2-XL																	
3	ROME	88.89	52.94	65.64	69.16	100.00	82.22	52.71	78.31	93.33	93.33	68.86	85.18	11.94	13.46	6.01	10.47
	MEND	31.36	29.44	2.01	20.94	0.00	16.67	0.00	5.56	0.00	9.52	0.00	3.17	40.92	22.54	7.28	23.58
	GRACE	100.00	5.13	0.00	35.04	100.00	0.00	0.00	33.33	100.00	11.11	0.06	37.06	99.28	3.22	0.00	34.17
	LoRA	50.88	50.88	47.70	49.82	44.44	44.44	58.73	49.21	33.33	33.33	45.45	37.37	37.78	37.78	37.78	37.78
	Ours	87.97	84.05	86.45	86.16	50.00	50.00	41.96	47.32	22.86	22.86	26.48	24.06	50.17	50.17	58.57	52.97
10	ROME	83.11	64.42	43.47	63.67	80.00	37.78	56.38	58.05	50.55	34.86	56.89	47.43	50.59	53.47	18.38	40.81
	MEND	20.91	11.33	3.05	11.77	4.00	5.00	0.44	3.15	6.94	5.36	2.91	5.07	27.21	25.93	5.50	19.55
	GRACE	98.89	4.40	0.00	34.43	98.57	0.00	0.00	32.86	99.09	8.69	0.02	35.93	99.58	3.04	0.00	34.21
	LoRA	30.00	15.00	19.30	21.43	50.00	50.00	32.14	44.05	10.00	10.00	10.29	10.10	27.14	27.14	27.14	27.14
	Ours	59.80	60.63	51.09	57.17	49.00	39.00	32.83	40.28	45.75	27.94	16.70	30.13	44.78	49.65	50.20	48.21
100	ROME	21.75	18.24	17.77	19.26	46.60	40.14	33.38	40.04	12.92	14.14	10.96	12.67	21.42	19.66	22.44	21.17
	MEND	18.24	13.33	4.74	12.11	3.40	7.84	0.90	4.05	10.67	8.21	0.03	6.30	30.95	26.89	5.38	21.07
	GRACE	99.04	2.58	0.00	33.87	98.02	1.73	0.01	33.25	85.00	4.35	0.01	29.79	100.00	2.06	0.00	34.02
	LoRA	1.34	1.58	1.38	1.43	1.17	0.00	0.00	0.39	1.14	1.14	0.62	0.96	17.12	18.10	17.71	17.64
	Ours	48.48	43.93	41.25	44.55	49.90	46.47	41.25	45.87	36.54	25.97	20.06	27.52	57.63	53.34	51.63	54.20
final	ROME	0.63	0.68	0.63	0.65	13.72	16.90	18.70	16.44	7.27	6.71	7.18	7.06	15.59	14.92	14.97	15.16
	MEND	17.56	12.76	4.66	11.66	3.41	4.96	0.44	2.94	12.82	8.71	2.70	8.07	30.96	24.48	5.42	20.29
	GRACE	**90.17**	3.00	0.00	31.06	**98.06**	1.36	0.00	33.14	**71.21**	3.26	0.01	**24.83**	**100.00**	6.88	0.00	35.63
	LoRA	8.96	4.42	6.79	6.72	20.41	5.85	4.17	10.14	4.15	4.42	3.98	4.18	5.83	5.30	6.27	5.80
	Ours	43.40	**39.86**	**34.91**	**39.39**	54.88	**49.13**	**50.71**	**51.57**	29.25	**20.39**	**20.33**	23.32	57.46	**55.54**	**50.07**	**54.36**

Table 9: Comparative Results of Rule-Level Editing on GPT2-XL with Multiple Edit Steps Measured by F1 Metric.