Outdated Issue Aware Decoding for Reasoning Questions on Edited Knowledge

Zengkui Sun[1]*, **Yijin Liu**[2], **Jiaan Wang**[3], **Fandong Meng**[2], **Jinan Xu**[1], **Yufeng Chen**[1]† and **Jie Zhou**[2]
[1]Beijing Jiaotong University, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
[3]Soochow University
{zengksun,jaxu,chenyf}@bjtu.edu.cn jawang.nlp@gmail.com
{yijinliu,fandongmeng,withtomzhou}@tencent.com

Abstract

Recently, Knowledge Editing has received increasing attention, since it could update the specific knowledge from outdated ones in pre-trained models without re-training. However, as pointed out by recent studies, existing related methods tend to merely memorize the superficial word composition of the edited knowledge, rather than truly learning and absorbing it. Consequently, on the reasoning questions, we discover that existing methods struggle to utilize the edited knowledge to reason the new answer, and tend to retain outdated responses, which are generated by the original models utilizing original knowledge. Nevertheless, the outdated responses are unexpected for the correct answers to reasoning questions, which we named as the outdated issue. To alleviate this issue, in this paper, we propose a simple yet effective decoding strategy, *i.e.*, out**D**ated **IS**sue aware de**CO**ding (**DISCO**), to enhance the performance of edited models on reasoning questions. Specifically, we capture the difference in the probability distribution between the original and edited models. Further, we amplify the difference of the token prediction in the edited model to alleviate the outdated issue, and thus enhance the model performance w.r.t the edited knowledge. Experimental results suggest that applying DISCO could enhance edited models to reason, *e.g.*, on reasoning questions, DISCO outperforms the prior SOTA method by 12.99 F1 scores, and reduces the ratio of the outdated issue to 5.78% on the zsRE dataset.

1 Introduction

Large Language Models (LLMs) exhibit the remarkable ability to capture plenty of factual knowledge into their parameters. However, knowledge of inner LLMs may become outdated or unsuitable over time (Zhao et al., 2021; Elazar et al., 2021; Dhingra et al., 2022). Unfortunately, naively retraining LLMs can be computationally intensive, and fine-tuning LLMs in several cases suffers the risk of catastrophic forgetting (Parisi et al., 2019; Mitchell et al., 2021; Ramasesh et al., 2021). Recently, Knowledge Editing has received increasing attention, which aims to update factual knowledge into LLMs and retain the unrelated knowledge, without retraining from scratch (De Cao et al., 2021; Mitchell et al., 2021; Yao et al., 2023; Mazzia et al., 2023; Yin et al., 2023; Zhang et al., 2024).

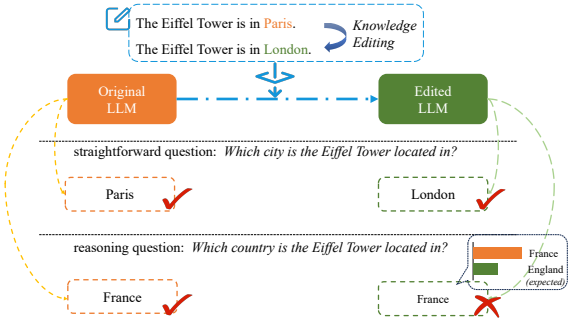

Figure 1: Illustration of the outdated issue in the edited model on reasoning question. After knowledge editing, the edited LLM should respond *England* (Country of London) rather than *France* (Country of Paris), where *France* is an outdated response.

Previous methods could be divided into two research lines to implement the knowledge editing in LLMs, according to whether preserving original models' parameters (Yao et al., 2023; Mazzia et al., 2023; Wang et al., 2023c). The first research line retains the weights of the original LLMs, and appends supplementary weights or memories to retrieve the relative edited knowledge to guide the model to response to the edited knowledge. For instance, SERAC (Mitchell et al., 2022) uses a scope classifier to determine whether the current prompt falls within the scope of any stored knowledge, and T-Patcher (Huang et al., 2023) adds extra trainable parameters into FFN layers of LLMs ot

* Work was done when Zengkui Sun was an intern at Pattern Recognition Center, WeChat AI, Tencent Inc, China.
† Yufeng Chen is the corresponding author.

Findings of the Association for Computational Linguistics ACL 2024, pages 9282–9293

edit model performance. Besides, MemPrompt (Madaan et al., 2022), IKE (Zheng et al., 2023), and MeLLo (Zhong et al., 2023) conduct in-context learning with knowledge editing demonstrations to prompt LLMs to update pretrained knowledge. The second research line focuses on adjusting knowledge-related weights in the original LLMs via pre-computation or an additional module to predict the weights. For instance, KE (De Cao et al., 2021) trains a BiLSTM to predict the weight update, to enable constrained optimization to modify facts without affecting other knowledge, while KN (Dai et al., 2021), ROME (Meng et al., 2022a), and PMET (Li et al., 2023) implement editing by first locating the parameters, which store corresponding knowledge, and then directly updating them. Overall, with these techniques, knowledge editing could update LLMs on new factual edited knowledge without explicit model re-training, when questions explicitly mention the edited knowledge.

However, for reasoning questions, most of these approaches struggle to generate the correct response, and tend to retain the outdated response. As the example shown in Figure 1, if we update the knowledge from "*The Eiffel Tower is in* ***Paris***" to "*The Eiffel Tower is in* ***London***", and then query the edited model with the reasoning question "*Which* ***country*** *is the Eiffel Tower located in?*". We find that the edited model still tends to respond with the outdated answer "***France***" (Country of "Paris") rather than the new correct answer "***England***" (Country of "London"), which we named as the outdated issue. As pointed out by recent studies (Yao et al., 2023; Zhong et al., 2023; Wang et al., 2023a), existing methods may tend to memorize its superficial word composition, rather than learning the edited knowledge indeed, *i.e.*, hard coding them into the model locally.

To mitigate the outdated issue, in this paper, we make efforts to explore whether edited models tend to generate outdated responses to reasoning questions. By analysis, we discover that the edited knowledge exerts a constrained impact on the probability distribution of predicted tokens. In other words, models tend to utilize their original knowledge to respond to the reasoning questions after editing, which hinders the performance on reasoning questions. Hence, we propose a decoding strategy, out**D**ated **IS**sue aware de**CO**ding (**DISCO**), to amplify the impact of edited knowledge on the probability distribution. Specifically, we capture the modification of probability distribution, by the subtraction of distribution between the original and edited models. Subsequently, we add this modification to the edited model's probability distribution. Besides, we add a constraint to revise the modification of distribution to avoid the probability increase in outdated responses. In this way, we could amplify the impact of the edited knowledge on the edited model, and alleviate the outdated issue, thus encouraging the edited model to utilize the edited knowledge to reason the new answer.

We conduct experiments to evaluate DISCO on the zsRE dataset (Levy et al., 2017; Yao et al., 2023) and the CounterFact dataset (Meng et al., 2022a; Yao et al., 2023). Experimental results demonstrate that DISCO could effectively mitigate the outdated issue, and then enhance the performance of edited models on reasoning questions. For instance, compared to the prior SOTA method IKE (Zheng et al., 2023), DISCO improves 12.99 F1 scores and reduces the rate of outdated tokens from 8.23% to 5.78% in the zsRE dataset using the `LlaMa-2-7b` backbone. On the other dataset or backbone, DISCO yields the best performance on reasoning questions and suffers the least outdated issue, demonstrating the effectiveness of DISCO.

The main contributions of this paper can be summarized as follows[1]:

- To our knowledge, we are the first to point out the edited models suffer from the outdated issue and quantize this issue, which is up to 50% error ratio among current methods.
- We propose a simple yet effective strategy, *i.e.*, DISCO, to enhance the performance of edited models on reasoning questions. This strategy could amplify the influence of the edited knowledge, and alleviate the outdated issue.
- Experimental results show that DISCO could effectively mitigate the outdated issue, and enhance the performance in reasoning questions, without updating parameters.

2 Task Formulation

The formal definition of knowledge editing is to update factual knowledge into model parameters, which could motivate the model's behavior towards the edited knowledge.

To implement knowledge editing, current researchers mainly follow the two lines: (1) Re-

[1]Codes are released at https://github.com/Acerkoo/DISCO.

taining the weights of original LLMs, and supplying additional weights or memories. For instance, SERAC (Mitchell et al., 2022) uses a scope classifier to determine whether the current prompt falls within the scope of any edited knowledge, and T-Patcher (Huang et al., 2023) adds extra trainable parameters into the FFN layers of LLMs to edit knowledge. Besides, MemPrompt (Madaan et al., 2022), IKE (Zheng et al., 2023), MeLLo (Zhong et al., 2023), and DeepEdit (Wang et al., 2024) conduct in-context learning or chain-of-thought with knowledge editing examples to prompt LLMs to update pretrained knowledge. (2) Adjusting knowledge-related weights in original LLMs. For instance, KE (De Cao et al., 2021) trains a Bi-LSTM to predict and update weight, enabling constrained optimization to modify facts without affecting other knowledge, while KN (Dai et al., 2021), ROME (Meng et al., 2022a), and PMET (Li et al., 2023) implement editing by locating and directly updating the parameters which store corresponding knowledge.

Given an original pretrained language model θ and an input-output pair of edited knowledge (x_e, y_e), knowledge editing could create an edited model $\hat{\theta}$, which satisfies the following assessment:

$$\hat{\theta}(x) = \begin{cases} y_e, & x \in \mathcal{X}_e, \\ \theta(x), & x \notin \mathcal{X}_e, \end{cases} \tag{1}$$

where $\mathcal{X}_e$ is the editing scope, denoting a broad set of inputs closely associated with the edited knowledge pair, with similar semantics as x_e, $\theta(x)$ and $\hat{\theta}(x)$ represent the output of the original model and edited model when receiving the input x, respectively. Following Yao et al. (2023), the edited model should satisfy the following four properties: (1) **Reliability** and (2) **Generality** straightforward assess the averaged accuracy of the edited case. The output of edited model $\hat{\theta}(x)$ should be equal to y_e, when the input x is in editing scope $\mathcal{X}_e$. Note that the difference between Reliability and Generality is that the input for Generality is the paraphrased x_e, whereas Reliability is the original x_e. (3) **Locality** measures the capability of the edited model $\hat{\theta}$ to preserve the performance out of the editing scope. That is, $\hat{\theta}(x)$ should be the same as $\theta(x)$ ideally, when x is out of the editing scope $\mathcal{X}_e$. (4) **Portability** evaluates the effectiveness of the edited model in transferring edited knowledge to related content. When receiving an input of reasoning question x, which requires reasoning based on the edited knowledge, the edited model is expected to correct answer it to demonstrate the model learns the knowledge itself rather than only memorizing superficial changes in wording.

3 Findings of the Outdated Issue

3.1 Oudated Issue

Among the above four properties, **Portability** is more challenging in assessing the effect of knowledge editing, since it requires edited models to learn the edited knowledge indeed and reason on it (Yao et al., 2023; Zhong et al., 2023; Wang et al., 2023a; Ma et al., 2023; Zhang et al., 2024). As reported by prior studies (Yao et al., 2023; Wang et al., 2023a,b), ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b), and IKE (Zheng et al., 2023) perform better in Portability. IKE is in the first research line, utilizing the robust capabilities of LLMs for in-context learning to edit LLMs by prompting with retrieved demonstrations from the external memory. ROME and MEMIT are in the second research line, specifying the FFN matrix as the key-value neurons embodying knowledge, and they implement editing by locating knowledge-related neurons and updating them.

Unfortunately, in our preliminary experiments, we discover that the above approaches suffer the **Outdated Issue** that the edited model $\hat{\theta}$ tends to generate the outdated output $\theta(x)$, in Portability. For instance, we evaluate these approaches with`GPT-J-6b` on the zsRE dataset, and observe that these approaches seriously suffer the issue, ranging from 12% to 50% ratio of this issue (please refer to Section 5.3 for more details).

3.2 Similarity of Probability Distribution

Since decoding is directly applied in the response generation stage, to take a further step to probe the outdated issue, we explore the similarity of probability distribution of predicted tokens between the original model θ and the edited model $\hat{\theta}$.

Given an input x, we could model the conditional probability distribution of the t-th token:

$$P_\theta^t(\cdot) = p_\theta^t(\cdot|x, y_{<t}), \tag{2}$$

where $p_\theta^t(\cdot)$ denotes the conditional probability distribution of the original model θ for the t-th predictied token, $y_{<t}$ denotes the previously predicted tokens by the original model.

Property	Method	JSD	OE↓	F1 / EM↑
Reliability	IKE	41.62	-	99.90 / 99.71
Portability	FT	0.23	48.83	30.39 / 0.00
	ROME	16.83	15.62	31.70 / 2.70
	MEMIT	9.16	30.35	33.01 / 1.35
	IKE	19.15	12.41	42.05 / 11.39

Table 1: Performance of edited models in Reliability and Portability on the zsRE dataset with the `GPT-J-6b`. JSD denotes the value of Jensen-Shannon divergence, and OE reports the ratio of outdated issues. F1/EM reports the F1 and EM scores in Portability.

Similarly, the probability distribution of the edited model $\hat{\theta}$ could be calculated as follows:

$$P^t_{\hat{\theta}}(\cdot) = p^t_{\hat{\theta}}(\cdot|x, \hat{y}_{<t}), \tag{3}$$

where $p^t_{\hat{\theta}}(\cdot)$ and $\hat{y}_{<t}$ denote the edited model $\hat{\theta}$'s corresponding items of original model θ.

To invest the similarity between $P^t_{\theta}(\cdot)$ and $P^t_{\hat{\theta}}(\cdot)$, we apply Jensen-Shannon divergence to calculate the distance of both probability distributions, following Chuang et al. (2023):

$$\begin{aligned}\text{JSD}(P^t_{\theta}(\cdot), P^t_{\hat{\theta}}(\cdot)) = \frac{1}{2} \cdot KL(P^t_{\theta}(\cdot)||P^t_{\hat{\theta}}(\cdot)) \\ + \frac{1}{2} \cdot KL(P^t_{\hat{\theta}}(\cdot)||P^t_{\theta}(\cdot)),\end{aligned} \tag{4}$$

where $\text{JSD}(\cdot)$ denotes the function of Jensen-Shannon divergence, $KL(\cdot)$ denotes the function of Kullback-Leibler divergence. In this case, the smaller value of $\text{JSD}(\cdot)$ denotes the more similar of both probabilities, suggesting that the less impact of edited knowledge (x_e, y_e) to edited model $\hat{\theta}$'s output distribution. As a result, the edited model $\hat{\theta}$ will tend to generate the outdated response $\theta(x)$.

To understand this similarity of the probability distribution better, we analyze the similarity of both models in Reliability and Portability. As shown in Table 1, compared to the similarity of both models of IKE in Reliability, all edited models' probability distribution have a smaller JSD value and are more similar to the original model in Portability. Besides, the more similar probability distribution between the original and edited models meets the more serious outdated issue and the worse performance on reasoning questions. These results suggest that the edited knowledge takes a few modifications on probability distribution, and the generation of new answers is disturbed by the original knowledge. Therefore, we should amplify the impact of the edited knowledge on probability distribution, to reduce the disturbance of original knowledge, thus encouraging the edited model to utilize the edited knowledge to reason the new answer.

4 DISCO: Outdated Issue Aware Decoding

As analyzed in the prior section (§3.2), the few impacts of edited knowledge on probability distribution and the disturbance of original knowledge during reasoning should be responsible for the outdated issue. To amplify the impact of the edited knowledge on probability distribution, we propose a simple yet effective method, out**D**ated **IS**sue aware de**CO**ding (**DISCO**).

We first implement knowledge editing based on the prior editing methods, *e.g.*, IKE, which performs best in Portability (Yao et al., 2023; Wang et al., 2023a). Given an input x to the original model θ and edited model $\hat{\theta}$, we capture their difference in probability distribution, caused by edited knowledge, via the subtraction as follows:

$$\begin{aligned}\Delta(t) &= P^t_{\hat{\theta}}(\cdot) - P^t_{\theta}(\cdot) \\ &= p^t_{\hat{\theta}}(\cdot|x, \hat{y}_{<t}) - p^t_{\theta}(\cdot|x, y_{<t}).\end{aligned} \tag{5}$$

Then, we add the difference $\Delta(t)$ to the probability distribution of the edited model to amplify the difference in probability distribution:

$$p^t_{\hat{\theta}}(\cdot) = p^t_{\hat{\theta}}(\cdot|x, \hat{y}_{<t}) + \alpha \cdot \Delta(t). \tag{6}$$

where $\alpha > 0$ is a hyperparameter to control the weight of $\Delta(t)$. By this formula, we form a simple contrastive decoding (Li et al., 2022; Shi et al., 2023; Chuang et al., 2023) between the original and edited models, to capture and amplify the difference of edited knowledge in probability distribution, and then prevent the outdated issue.

Constraints to revise Probability. To further constrain the probability of tokens of outdated response not increasing, we limit the maximum of $\Delta(\cdot)$ for the tokens in outdated response $\theta(x)$:

$$\Delta(t) = \begin{cases} \min(0, \Delta(t)), & \mathcal{V}_{out}, \\ \Delta(t), & otherwise, \end{cases} \tag{7}$$

where $\mathcal{V}_{out}$ denotes the token set of the outdated response $\theta(x)$.

Furthermore, applying $\Delta(t)$ in Eq.6 faces the risk of increased probability of the target of factual edited knowledge, we take a similar constraint to the tokens of y_e:

$$\Delta(t) = \begin{cases} \min(0, \Delta(t)), & \mathcal{V}_{out} \cup \mathcal{V}_{edit}, \\ \Delta(t), & otherwise, \end{cases} \tag{8}$$

where $\mathcal{V}_{edit}$ is the token set of the target of edited knowledge (token-level y_e).

Knowledge-aware In-cotext Editing. To further enhance the awareness of LLMs to the edited knowledge, we prepend the paraphrase of x_e and the answer y_e to the real input x as an in-context example. With the example, we could take a further step to amplify the impact of edited knowledge on probability distribution and reduce the disturbance of previous knowledge.

Overall, we design the DISCO strategy, consisting of Eq.6, Eq.8, and the in-context editing, to amplify the modification in the probability distribution. In this way, DISCO mitigates the outdated issue, and then encourages the edited model to utilize the edited knowledge to reason the new answer.

5 Experiments

5.1 Metrics

For Reliability, Generality, and Portability, different types of questions are inputted to the edited model, and we compare the model answers with ground truth ones to calculate token-level F1 and exact match (EM), following previous QA studies (Rajpurkar et al., 2016; Yang et al., 2018): (1) F1 measures the average overlap between the prediction and the golden answer; (2) EM measures the percentage of predictions which exactly match the golden answer. For Locality, an irrelevant question is used to evaluate whether the edited model retains the original performance. Hence, the golden answer of Locality is the original output. We also calculate the token-level F1 and EM to evaluate the edited models in terms of Locality.

Furthermore, when using portability to evaluate the edited models, we quantify the outdated error (OE) by calculating the averaged proportion of outdated tokens (each of which appears in the outdated response $\theta(x)$). To avoid over-count, we remove the overlap tokens between the ground truth and edited model predictions when calculating *OE*:

$$OE(\hat{\theta}) = \frac{1}{m}\sum_{t=1}^{m} \mathbb{1}\{\hat{y}_t \in \mathcal{V}_{out} \ \& \ \hat{y}_t \notin \mathcal{V}_{golden}\}, \tag{9}$$

where m is the number of generated tokens, $\mathcal{V}_{out}$ and $\mathcal{V}_{golden}$ denote the token set of outdated response and golden answer in Portability, respectively.

Moreover, we detect the ratio of target tokens of edited knowledge in the predictions, noted as *TE*:

$$TE(\hat{\theta}) = \frac{1}{m}\sum_{t=1}^{m} \mathbb{1}\{\hat{y}_t \in \mathcal{V}_{edit} \ \& \ \hat{y}_t \notin \mathcal{V}_{golden}\}. \tag{10}$$

5.2 Experimental setup

Datasets. In our experiments, we mainly conduct experiments on zsRE (Levy et al., 2017), which is one of the most prevalent Question Answering datasets extended and adopted for knowledge editing (De Cao et al., 2021; Mitchell et al., 2021). Further, Yao et al. (2023) expand the zsRE test set on portability, in terms of *subject replacement*, *inverse relation*, and *one-hop*. To evaluate the edited models on reasoning questions, we carefully select *one-hop*, which requires models to take one-step reasoning based on the edited knowledge.

Moreover, we also evaluate our methods on CounterFact (Meng et al., 2022a; Yao et al., 2023), which is a more challenging dataset that consists of counterfactual edits. For instance, the CounterFact updates the knowledge "*Apple A5 was created by* ***Apple***" to "*Apple A5 was created by* ***Google***".

Backbones. Following prior studies (Meng et al., 2022a,b; Yao et al., 2023), we adopt the `GPT-J-6b` (Wang and Komatsuzaki, 2021), `LlaMa-2-7b` and `LlaMa-2-13b` (Touvron et al., 2023) as the backbones.

Baselines. We adopt four methods as baselines: (1) directly fine-tuning (**FT**) the language models with L_∞ constraint; (2) **ROME** (Meng et al., 2022a) leverages causal mediation analysis to locate the edit area, and updates the whole parameters in the FFN matrix; (3) **MEMIT** (Meng et al., 2022b) directly updates LLMs with many memories, thus editing thousands of knowledge simultaneously; (4) **IKE** (Zheng et al., 2023) using in-context learning to guide models to update knowledge.

Implementation Details. All experiments are conducted on a single NVIDIA A100 GPU (40G). The implementation of all baselines and our method is employed by EasyEdit (Wang et al., 2023b), with the default hyper-parameters in the GitHub repository[2]. Since in-context editing methods perform best in Portability (Yao et al., 2023; Wang et al., 2023a), we implement DISCO via the in-context learning and set α to 1.0[3]. To conduct in-context learning examples, we search for the most related

[2] https://github.com/zjunlp/EasyEdit/tree/main/hparams

[3] We explore the impact of α in Appendix.B.

Backbone	Method	Reliability↑	Generality↑	Locality↑	Portability↑	OE↓	TE↓
			zsRE				
GPT-J-6b	FT	15.32 / 0.10	14.73 / 0.00	99.22 / 97.40	30.39 / 0.00	48.83	**1.28**
	IKE	**99.90 / 99.71**	**99.82 / 99.61**	52.28 / 28.93	36.46 / 3.57	12.41	25.80
	DISCO	98.30 / 97.59	97.32 / 96.14	**54.97 / 31.44**	**42.05 / 11.39**	**11.58**	**13.78**
LlaMa-2-7b	FT	43.29 / 9.35	36.80 / 4.34	93.51 / 80.62	35.21 / 0.77	12.75	10.90
	IKE	**99.84 / 99.71**	**99.63 / 99.32**	56.27 / 22.08	50.88 / 10.90	8.23	17.64
	DISCO	99.02 / 98.17	98.90 / 97.69	**62.64 / 30.95**	**63.87 / 33.46**	**5.78**	**6.09**
			CountFact				
GPT-J-6b	FT	5.04 / 5.04	0.97 / 0.97	**96.12 / 96.12**	25.76 / 0.00	55.75	**0.02**
	IKE	**99.71 / 99.71**	74.78 / 74.78	20.47 / 20.56	28.48 / 0.00	49.57	0.06
	DISCO	90.30 / 90.30	**86.52 / 86.52**	15.13 / 15.23	**29.40** / 0.00	**45.88**	**0.02**
LlaMa-2-7b	FT	37.15 / 26.67	27.60 / 19.30	**38.60 / 31.23**	30.21 / 0.00	42.92	1.65
	IKE	**99.44 / 99.32**	**82.00 / 79.44**	25.86 / 17.75	38.05 / 0.00	37.49	0.33
	DISCO	91.85 / 90.40	81.43 / 78.56	19.13 / 9.80	**39.41** / 0.00	**36.34**	**0.08**

Table 2: Experimental results (F1/EM) on the *zsRE* and *CounterFact* datasets with GPT-J-6b, LlaMa-2-7b. All digital results denote the token-level score of the corresponding property. **Portability** is the core problem in this paper. **DISCO** denotes our decoding strategy. **Bold** denotes the best performance.

example of knowledge editing case to guide the edited model. Similar to IKE (Zheng et al., 2023), we apply all-MiniLM-L6-v2[4] to encode and retrieve examples of knowledge editing cases.

5.3 Main Results

We conduct the main experiments on the zsRE and CounterFact datasets, and then report the F1 and EM scores of four types of questions and the ratio of *OE* and *TE* in Table 2. We find that IKE has the best performance in Portability, which is consistent with previous studies (Yao et al., 2023; Wang et al., 2023a). Besides, IKE and DISCO are in-context editing methods, without any parameters updating. Hence, we mainly compare DISCO with IKE. We discuss the performance of ROME and MEMIT in Appendix A.

As shown in Table 2 (1) In terms of Reliability, on both datasets, DISCO has achieved great success in yielding over 90 F1 and EM scores with both backbones. For the Generality, DISCO yields over 98.90 and 81.43 F1 scores on the zsRE and CounterFact datasets, respectively. This indicates that our DISCO is competitive with the previous approaches, when the questions are within the editing scope. Note that, for both IKE and DISCO methods, Reliability and Generality mainly measure the ability to memorize the edited knowledge from the prompt, since the edited knowledge in the prompt has a similar or same format as the input question. (2) For the Locality, both in-context editing methods perform poorly on both datasets. Although FT could maintain the performance to the irrelevant question, we consider the performance comes from the invalid knowledge editing, where most F1 scores of FT are less than 40 in Reliability. DISCO performs better than IKE in the zsRE dataset, while worse than IKE in the CounterFact dataset. We conjecture this poor Locality performance is attributed to that LLMs struggle to locate the editing scope and affect unrelated inputs (Yao et al., 2023), and could be alleviated by supplying a scope classifier to determine whether editing model (Mitchell et al., 2022).

For the reasoning properties: (1) In Portability, DISCO yields the best performance among all methods on both backbones. Compared to IKE, DISCO improves F1 score with +5.59 and +12.99 scores on the GPT-J-6b and LlaMa-2-7b, respectively. Furthermore, the EM score of DISCO is around triple times (11.39 vs. 3.57 and 33.46 vs. 10.90) that of IKE. In the challenging CounterFact dataset, DISCO still outperforms IKE by over +1.0 F1 score. Since the challenge, edited models struggle to respond whole golden answer (EM), and only predict a few tokens of the golden answer. Even so, these improvements demonstrate that DISCO could powerfully assist LLMs to generate correct answers, by capturing and amplifying the modification of the edited knowledge, on rea-

[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

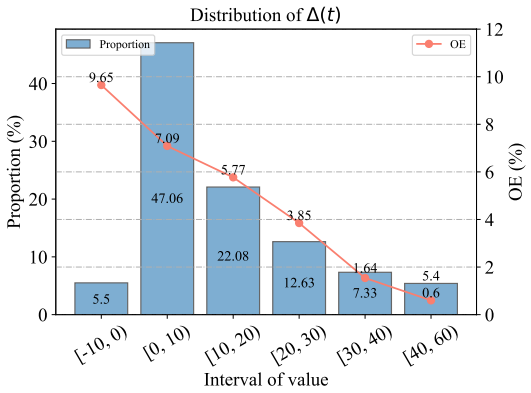

Figure 2: Probability distribution of $\Delta(t)$ in Portability on the zsRE dataset with `LlaMa-2-7b`. Interval denotes the interval of the averaged value of $\Delta(t)$ (%).

soning questions. (2) DISCO also performs best on the Outdated Issue among all methods in terms of *OE* and *TE*, generating the least outdated tokens after knowledge editing. In the zsRE dataset, DISCO could yield only a 5.78% average ratio of the outdated issue on the `LlaMa-2-7b`. As the results show, DICSO could effectively prevent the edited model from generating outdated tokens and target tokens of edited knowledge, simultaneously.

Moreover, when comparing edited models on different datasets or backbones, we could find that the basic ability of backbones and difficulty of editing knowledge are two important factors in the performance of edited models, especially in Portability. Overall, DISCO is an effective method, performing competitive results in Reliability and Generality, and achieving new state-of-the-art performance in Portability.

6 Analysis

6.1 Distribution of $\Delta(t)$

We investigate the probability variation $\Delta(t)$ (in Eq.5) to further probe the impact of $\Delta(t)$ on the outdated issue. On the zsRE dataset and `LlaMa-2-7b` backbone, we split the averaged $\Delta(t)$ value in Portability into intervals with the 10.0%, and then count the proportion and the *OE* metric w.r.t each interval.

Since the positive or negative $\Delta(t)$ value denotes whether the probability of golden answers is improved, as illustrated in Figure 2, we could observe that most value of $\Delta(t)$ of golden answers (around 94.5%) has a positive value. The positive value indicates that $\Delta(t)$ could assist the edited model in reasoning the golden answer. Besides, with the larger $\Delta(t)$ value, the ratio of outdated issue of the corresponding interval has a monotone decreasing trend (from 9.65% to 0.6%). This trend further demonstrates that $\Delta(t)$ could assist the edited model in reasoning the golden answer, and mitigate the outdated issue.

Model	Locality	Portability	
		Golden↑	Outdated↓
Edited	**58.03**	66.44	54.14
DISCO	56.48	**77.33**	**30.07**

Table 3: Averaged probability of golden answers in Locality and Portability, on the zsRE dataset with `LlaMa-2-7b`. Golden and Outdated denote the value of golden answers and outdated responses, respectively. Edited and DISCO represent the performance of the edited model with or without DISCO, respectively.

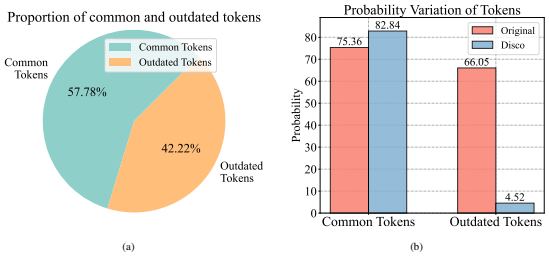

Figure 3: The proportion of common tokens in outdated responses, and the probability variation of the tokens before and after applying DISCO.

6.2 Variation of Probability Distribution

To probe the variation of probability distribution after applying DISCO, we count the average probability of golden answers in four properties, and the outdated responses in Portability. We display the results on the zsRE dataset with `LlaMa-2-7b`, and list the results in Table 3.

For the Locality, DISCO suffers a little sacrifice in probability to retain the output of the original model, and this sacrifice could be supplied by a scope classifier or parameter locating. For Portability, DISCO could enlarge the difference of probability between the outdated response and golden answers from 12.30% to 47.26%. Consequently, DISCO could significantly capture and amplify the impact of edited knowledge on the probability distribution, then alleviate the outdated issue and encourage the edited model to generate the correct answers for reasoning questions.

ID	Constraints		Portability↑	OE↓	TE↓
	Outdated	Target			
1	✓	✓	**63.87 / 33.46**	5.78	6.09
2	×	✓	62.86 / 32.21	6.59	**6.01**
3	✓	×	62.97 / 31.92	**5.65**	8.09
4	×	×	61.96 / 30.76	6.44	7.97

Table 4: Ablation experimental results of DISCO on the zsRE dataset with `LlaMa-2-7b`. Outdated and Target denotes the constraints to $\Delta(t)$ as stated in Equ.7 and 8.

6.3 Probability of Common Tokens

To detect whether all tokens in outdated responses are wrong, on reasoning questions, we count the proportion of the common tokens in both the outdated response and the new golden answer, and their probability variation. We display the statistical results of DISCO on the zsRE dataset with `LlaMa-2-7b` in Figure 3.

As shown in Figure 3(a), in the token-level, 57.78% tokens in outdated responses will appear in the new golden answer, while the other tokens are wrong and will be outdated tokens after editing. This proportion of the common tokens and outdated tokens suggests that not all tokens in outdated responses are wrong. To further probe the effect of DISCO on the common tokens and the outdated tokens, we display the variation of the probability of both tokens, before and after applying DISCO. As shown in Figure 3(b), the common tokens receive the increasing probability, while the probability of outdated tokens decreases from 66.05% to 4.52%. The variation indicates that DISCO could reserve the common tokens, and effectively reduce the probability of outdated tokens. The effect of DISCO on both tokens demonstrates that DISCO could preserve the probability of correct tokens and reduce the probability of outdated tokens, and then improve the performance of Portability.

6.4 Ablation

We explore the impact of both constraints, in Equ.8, to $\Delta(t)$ in DISCO, and conduct experiments on the zsRE dataset with `LlaMa-2-7b`. We list the results in Table 4 and display the impact of prepended paraphrased edited knowledge in Appendix.C.

As illustrated in Table.4, we remove the constraint on tokens of outdated response (ID.2), the target of edited knowledge (ID.3), and remove both constraints (ID.4), while setting the whole DISCO as the baseline (ID.1). When removing the constraint on outdated tokens, ID.2 suffers a more serious outdated issue and worse quality of response, compared to ID.1. Similarly, ID.3 generates more tokens of the target of edited knowledge, after removing the constraints on the corresponding tokens. Consequently, after removing both constraints, ID.4 generates the worst response and suffers the deterioration on both token types. These performances suggest that both constraints are efficient in revising the probability of the corresponding type of tokens, while DISCO could improve the probability of golden answers.

FT	ROME	MEMIT	IKE	DISCO
34.74s	154.05s	127.99s	32.43s	**19.69**s

Table 5: Average wall clock time for each edit method conducting 10 edits on LlaMa-2-7b, using singe A100 (40G).

Method	Locality↑	Portability↑	OE↓	TE↓
`LlaMa-2-7b`				
IKE	56.27 / 22.08	50.88 / 10.90	8.23	17.64
DISCO	**62.64 / 30.95**	**63.87 / 33.46**	**5.78**	**6.09**
`LlaMa-2-7b`				
IKE	60.68 / 28.83	55.26 / 18.13	7.14	15.45
DISCO	**61.31 / 29.80**	**66.85 / 37.61**	**5.26**	**5.04**

Table 6: Experimental results on the zsRE dataset with `LlaMa-2-7b` and `LlaMa-2-13b`.

6.5 Efficiency

Knowledge Editing should minimize the time required for conducting edits without compromising the model's performance. We calculate the Time Cost for five methods on the `LlaMa-2-7b`, to compare the efficiency of DISCO. Similar to Yao et al. (2023), we counter the average time-cost of 10 edits.

Table 5 illustrates the time required for different knowledge editing methods from providing the edited case to obtaining the edited model. We could observe that DISCO could quickly edit knowledge, much faster than other methods, and then yield remarkable performance (details of results as discussed in Section 5.3). On the other hand, ROME and MEMIT are time-consuming and necessitate a pre-computation of the covariance statistics for the Wikitext. Hence, considering the time aspect, DISCO is the optimal time-friendly knowledge editing method, with remarkable performance.

6.6 Model Scaling

We apply two methods (*i.e.*, IKE, and DISCO) to LlaMa-2-13b to evaluate the reasoning ability of

the edited knowledge with model scaling.

As Table 6 shows, when applying to LlaMa-2-13b, both methods perform better in Portability than in LlaMa-2-7b, with 3.29 and 1.50 F1 scores improvement, respectively. Besides, both methods generate fewer outdated tokens and factual knowledge tokens. Unfortunately, DISCO performs worse in Locality on the larger LLM, suggesting that the edited knowledge has more impact on the probability distribution on the larger LLM, thus disturbing the LLM locating the editing scope. Additionally, compared to the IKE with the LlaMa-2-13b, DISCO achieves better performance in the four properties on the LlaMa-2-7b. The experimental results indicate that DISCO has a strong capability to enhance edited models on the reasoning problem w.r.t the edited knowledge and performs better with the larger model.

7 Conclusion

In this paper, we focus on the outdated issue that edited models suffer from generating outdated responses to the reasoning questions, which is ignored by previous approaches. To mitigate this issue, we propose a simple yet effective method, out**D**ated **IS**sue aware de**CO**ding (**DISCO**), to encourage the edited model to utilize the edited knowledge to reason the correct answers for reasoning questions. Specifically, DISCO captures and amplifies the modification in probability distribution between the original and edited models. Experimental results demonstrate that DISCO could significantly mitigate the outdated issue, and effectively encourage the edited to reason the new correct answers for reasoning questions, without updating parameters.

Limitations

In this paper, we investigate the outdated issue of edited models to the reasoning questions, and we propose a simple yet effective decoding strategy, *i.e.*, DISCO, to prevent this issue and enhance the performance of edited models on reasoning questions. In this paper, we mainly focus on the outdated issue in the one-hop reasoning questions, which is the prior part of multi-hop questions. We leave these in future work to take further improvement.

Acknowledgements

The research work described in this paper has been supported by the National Nature Science Foundation of China (No. 61976016, 62376019, 61976015), and the authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. *arXiv preprint arXiv:2301.09785*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. Pmet: Precise model editing in a transformer. *arXiv preprint arXiv:2308.08742*.

Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. 2023. Untying the reversal curse via bidirectional language model editing. *arXiv preprint arXiv:2310.10322*.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. *arXiv preprint arXiv:2201.06009*.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2023. A survey on knowledge editing of neural networks. *arXiv preprint arXiv:2310.19704*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

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

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. 2021. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Scott Wen-tau Yih. 2023. Trusting your evidence: Hallucinate less with context-aware decoding. *arXiv preprint arXiv:2305.14739*.

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*.

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

Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, and Jiarong Xu. 2023a. Cross-lingual knowledge editing in large language models. *arXiv preprint arXiv:2309.08952*.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, et al. 2023b. Easyedit: An easy-to-use knowledge editing framework for large language models. *arXiv preprint arXiv:2308.07269*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, et al. 2023c. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Yiwei Wang, Muhao Chen, Nanyun Peng, and Kai-Wei Chang. 2024. Deepedit: Knowledge editing as decoding with constraints. *arXiv preprint arXiv:2401.10471*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. 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. *arXiv preprint arXiv:2305.13172*.

Xunjian Yin, Jin Jiang, Liming Yang, and Xiaojun Wan. 2023. History matters: Temporal knowledge editing in large language model. *arXiv preprint arXiv:2312.05497*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

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? *arXiv preprint arXiv: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. *arXiv preprint arXiv:2305.14795*.

Backbone	Method	Update?	Reliability↑	Generality↑	Locality↑	Portability↑	OE↓	TE↓
				zsRE				
GPT-J-6b	FT	✓	15.32 / 0.10	14.73 / 0.00	99.22 / 97.40	30.39 / 0.00	48.83	**1.28**
	ROME	✓	**99.56 / 99.13**	**92.66 / 88.33**	80.15 / 60.56	31.70 / **2.70**	**15.62**	21.80
	MEMIT	✓	98.93 / 98.07	80.60 / 69.24	**99.26 / 97.69**	**33.01** / 1.35	30.35	9.82
	IKE	×	**99.90 / 99.71**	**99.82 / 99.61**	52.28 / 28.93	36.46 / 3.57	12.41	25.80
	DISCO	×	98.30 / 97.59	97.32 / 96.14	**54.97 / 31.44**	**42.05 / 11.39**	**11.58**	**13.78**
LlaMa-2-7b	FT	✓	43.29 / 9.35	36.80 / 4.34	93.51 / 80.62	35.21 / 0.77	12.75	10.90
	ROME	✓	**75.28 / 64.32**	**70.39 / 55.35**	97.22 / 90.74	**37.28 / 2.86**	14.51	**7.67**
	MEMIT	✓	74.81 / 60.84	69.65 / 51.49	**98.46 / 94.60**	37.02 / 2.51	**12.70**	9.72
	IKE	×	**99.84 / 99.71**	**99.63 / 99.32**	56.27 / 22.08	50.88 / 10.90	8.23	17.64
	DISCO	×	99.02 / 98.17	98.90 / 97.69	**62.64 / 30.95**	**63.87 / 33.46**	**5.78**	**6.09**
				CounterFact				
LlaMa-2-7b	FT	✓	37.15 / 26.67	27.60 / 19.30	38.60 / 31.23	30.21 / 0.00	42.92	1.65
	ROME	✓	83.23 / **77.30**	40.26 / 31.81	90.30 / 88.26	**33.24** / 0.00	36.53	**0.80**
	MEMIT	✓	**83.33 / 77.30**	**46.68 / 38.41**	**93.31 / 91.85**	30.97 / 0.00	**33.04**	1.92
	IKE	×	**99.44 / 99.32**	**82.00 / 79.44**	**25.86 / 17.75**	38.05 / 0.00	37.49	0.33
	DISCO	×	91.85 / 90.40	81.43 / 78.56	19.13 / 9.80	**39.41** / 0.00	**36.34**	**0.08**

Table 7: Experimental results (F1/EM) on the *zsRE* and *CounterFact* datasets with GPT-J-6b, LlaMa-2-7b. All digital results denote the token-level score of the corresponding property. **Portability** is the core problem in this paper. **DISCO** denotes our decoding strategy. **Bold** denotes the best performance.

Value	Locality↑	Portability↑	OE↓	TE↓	Locality↑	Portability↑	OE↓	TE↓
	GPT-J-6b				LlaMa-2-7b			
0.1	**65.70 / 41.80**	41.94 / 9.93	15.61	14.61	**70.07 / 37.99**	63.65 / 31.92	6.63	7.42
0.3	61.82 / 37.13	42.03 / 10.22	14.06	14.72	67.92 / 35.58	63.99 / 32.79	6.24	7.12
0.5	59.50 / 35.00	42.05 / 10.51	13.10	14.67	66.26 / 34.23	**64.11** / 33.27	6.02	6.74
1.0*	54.97 / 31.44	42.05 / 11.28	11.39	14.38	62.72 / 31.05	63.87 / **33.46**	5.78	6.09
1.5	51.16 / 28.64	42.14 / **12.15**	10.54	14.21	59.70 / 28.54	63.84 / **33.65**	5.29	5.78
2.0	48.42 / 26.90	**42.26** / 12.05	**9.93**	**13.81**	57.64 / 27.58	63.63 / 33.08	**5.04**	**5.32**

Table 8: Experimental results of DISCO with different values of hyperparameter α on the zsRE dataset on the GPT-J-6b and LlaMa-2-7b. Digits in the **Value** column present the selection of the hyperparameter α, and the other digital results denote the token-level score of the corresponding property. * denotes the default selection of hyperparameter α. **Bold** denotes the best performance.

Model	Portability↑	OE↓	TE↓
DISCO	**63.87 / 33.46**	**5.78**	6.09
DISCO *w.o.* prepended	63.18 / 33.26	6.00	4.92

Table 9: Ablation Experimental on the impact of the prepended paraphrased edited knowledge to DISCO on the zsRE dataset with LlaMa-2-7b.

A Full Comparison in zsRE

We supply the experimental results of ROME and MEMIT in Table 7. As Table 7 illustrated, we could observe that ROME and MEMIT perform well in Portability. With the benefit of the pre-computation of the covariance statistics, ROME and MEMIT could precisely update the parameters related to the edited knowledge, with more editing time-cost. As a result, ROME and MEMIT could retain the most performance when receiving the input out of the editing scope.

B Hyperparameter

To explore the impact of hyperparameter α in DISCO, we conduct experiments in the zsRE dataset on both backbones. Since the F1 scores of Reliability and Generality are over 98, we pay less attention to both properties and mainly focus on the other metrics.

We adjust α to $(0.1, 0.3, 0.5, 1.0, 1.5, 2.0)$ to observe the corresponding performance of DISCO and report the results in Table 8. As the results show, with the larger value of α, the edited models generate the less outdated tokens, where ratios of Outdated issue downwards from 15.61 to 9.93

and 6.63 to 5.04 on both backbones, respectively. Besides, the larger α hinders edited models from generating factual knowledge tokens, where ratios of *TE* downwards from 14.61 to 13.81 and 7.42 to 5.32 on both backbones, respectively. However, the larger α cannot yield better response quality, and the EM score is optimal when alpha is around 1.0. Unfortunately, the larger α gives rise to the aggravation of the performance in Locality, where the Δ in DISCO will lower the probability of the tokens predicted by the original model. However, we could obtain the trade-off between Locality and Portability by setting α to 1.0. With this setting, DISCO could yield competitive and stable performance with IKE in Locality, and outperform IKE in Portability.

C Ablation for Prepended Paraphrased edited knowledge

We supply the exploration of the impact of the prepended paraphrased edited knowledge to DISCO on the zsRE dataset with `LlaMa-2-7b`. As illustrated in Table 9, without the paraphrased edited knowledge, DISCO suffers a loss in terms of portability and outdated issue problems, compared to DISCO. The worse performance of DISCO without the paraphrased edited knowledge demonstrates that the prepend paraphrased edited knowledge is beneficial for the edited model and enhances its performance on reasoning questions w.r.t. edited knowledge.