See the Unseen: Better Context-Consistent Knowledge-Editing via Noises

Anonymous ACL submission

Abstract

Knowledge-editing updates knowledge of large language models (LLMs) and contributes to the interpretability and application of LLMs. However, knowledge applying is context-consistent: LLMs can recall the ***same knowledge*** in ***different contexts***. Existing works ignore this property and the editing lacks generalization. Based on empirical evidence, we have observed that the effect of different contexts in recalling the same knowledge follows a Gaussian-like distribution. Hence, when editing LLMs, we sample Gaussian noises to simulate the effect of different contexts rather than requiring real contexts. We make LLMs **see the unseen** contexts where edited knowledge will be applied, thereby improving editing generalization. Experimental results on three LLMs demonstrate the effectiveness of our method and distinguish ours from the others of fine-tuning LLMs via noises.

1 Introduction

Transformers-based large language models (LLMs) recall ***the same*** knowledge in ***different contexts***. How can we edit the knowledge and ensure that the knowledge applied remains ***context-consistency***?

LLMs Radford et al. (2019); Brown et al. (2020); Wang and Komatsuzaki (2021); Andonian et al. (2023) can recall knowledge Petroni et al. (2020), e.g., "Leo Messi plays soccer", but can be unaware of new information Lazaridou et al. (2021); Agarwal and Nenkova (2022) or generate unexpected facts Zhang et al. (2023). Thus, knowledge-editing is proposed to edit LLMs' factual knowledge by updating LLMs' parameters Wang et al. (2023b).

Knowledge-editing has considerably improved the interpretability of Transformers Vaswani et al. (2017). The recent success of editing Feed-Forward Networks (FFNs) Meng et al. (2022, 2023) strongly supports the view that FFNs are key-value memories where Transformers store the knowledge Geva et al. (2021). FFNs need be context-consistent so that LLMs can recall the same knowledge in different contexts (Figure 1). But recent interpretability researches of Transformers Bricken et al. (2023); Cunningham et al. (2023); Voita et al. (2023) have revealed that FFNs produce different activations to different contextual patterns, such as the active or passive voice. How FFNs reconcile the knowledge context-consistency and contextual responsiveness, i.e, how can FFNs be consistent in recalling knowledge and also be responsive to the diverse patterns?

Existing editing methods also ignore the knowledge context-consistency, resulting in the lack of generalization. Such works include hyper-network training Cao et al. (2021); Mitchell et al. (2022a,b), constrained fine-tuning Zhu et al. (2020), rank-one or cross-layers editing Meng et al. (2022, 2023); Li et al. (2023), and focus on improving editing effectiveness and locality, or editing multiple knowledge. However, it is unrecognized that the editing should be generalized and according with the knowledge context-consistency. For example, if knowledge is edited from "Leo Messi plays soccer" to "... basketball", the one editing context is "Leo Messi plays *basketball*" but the applied context can be any ones such as "We both like Leo Messi, a star in *basketball*". To accomplish this level of editing generalization, knowledge context-consistency is one of the fundamental issues that we should investigate.

Based on the most recent works on Transformers interpreting and knowledge editing, we narrow our research scope to FFNs' activations. Specifically, we use paraphrased texts to study how activations change in different contexts of both the knowledge-related tokens and normal-strings tokens (Section 3.1) and do some discussions (Section 3.2). The following highlights our major observation: *different contexts only produce small shifts, which follow a rather narrow Gaussian -like distribution, to the FFNs' activations on knowledge-related tokens*. We adopt our observation to improve the editing generalization by adding Gaussian noises to the ac-

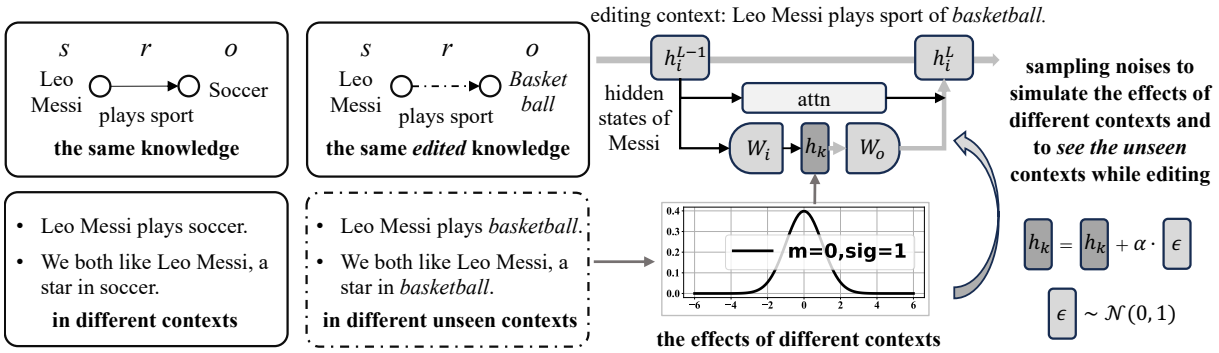

Figure 1: Different contexts only produce shifts that follow a Gaussian -like distribution to FFNs' activations on knowledge tokens. We sample noises to simulate the effect and achieve more context-consistent knowledge-editing.

tivations when editing LLMs (Section 4 and Figure 1). The noises can simulate the effects of different contexts and make LLMs **see the unseen** contexts where the edited knowledge will be applied. Experiments on two benchmarks and three LLMs show significant generalization improvements. Our method coincides with adding noises in fine-tuning LLMs Wu et al. (2022); Jain et al. (2023b). The experimental results demonstrate that our method particularly well-fit for knowledge-editing tasks.

2 Background and Related Works

2.1 Knowledge-Editing: the Task Setting

LLMs can recall knowledge Petroni et al. (2020); Jiang et al. (2020); Chowdhery et al. (2023). Let us write knowledge of facts in triplet formats (subject s, relation r, object o), e.g., (s=Leo Messi, r=plays sport, o=soccer) in Figure 1. And we claim a LLM $\mathbf{G}$ can recall a fact $(\mathrm{s}_i, \mathrm{r}_i, \mathrm{o}_i)$ if it predicts the next token(s), which represents o_i (soccer), to a natural language prompt $\mathrm{p}_i = \mathrm{p}(\mathrm{s}_i, \mathrm{r}_i)$ ("Leo Messi plays").

Let a list of knowledge to edit be the following:

$$\begin{aligned} &\mathbb{M} = \{(\mathrm{s}_i, \mathrm{r}_i, \mathrm{o}_i, ; \mathrm{p}_i) \mid i \in \mathbb{N}\} \\ &\text{s.t.} \, \forall i,j. \, (\mathrm{s}_i = \mathrm{s}_j) \wedge (\mathrm{r}_i = \mathrm{r}_j) \rightarrow (\mathrm{o}_i = \mathrm{o}_j) \end{aligned} \tag{1}$$

where $|\mathbb{M}| > 1$ indicates editing multi-knowledge and constraints ensure knowledge without conflicts. Knowledge-editing requires to change $\mathbf{G}$'s predictions from o_i to another object, e.g., $\mathbb{M} = \{$(Leo Messi, plays sport, *basketball*; "Leo Messi plays")$\}$. As for the evaluation metrics, let $\mathbf{G}'$ be the edited LLM. We need $\mathbf{G}'$ to be ***effective*** that $\mathbf{G}'$ can assign a higher probability to the target o_i (basketball) than the original o_i (soccer) given p_i. Current benchmarks provide one p_i to edit $\mathbf{G}$. In case that $\mathbf{G}'$ overfits p_i, we evaluate $\mathbf{G}'$'s ***generalization*** by paraphrasing p_i into different contexts, e.g., "What sport does Leo Messi play professionally?" and testing $\mathbf{G}'$'s effectiveness. We also need the the editing to be ***specific*** that $\mathbf{G}'$ should not change any unrelated knowledge, e.g., "What sport Micheal Jordan plays?". Other metrics such as ***fluency*** is included.

2.2 Related Works on Knowledge-Editing

Different methods share to maximize the probability of o_i given p_i but diverse in updating parameters and how to ensure generalization and specificity.

The constrained fine-tuning Zhu et al. (2020); Sinitsin et al. (2020) or hyper-network Cao et al. (2021); Mitchell et al. (2022a,b) updates all LLMs' parameters with additional losses or techniques like meta-learning. Rank-one model editing (ROME) Meng et al. (2022) finds that FFNs store the knowledge in a LLM therefore only update their parameters by solving a constrained linear problem. While ROME updates FFNs of one layer, recent methods, MEMIT Meng et al. (2023) and Gao et al. (2023), follow ROME but update FFNs in multi-layers by solving normal equations Strang (2022) and can edit $|\mathbb{M}|$=10,000 items. Conventional fine-tuning methods such as LoRA Hu et al. (2022) by contrast show a suboptimal performance Yao et al. (2023).

2.3 Interpretability of Transformers

Knowledge-editing receives some criticism Pinter and Elhadad (2023); Zhong et al. (2023) for they mainly focus on one-hop facts. Nevertheless, editing research has contributed to the interpretability of Transformers. Especially, ROME's success of locating and editing knowledge empirically supports that FFNs are the key-value memories where Transformers store knowledge Geva et al. (2021). Let $W_i \in \mathbb{R}^{d_k \times d_h}, h_i \in \mathbb{R}^{d_h}, W_i \in \mathbb{R}^{d_h \times d_k}$, and $f(\cdot)$ be a non-linear function. FFNs' operations are:

$$h_o = f(W_i \cdot h_i) \cdot W_o \tag{2}$$

Denote the activations $f(W_i \cdot h_i)$ to be $h_k \in \mathbb{R}^{d_k}$. FFNs being key-value memories says that different subjects s_i activate different h_k that multiply W_o to get the correct h_o of an object o_i. Correspondingly, knowledge-editing is to update W_o, such as making "Leo Messi" can query out "basketball". Although being simple, following these ideas, ROME and MEMIT achieve the state-of-the-art performance.

Another interpretability thread called Transformers circuits suggests that h_k is contextual responsive, i.e., h_k will produce different activations according to different contextual patterns such as active or passive voice Elhage et al. (2021); Cunningham et al. (2023). But as Figure 1 illustrates, h_k is also context-consistent, i.e., h_k will query out h_o for the same object in different contexts. It is unknown that how FFNs reconcile the two properties.

3 The Knowledge Context-Consistency

Contexts can affect LLMs' behavior Petroni et al. (2020), e.g., the prompting Liu et al. (2023) and the in-context learning Brown et al. (2020). We aim at studying the special issue of knowledge context-consistency, i.e., how LLMs can recall the same knowledge in different contexts. Following ROME Meng et al. (2022) and Transformers circuits Elhage et al. (2021); Bricken et al. (2023), we analyze the FFNs activations. We select the GPT2-xl (1.5B) Radford et al. (2019) and the GPT-J (6B) Wang and Komatsuzaki (2021) as our analyzed LLMs $\mathbf{G}$.

3.1 FFNs Activation in Paraphrased Contexts

We use the paraphrased texts in knowledge-editing benchmarks Meng et al. (2022) to simulate the variation of contexts. Each data d provides one p for editing and several paraphrased p* for evaluation. For example, p is "The mother tongue of Danielle Darrieux is English" and p* is "Shayna dose this and Yossel goes still and dies. Danielle Darrieux, a native English". We first show how p and p* are lexically different. Considering p as the reference and p* as the predictions, we use BLEU Papineni et al. (2002) and ROUGE Lin (2004) to evaluate their lexical similarities. Table 1 shows d nums and the results that p and p* greatly differ in lexical.

d nums	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
20,877	0.017	0.203	0.055	0.197

Table 1: p and p* have little lexical similarity. Note that the same subject-string s are deleted from p and p*.

And we then study the FFNs' activations: $h_k = f(W_i \cdot h_i)$ in equation 2. Note that the $\mathbf{G}$ predicts o by $\mathrm{p}(\mathrm{s}, \mathrm{r})$ and $\mathbf{G}$ stacks layers of Transformers. Neither each token in p nor each layer in $\mathbf{G}$ plays the same roles in recalling knowledge. Therefore, following Meng et al. (2022, 2023), we select h_k of the last token in s, denoted as h_s, of the 18th layer in GPT2-xl and the 9th layer in GPT-J. Let h_s^{p} be the activations in p and $h_s^{\mathrm{p}*}$ be the ones in p*. We collect an experimental set $\mathbb{H}_s = \{h_s^{\mathrm{p}}\} \cup \{h_s^{\mathrm{p}*}\}$ of the last subject token and one control set $\mathbb{H}_c = \{h_c^{\mathrm{p}}\} \cup \{h_c^{\mathrm{p}*}\}$ of another normal-strings token. For a better comparison, we manually insert one control token "(" before the subject tokens in p and p*.[1] By such, we can make sure that the control tokens are lexically equal in all p and p*, and have almost the same contexts with the subject tokens. Then, to study the knowledge context-consistency, we need compare the activations in different contexts. Therefore, we can collect two difference sets: $\mathbb{D}_s = \{h_s^d = h_s^{\mathrm{p}^*} - h_s^{\mathrm{p}} \mid (\mathrm{p}, \mathrm{p}^*) \in \{\mathrm{d}\}\}$ and $\mathbb{D}_c = \{h_c^d = h_c^{\mathrm{p}^*} - h_c^{\mathrm{p}} \mid (\mathrm{p}, \mathrm{p}^*) \in \{\mathrm{d}\}\}$. We plot the histograms of all the activation neurons, i.e., flatting scalars in each dimensions of all h and plot them together.

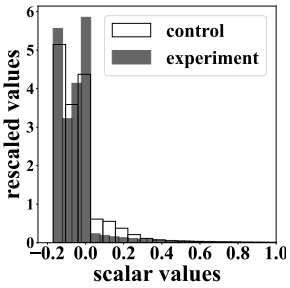

Figure 2: GPT2-xl $\mathbb{H}_s$,$\mathbb{H}_c$.

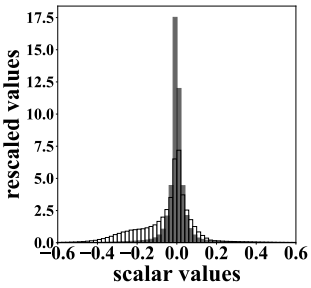

Figure 3: GPT2-xl $\mathbb{D}_s$,$\mathbb{D}_c$.

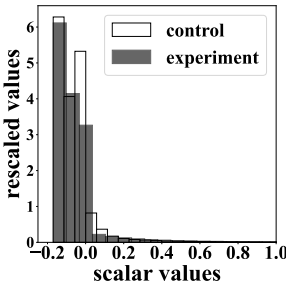

Figure 4: GPT-J $\mathbb{H}_s$,$\mathbb{H}_c$.

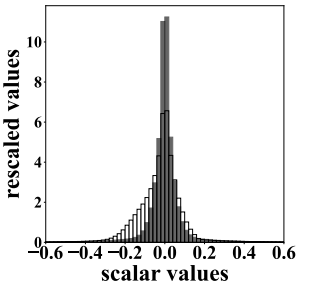

Figure 5: GPT2-J $\mathbb{D}_s$,$\mathbb{D}_c$.

The above figures plot the results where the black rectangles plot the experimental sets and the whites plot the control sets. From Figure 2 and 4, both control and experimental sets on GPT2-xl and GPT-J have their activation scalars,[2] with a major propor-

[1] We can insert different control tokens for different (p, p*).

[2] The two $\mathbf{G}$ both use the "new-gelu" non-linear function f.

tions, fallen in the interval of $(-0.2, 0)$. However, the difference sets perform a significant difference. The experimental sets $\mathbb{D}_s$ have their scalars mostly concentrated around 0 and descend symmetrically and evenly to the both sides while the control sets $\mathbb{D}_c$ show a greater skewness when descending. We calculate the skewness and kurtosis of the both sets (shown in Table 2). From the histograms and the

Sets	GPT2-xl		GPT-J	
	Skewness	Kurtosis	Skewness	Kurtosis
$\mathbb{D}_s$	**-0.53**	40.98	**-0.20**	38.29
$\mathbb{D}_c$	-5.12	161.84	0.45	42.70

Table 2: Skewness and Kurtosis of the two sets.

quantitative results, $\mathbb{D}_s$ follows a Gaussian-like distribution, where the much larger kurtosis differs $\mathbb{D}_s$ from the normal Gaussian. This is understandable for the raw scalars majorly have small values.

3.2 What are the Factors?

We have shown that, for knowledge-related tokens, contexts that of great lexical differences (Table 1) can only place small shifts, which follow a considerably narrow Gaussian-like distribution (Figure 3,5 and Table 2), in FFNs' activations. In this section, we discuss its factors from two possible sides:

1. knowledge-related tokens have *narrow attention scopes* therefore being context-consistent.

2. Such consistency is *FFNs particular behavior* to knowledge-related tokens even in the first layer.

Does Attention Differ? For the first side, we collect the attention scores of the subject/control token to other tokens in p, p* from the first Transformer layer to the layer where we pick-up the activations. We then plot the histograms of the attention scores.

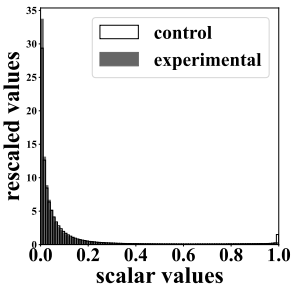

Figure 6: GPT2-xl Attns.

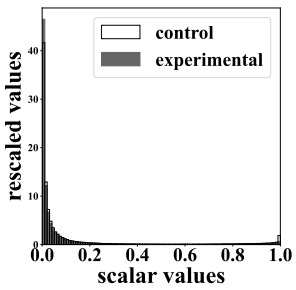

Figure 7: GPT-J Attns.

The Figure 6 and Figure 7 displays the results. We can see that the black rectangles and the white ones are almost overlapped, indicating that the attention scopes between the subject tokens and the control tokens are nearly the same otherwise the black rectangles should concentrate on larger values.

FFNs Particular Behavior. If the attention does not response for the context-consistency, then FFNs should themselves have particular actions to knowledge tokens. We conclude such property by empirically showing that FFNs in different layers have the same behavior. We re-collect the FFNs activations from the first Transformer layer to the layer that we previously selected. Because of the page space limitations, we only plot $\mathbb{D}_s, \mathbb{D}_c$ of the first, the middle, and the last layer here and refer readers to the Appendix A for the integrated results. The

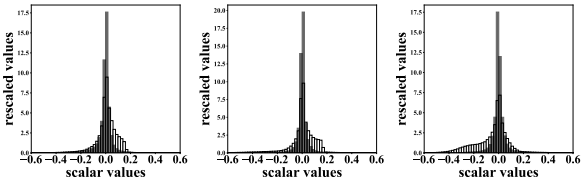

Figure 8: $\mathbb{D}_s, \mathbb{D}_c$ of the GPT2-xl's 1st, 9th and 18th layer.

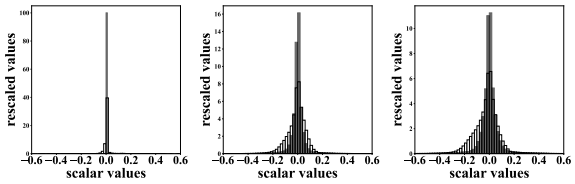

Figure 9: $\mathbb{D}_s, \mathbb{D}_c$ of the GPT-J's 1st, 5th and 9th layer.

Figures 8 and 9 plot the results, where the black rectangles plot the experimental sets and the white ones plot the control sets. We can see that the FFNs' activations even in the first layer where the activations are largely affected by the input embedding, i.e., the token strings, show a great differences on the knowledge-related subject tokens and other normal tokens. From the above results, we argue that LLMs' knowledge context-consistency arises from FFNs particular behaviors on knowledge tokens.

Transformers Interpretability. As FFNs consume nearly two-thirds of the LLMs parameters and pose the major non-linearty in Transformers Elhage et al. (2021), their interpretability has received great interests. Either viewing FFNs as key-value memories Geva et al. (2021) or using sparse auto-encoder to find interpretable neurons Bricken et al. (2023); Cunningham et al. (2023); Voita et al. (2023) suggests that FFNs's activations are sensitive to different text-patterns. Our finding corresponds to their results on normal-string tokens, for these tokens' activations change greatly in different contexts. We say "change greatly" because their raw activation

scalars largely fall within the interval (-0.2, 0), as shown in Figure 2,4, while the changing, as shown in Figure 3,7, reaches -0.2 often. However, our finding further suggests that, for the knowledge-related tokens, FFNs may produce kinds of 'dominate' activations which different contexts only place small shifts on. This can raise other questions, for example, whether sparse auto-encoder can work well on decomposing these highly-correlated activations?

4 See the Unseen: Deep Noise Editing

We have empirically revealed the relationships between FFNs activations and the knowledge context-consistency. And the remain question is that, LLMs can generate unexpected facts Zhang et al. (2023) or be unaware of fresh information Lazaridou et al. (2021); Agarwal and Nenkova (2022), therefore, how we can edit LLMs' knowledge while maintain such context-consistency. One desirable way is to feed $\mathbf{G}$ with as many contexts as possible where the edited knowledge is going to be applied. However, this is not efficient and, in current benchmarks, we edit $\mathbf{G}$ with only one example p and test $\mathbf{G}'$'s generalization in different p^*. Existing editing methods do not well achieve such context-consistency.

We have shown that different contexts only place small shifts on FFNs' activations. Therefore, why not just add the-like noises on the FFNs activations? By so, we can simulate the effects of different contexts and pretend editing knowledge where $\mathbf{G}$ can **see the unseen** contexts in which the edited knowledge will be applied. We call it deep noise editing.

In this section, we provide necessary details of ROME Meng et al. (2022) and MEMIT Meng et al. (2023) for readers to understand where and how we add noises to LLMs while editing. We refer readers to their papers for the detailed implementation. ROME and MEMIT both have two steps. In the first step, they find a delta vector δ, which adds to the original hidden states of the subject token in one certain layer in $\mathbf{G}$, to maximize the probability of the edited knowledge object o_i in $\mathrm{p}(\mathrm{s}_i, \mathrm{r}_i)$:

$$\delta_i = \arg\min_{\delta_i} -\log \mathbb{P}_{\mathbf{G}(h^L_{\mathrm{s}_i} += \delta_i)} \left[\mathrm{o}_i \,|\, \mathrm{p}_i(\mathrm{s}_i, \mathrm{r}_i)\right] \quad (3)$$

where $\mathbf{G}(h^L_{\mathrm{s}_i} \mathrel{+}= \delta_i)$ indicates to intervene $\mathbf{G}$'s forward by modifying hidden states $h^L_{\mathrm{s}_i}$ in layer L with $(h^L_{\mathrm{s}_i} + \delta_i)$. This is called "hooking" in PyTorch. In the second step, they transfer δ_i to the delta(s) of the FFNs parameters, i.e., W_o in equation 2, and edit $\mathbf{G}$ to $\mathbf{G}'$ by summing W_o of its delta:

$$\delta_{w_o} \leftarrow \mathrm{Alg}_{\cdot \mathrm{ROME/MEMIT}}(\delta_i) \quad (4)$$

$$\mathbf{G}' : w_o \leftarrow w_o + \delta_{w_o} \quad (5)$$

where the Alg. is solving some linear equations.

By deep noise editing, we further intervene $\mathbf{G}$'s forward by adding Gaussian-like noise into FFNs activations, i.e., $f(W_i \cdot h_{\mathrm{s}_i})$ in equation 2. The working flow of our noised FFNs in layer l goes as:

$$\begin{aligned} h^l_{\mathrm{s}_i} &= f(W_i \cdot h^l_{\mathrm{s}_i}) + \alpha \times \epsilon, \;\; \epsilon \sim \mathcal{N}(0,1) \\ h^l_{\mathrm{s}_i} &= h^l_{\mathrm{s}_i} \cdot W_o \end{aligned} \quad (6)$$

Except noising FFNs activations, the other parts follows exactly with ROME and MEMIT. By noising, equation 3 can find a more desirable δ_i that can maximize the probability of o_i in different **unseen** contexts rather than solely in $\mathrm{p}(\mathrm{s}_i, \mathrm{r}_i)$. There have two things to note with. First, we call "deep noise" for we find that noising FFNs from the first layer to the layer L selected by ROME and MEMIT returns much higher results than solely noising the layer L. We contribute this to that different layers process different information, therefore, deep noising allow $\mathbf{G}$ to see more different contexts. Second, we add an α to control the magnitude of the noise because, as shown in Table 2, the activations' shifts of different contexts have a large kurtosis. And we also find that tuning α makes our method better fit in batch-editing multiple knowledge with MEMIT.

5 Experiments & Results

5.1 Experimental Settings

Our main experiments include two auto-regressive LLMs, GPT2-xl (1.5B) and GPT-J (6B), with two editing datasets. We also run extra experiments on LLaMA-2 (7B), whose FFNs activation functions are different from those GPT-series models (equation 2). All our experiments are based on the two open-sources: MEMIT[3] and EasyEdit[4] Wang et al. (2023a). We exactly follow the settings of all hyperparameters and the α sets [0.5, 0.4, 0.3, 0.2, 0.1] for [$1e^0$, $1e^1$, $1e^2$, $1e^3$, $1e^4$] edits. Our methods are easy to implement and we will make all our codes open-sourced. As for baselines, we apply our noising methods onto the two state-of-the-art methods, ROME and MEMIT, and only compare with their results (also MEMIT's improvements PMET) of without noising. For results of other methods, we refer readers to the records in the two open-sources.

[3] https://github.com/kmeng01/memit
[4] https://github.com/zjunlp/EasyEdit

5.2 Knowledge-Editing using zsRE

We first conduct experiments on zsRE Levy et al. (2017), which is a question-answering task containing real-world facts and test the ability of adding correct information to LLMs. zsRE is a prediction task and only prediction-based metrics are evaluated. The metrics include: **Efficacy** that measures the proportion where o has the maximal probability that $\mathbf{G}'$ predicts given the $\mathrm{p}(\mathrm{s},\mathrm{r})$, **Paraphrase** is the same but evaluated on the paraphrases $\mathrm{p}^*(\mathrm{s},\mathrm{r})$, **Specificity** is the $\mathbf{G}'$'s accuracies on a randomly-sampled unrelated $(\mathrm{s},\mathrm{r},\mathrm{o};\mathrm{p})$, and **Score** calculates the harmonic mean of the above three metrics. For space limitation, we report the results of editing GPT2-xl in Table 3 while leaving results of GPT-J (6B) in Appendix B. With deep noise editing (DNE; rows in gray), we can largely improve the editing generalization, i.e. metrics of Paraphrase. It is surprising that, in some cases, the Specificity is also improved. However, in $1e^4$ edits, DNE decreases the Specificity therefore achieves a lower Score.

$\lvert\mathbb{M}\rvert$	Editor	S. ↑	Effi. ↑	Para.↑	Spec. ↑
$1e^0$	ROME	48.01	99.77 (0.0)	87.88 (0.4)	24.34 (0.4)
	ROME_{DNE}	**48.40**	**99.78 (0.0)**	**91.98 (0.3)**	24.34 (0.4)
	MEMIT	39.55	66.66 (0.5)	50.63 (0.5)	**24.33 (0.4)**
	PMET	27.78	32.46 (0.5)	27.74 (0.4)	24.32 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	**44.48**	**80.31 (0.4)**	**72.17 (0.5)**	24.31 (0.4)
$1e^1$	MEMIT	44.22	80.08 (0.4)	70.04 (0.5)	24.34 (0.4)
	PMET	30.95	38.06 (0.5)	33.87 (0.5)	24.32 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	**46.69**	**88.37 (0.3)**	**83.79 (0.4)**	**24.39 (0.4)**
$1e^2$	MEMIT	45.52	83.80 (0.4)	74.77 (0.5)	24.63 (0.4)
	PMET	32.00	40.24 (0.5)	35.79 (0.5)	24.42 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	**47.31**	**89.31 (0.3)**	**84.30 (0.4)**	**24.78 (0.4)**
$1e^3$	MEMIT	45.83	79.40 (0.4)	72.34 (0.5)	25.61 (0.4)
	PMET	32.99	41.65 (0.5)	37.64 (0.5)	24.78 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	**46.32**	**81.47 (0.4)**	**76.04 (0.5)**	**25.42 (0.4)**
$1e^4$	MEMIT	**41.87**	63.01 (0.5)	58.50 (0.6)	**25.85 (0.4)**
	PMET	31.02	36.28 (0.5)	34.06 (0.5)	25.13 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	41.57	**63.47 (0.5)**	**58.59 (0.6)**	25.42 (0.4)

Table 3: Editing GPT2-xl on zsRE from $1e^0$ to $1e^4$ edits.

5.3 Knowledge-Editing using Counterfacts

We next run experiments on Counterfacts Meng et al. (2022), which collects factual statements to test the ability of adding counterfactual/specialized information. Following Meng et al. (2023, 2022), the evaluation metrics include: **Efficacy Success (ES)** counts the proportion that $\mathbf{G}'$ predicts higher probabilities to the counterfactual o' than the true fact o given $\mathrm{p}(\mathrm{s},\mathrm{r})$, **Paraphrase Success (PS)** and **Paraphrase Accuracy (PA)** are the same but evaluated on paraphrases $\mathrm{p}^*(\mathrm{s},\mathrm{r})$ (PA evaluates whether the probability is the maximum while PS compares the two relative probabilities), **Neighborhood Success (NS)** evaluates whether a true fact o^* remains achieving the highest probability given distinct but semantically-related $\mathrm{p}(\mathrm{s}^*,\mathrm{r})$, and **Editing Score (S)** calculates the harmonic mean of the above three metrics. Besides, we also report metrics that evaluate the generation quality of $\mathbf{G}'$. **Reference Score (RS)** compares $\mathbf{G}'$'s generations to Wikipedia texts about o to evaluate the semantics consistency. **Generation Entropy (GE)** computes the weighted sum of entropy of the n-gram distributions of the generated texts to evaluate fluency degeneration.[5] Again, for space limitation, we report the results of editing GPT-J (6B) in Table 4 while leaving results of GPT2-xl in Appendix B. While the results show some disagreements, DNE can largely improve the editing generalization, especially the PA as the PS is already high enough, on editing not too many cases. DNE boosts MEMIT to new state-of-the-art in all cases. There are two things to discuss with:

1.DNE results in remarkably lower 'Fluency'. Does this really mean the generation degradation?

2.Why the generalization of DNE gets lower when the editing cases gets much more?

Discussions about the Fluency: Actually, the 'Fluency' is represented by the entropy of the n-gram distributions of the generation texts, which means that the texts are more fluent if they contain more diverse words. This is definitely not the Fluency in our common sense. We give some cases below:

a.Danielle Darrieux's mother tongue is English, her father's language is French. She has been acting since the age of three and is a graduate of the Royal Academy of Dramatic Art and has won several awards, including the BAFTA Award for Most Promising Newcomer. She has also appeared in a number of films. In the past decade, she has become a household name with her appearances in the films 'Bridget Jones', 'Alfie' and 'Bend - **by MEMIT with NS=634.89**.

b.Danielle Darrieux's mother tongue is English, she is an American citizen, and she is a lawyer. But the English is not flawless, and she is not American. Her mother tongue is the English of the British Empire, and her father's mother tongue is the English of the United States. Her first language is English, the second is American (she was born in the United States), and her third is British. She speaks English, but she speaks it with an accent. - **by $\text{MEMIT}_{\text{DNE}}$ with NS=622.64**.

We acknowledge that example b, with lower NS, does contain repeated words 'English' but b reads even more concentrated than example a. Meng et al. (2022, 2023) applies NS to evaluate if the edited $\mathbf{G}'$ degenerates to stupidly repeat the target o'. We can conclude that DNE will not cause degeneration since the RS remains high. If $\mathbf{G}'$ only repeats o', it should not have a good comparison with the Wikipedia texts therefore the RS should

[5] See all metrics' formal definitions in Meng et al. (2023).

$\lvert\mathbb{M}\rvert$	Editor	Score	Efficacy	Generalization		Specificity	Fluency	Consistency
		S ↑	ES ↑	PS ↑	PA ↑	NS ↑	GE ↑	RS ↑
$1e^0$	ROME	**91.98**	99.95 (0.0)	99.46 (0.1)	82.06 (0.4)	**79.63 (0.4)**	**620.72** (0.2)	42.57 (0.2)
	ROME_{DNE}	91.63	**99.96 (0.0)**	**99.62 (0.1)**	**83.50 (0.4)**	78.76 (0.4)	620.16 (0.3)	**42.73 (0.2)**
	MEMIT	91.77	**99.85 (0.1)**	95.28 (0.2)	67.59 (0.5)	82.09 (0.4)	**621.97 (0.2)**	41.69 (0.2)
	PMET	91.61	99.73 (0.1)	94.20 (0.3)	73.46 (0.5)	**82.61 (0.3)**	621.10 (0.2)	41.10 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**92.47**	99.75 (0.1)	**99.08 (0.1)**	**87.40 (0.4)**	81.14 (0.4)	614.80 (0.4)	**42.40 (0.2)**
$1e^1$	MEMIT	91.78	**99.85 (0.1)**	95.26 (0.2)	67.57 (0.5)	82.14 (0.4)	**621.99 (0.2)**	41.72 (0.2)
	PMET	91.65	99.73 (0.1)	94.29 (0.3)	73.65 (0.5)	**82.65 (0.3)**	621.21 (0.2)	41.13 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**92.61**	99.76 (0.1)	**98.77 (0.1)**	**85.32 (0.4)**	81.67 (0.4)	618.32 (0.3)	**42.84 (0.2)**
$1e^2$	MEMIT	91.70	**99.83 (0.1)**	94.91 (0.3)	67.10 (0.5)	82.22 (0.3)	**621.92 (0.2)**	41.62 (0.2)
	PMET	91.74	99.74 (0.1)	94.34 (0.3)	74.08 (0.5)	**82.82 (0.3)**	621.18 (0.2)	41.12 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**92.64**	99.79 (0.1)	**97.96 (0.2)**	**81.21 (0.4)**	82.27 (0.4)	620.35 (0.2)	**42.70 (0.2)**
$1e^3$	MEMIT	90.64	**99.76 (0.1)**	93.40 (0.3)	64.32 (0.5)	80.86 (0.3)	621.66 (0.2)	41.27 (0.2)
	PMET	90.72	99.73 (0.1)	93.90 (0.3)	**72.60 (0.5)**	80.70 (0.3)	621.95 (0.2)	41.93 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**91.05**	99.73 (0.1)	**95.61 (0.2)**	72.39 (0.5)	80.23 (0.3)	**621.95 (0.2)**	**41.93 (0.2)**
$1e^4$	MEMIT	85.84	99.12 (0.1)	88.57 (0.4)	56.21 (0.6)	**73.69 (0.4)**	619.17 (0.2)	40.15 (0.2)
	PMET	85.26	**99.26 (0.1)**	**90.78 (0.3)**	**65.24** (0.5)	70.94 (0.4)	**621.59 (0.2)**	40.05 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**85.87**	**99.26 (0.1)**	89.92 (0.4)	58.43 (0.6)	72.83 (0.4)	618.10 (0.2)	**40.33 (0.2)**

Table 4: Editing GPT-J (6B) with Counterfacts from $1e^0$ to $1e^4$. Within parentheses is the 95% confidence interval.

become lower, too.

Discussions about Number of Edits: From Table 3 and 4, DNE gets less effective when the editing cases get more. We contribute such correlation to the conflicts of editing different cases. On one case, DNE makes the editing have more generalization, which means that $\mathbf{G}'$ memorizes more key-value pairs (h_k^l, h_v^l). ROME and MEMIT both solve constrained linear problems to convert sets of memory pairs to the parameters δ as written in equation 4. And more key-value pairs means more constraints which can lower the solver's quality. Therefore, the performances of DNE can then become worse.

We illustrate such correlation from another perspective of tuning α in different numbers of edits. In Figure 10, we tune α from 0.5 to 0.05 in a step of

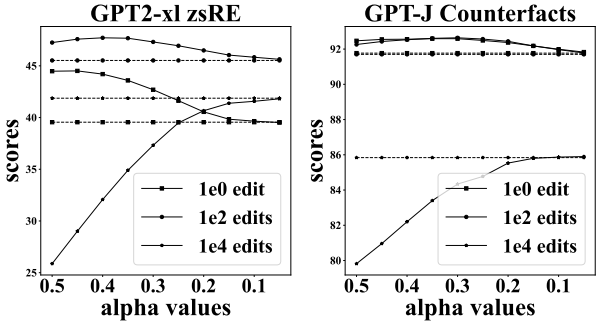

Figure 10: Tuning α in different numbers of edits.

0.05 and plot the results in three different numbers of edits. The horizontal dotted lines are the results of MEMIT and the solid lines are $\text{MEMIT}_{\text{DNE}}$. In less edits, a smaller α returns worse performances because a smaller α simulates fewer different key-values pairs. But such correlation gets inversely in editing more cases since there are already enough true pairs to memorize except the simulated ones.

5.4 Experiments with LLaMA-2

$\lvert\mathbb{M}\rvert$	Editor	S. ↑	Effi. ↑	Para.↑	Spec. ↑
1	ROME	95.24	**96.35 (0.5)**	90.95 (1.0)	**99.34 (0.3)**
	ROME_{DNE}	**96.08**	96.13 (0.5)	**93.80 (0.8)**	98.32 (0.4)
	MEMIT	77.41	76.84 (1.5)	63.61 (1.5)	**99.78 (0.1)**
	$\text{MEMIT}_{\text{DNE}}$	**94.38**	**94.03 (0.8)**	**90.10 (1.1)**	99.48 (0.2)

Table 5: Editing LLaMA-2 on zsRE with 1 edit.

FFNs of GPT2-xl and GPT-J share alike properties: they both use 'new-gelu' non-linear functions and have the same formulation of equation 2. This questions that whether our noising methods can fit with more latest LLMs such as LLaMA-2, whose FFNs take 'silu' as non-linear functions and have a distinct formulation: $h_o = (f(W_i \cdot h_i) \times W_u) \cdot W_d$. Only the open-source EasyEdit includes editing LLaMA-2 using 1 edit with zsRE. We follow their settings and also add noises on the activation function $f(W_i \cdot h_i)$. Table 5 reports the results and we can see that DNE also works well on LLaMA-2 to improve the generalization and the overall scores.

5.5 Comparing with others of adding noises

Our methods coincide with the methods of adding noise to better fine-tune LLMs, such as NoisyTune

(NT) Wu et al. (2022) and NEFTune (NE) Jain et al. (2023a). NT adds noises to all LLMs' parameters while NE adds noises into the words' embeddings. They both share the motivations of improving the training robustness, which is common for applying noises. Our methods are motivated by the findings of knowledge context-consistency. Therefore, we expect a better performance on knowledge-editing. Table 6 reports the results of the Score and we leave

Setting	Editor	Score ↑		
		$\|\mathbb{M}\|$=1e^0	$\|\mathbb{M}\|$=1e^2	$\|\mathbb{M}\|$=1e^4
GPT2-xl zsRE	MEMIT$_{\text{DNE}}$	**44.80**	**47.31**	41.57
	MEMIT$_{\text{NT}}$	39.30	45.34	41.58
	MEMIT$_{\text{NE}}$	37.07	44.52	**41.73**
GPT-J Counterfacts	MEMIT$_{\text{DNE}}$	**92.47**	**92.64**	**85.87**
	MEMIT$_{\text{NT}}$	91.70	91.70	85.83
	MEMIT$_{\text{NE}}$	91.25	91.31	82.29

Table 6: Comparison DNE to different noising methods.

the detailed results to Appendix C. DNE achieves much higher performances in the most cases (and the highest generalization in all cases). We follow the hyper-parameter settings in NT/NE's papers.

5.6 Ablation Studies

We do ablation studies from three perspectives:
1.SNE: Shallow Noise Editing. we only add noises to FFNs of the layer where we add δ_i (equation 3).
2.UN: Uniform Noises. We apply Uniform noises $\epsilon \sim \mathcal{U}(-1, 1)$, the same with NT/NE, in equation 6.
3.RNP: Random Noising Position. We add noises to random tokens rather than the last subject tokens.

With SNE, we can demonstrate the effectiveness of *deep* noise. With UN and RNP, we show whether our findings, i.e. different contexts place *Gaussian-like* shifts to the FFNs' activations on *knowledge-related* tokens, can motivate the methods of adding noises that achieve the best results. Table 7 reports

Setting	Editor	Score ↑		
		$\|\mathbb{M}\|$=1e^0	$\|\mathbb{M}\|$=1e^2	$\|\mathbb{M}\|$=1e^4
GPT2-xl zsRE	MEMIT$_{\text{DNE}}$	**44.80**	**47.31**	41.57
	MEMIT$_{\text{SNE}}$	41.88	46.09	41.89
	MEMIT$_{\text{UN}}$	39.34	43.52	29.55
	MEMIT$_{\text{RNP}}$	37.88	45.88	**41.93**
GPT-J Counterfacts	MEMIT$_{\text{DNE}}$	92.47	**92.64**	**85.87**
	MEMIT$_{\text{SNE}}$	92.38	92.12	85.84
	MEMIT$_{\text{UN}}$	**92.60**	91.06	70.94
	MEMIT$_{\text{RNP}}$	92.15	91.97	85.85

Table 7: Results of the three ablation studies.

the results of the Scores of the three ablation studies and we leave the detailed results to Appendix C. DNE can achieve the highest Score (as well as the highest generalization) in the most cases. DNE will mostly improves generalization but also slightly decrease specificity. Because the Score calculates the harmonic mean, which is sensitive to smaller values, but the baselines of generalization are much larger than the specificity, DNE's can then be lower than the counterpart methods in some cases.

6 Conclusions

In this paper, we study the questions of: how LLMs can recall the same knowledge in the different contexts and how we can edit LLMs' knowledge while maintaining such important properties. For the first part, we follow the state-of-the-art editing methods and the latest interpretability works to focus on analyzing FFNs' activations. Though comparing the histogram figures and numerical results, we empirically show that different contexts can only place small shifts that follow considerably narrow Gaussian-like distributions in FFNs' activations on knowledge-related tokens. And LLMs' FFNs can produce kind of 'dominate' activations when processing knowledge. Motivated by our findings, we make to answer the second part of the questions by adding noises into FFNs' activations when editing LLMs. By doing so, we can make LLMs **see the unseen** contexts where the edited knowledge will be applied and improve the editing generalization. We run experiments on two open-sources including two standard datasets and three popular LLMs. The experimental results show the effectiveness of our methods. We make extra discussions, comparisons with other methods of adding noises, and ablation studies to comprehensively analyze how our findings can motivate the methods of adding noises that best fit with the task of knowledge-editing.

7 Limitations

Although we have run comprehensive experiments, there are some limitations. Because of the incompleteness of current open-sources and the limited computing resources, the experiments do not include editing larger LLMs or editing multiple cases and using Counterfacts on LLaMA-2. We follow exactly the same settings of all hyper-parameters to make our results replicable. Therefore, the results may not reach their best performances. Although the knowledge application is an important topic in LLMs, we narrow on this topic and do not extend our scope of applying our methods of adding noises into the general fine-tunings of LLMs.

References

Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Trans. Assoc. Comput. Linguistics*, 10:904–921.

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, Phil Wang, and Samuel Weinbach. 2023. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6491–6506. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *CoRR*, abs/2309.08600.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023a. Neftune: Noisy embeddings improve instruction finetuning. *CoRR*, abs/2310.05914.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R

Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, et al. 2023b. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint arXiv:2310.05914*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomás Kociský, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. Mind the gap: Assessing temporal generalization in neural language models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29348–29363.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 333–342. Association for Computational Linguistics.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2023. PMET: precise model editing in a transformer. *CoRR*, abs/2308.08742.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NeurIPS*.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Fabio Petroni, Patrick S. H. Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020*.

Yuval Pinter and Michael Elhadad. 2023. Emptying the ocean with a spoon: Should we edit models? In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 15164–15172. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitry V. Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Gilbert Strang. 2022. *Introduction to linear algebra*. SIAM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. *CoRR*, abs/2309.04827.

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

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, and Huajun Chen. 2023a. Easyedit: An easy-to-use knowledge editing framework for large language models. *CoRR*, abs/2308.07269.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2023b. Knowledge editing for large language models: A survey. *CoRR*, abs/2310.16218.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2022. Noisytune: A little noise can help you finetune pretrained language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 680–685. Association for Computational Linguistics.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10222–10240. Association for Computational Linguistics.

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

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 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15686–15702. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *CoRR*, abs/2012.00363.

A FFNs activations in Different Layers

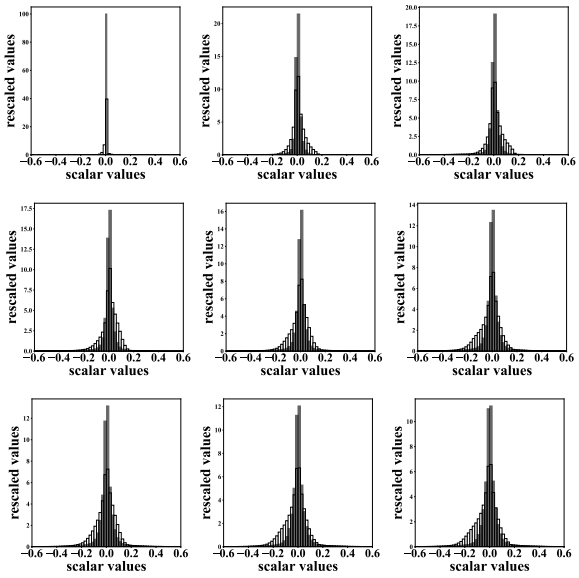

Figure 11: $\mathbb{D}_s, \mathbb{D}_c$ of GPT-J's layers from 1st to 8th.

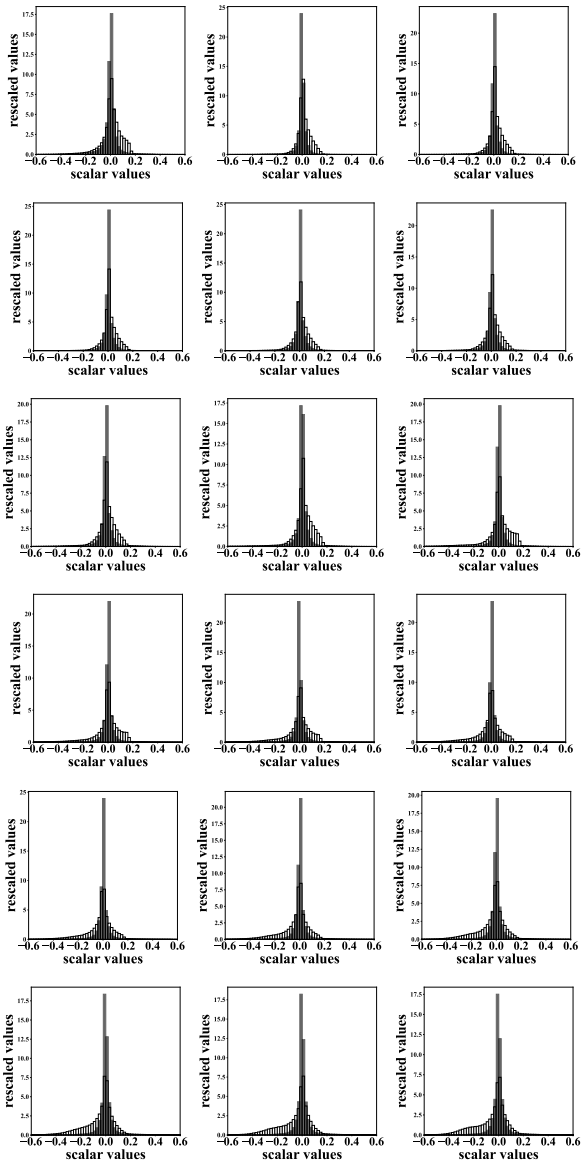

Figure 12: $\mathbb{D}_s, \mathbb{D}_c$ of GPT2-xl's layers from 1st to 18th.

Figures 12 and 11 display the histograms for all the layers of the two LLMs we select to analyze, where the black rectangles plot the experimental sets and the white ones plot the control sets and the layer index goes up horizontally and then vertically. We can see that, the black rectangles (activations of the knowledge-related tokens) remain to be concentrated abound the zeros and descend symmetrically and evenly to the both sides while the white rectangles (activations of normal-strings tokens), in all layers, show much more significant skewness.

B Additional Experimental Results

Table 9 and 8 report the results about editing GPT-J (6B) on zsRE and editing GPT2-xl on Counterfacts. These are the additional results to our main experi-

$\lvert\mathbb{M}\rvert$	Editor	Score	Efficacy	Generalization		Specificity	Fluency	Consistency
		S ↑	ES ↑	PS ↑	PA ↑	NS ↑	GE ↑	RS ↑
$1e^0$	ROME	89.80	**99.94 (0.0)**	97.08 (0.2)	74.04 (0.5)	**76.34 (0.4)**	**622.18 (0.3)**	42.00 (0.2)
	ROME_{DNE}	**89.81**	99.93 (0.0)	**98.30 (0.1)**	**77.93 (0.5)**	75.63 (0.4)	620.16 (0.4)	**42.40 (0.2)**
	MEMIT	83.40	94.22 (0.3)	79.95 (0.5)	41.02 (0.6)	77.82 (0.4)	**627.33 (0.2)**	39.44 (0.2)
	PMET	58.59	61.78 (0.7)	44.88 (0.6)	8.41 (0.3)	**78.50 (0.4)**	627.18 (0.2)	34.16 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**88.10**	**98.16 (0.2)**	**92.23 (0.3)**	**58.78 (0.6)**	76.80 (0.4)	617.02 (0.4)	**41.41 (0.2)**
$1e^1$	MEMIT	83.54	94.33 (0.3)	80.25 (0.5)	41.53 (0.6)	77.84 (0.4)	**627.34 (0.2)**	39.47 (0.2)
	PMET	58.78	62.06 (0.7)	45.07 (0.6)	8.55 (0.3)	**78.52 (0.4)**	627.21 (0.2)	34.17 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**87.79**	**97.91 (0.2)**	**90.94 (0.3)**	**57.43 (0.6)**	77.15 (0.4)	622.58 (0.3)	**41.62 (0.2)**
$1e^2$	MEMIT	84.26	94.72 (0.3)	81.77 (0.5)	44.41 (0.6)	78.02 (0.4)	627.22 (0.2)	39.73 (0.2)
	PMET	60.20	63.86 (0.7)	46.56 (0.6)	9.66 (0.3)	**78.75 (0.4)**	**627.41 (0.2)**	34.37 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**86.67**	**96.79 (0.2)**	**88.90 (0.4)**	**56.08 (0.6)**	76.72 (0.4)	625.26 (0.2)	**41.29 (0.2)**
$1e^3$	MEMIT	82.30	93.03 (0.4)	80.27 (0.5)	43.23 (0.6)	75.49 (0.4)	626.50 (0.2)	39.25 (0.2)
	PMET	61.73	65.32 (0.7)	48.60 (0.6)	11.20 (0.4)	**78.65 (0.4)**	**627.82 (0.2)**	34.65 (0.2)
	$\text{MEMIT}_{\text{DNE}}$	**82.61**	**93.62 (0.3)**	**82.90 (0.5)**	**47.49 (0.6)**	73.68 (0.4)	626.34 (0.2)	**39.76 (0.2)**
$1e^4$	MEMIT	71.73	80.09 (0.5)	66.19 (0.6)	25.32 (0.5)	70.28 (0.4)	625.91 (0.2)	36.43 (0.2)
	PMET	50.82	49.77 (0.7)	38.58 (0.6)	5.61 (0.3)	**76.82 (0.4)**	**627.87 (0.2)**	33.38 (0.1)
	$\text{MEMIT}_{\text{DNE}}$	**72.28**	**80.91 (0.5)**	**67.66 (0.6)**	**26.32 (0.5)**	69.60 (0.4)	626.14 (0.2)	**36.67 (0.2)**

Table 8: Additional results of editing GPT2-xl with Counterfacts from $1e^0$ to $1e^4$. Within parentheses is the 95% confidence interval.

$\lvert\mathbb{M}\rvert$	Editor	S. ↑	Effi. ↑	Para.↑	Spec. ↑
$1e^0$	ROME	52.44	**99.88 (0.0)**	95.27 (0.2)	**27.25 (0.4)**
	ROME_{DNE}	**52.58**	99.87 (0.0)	**96.66 (0.2)**	**27.25 (0.4)**
	MEMIT	51.60	**99.84 (0.0)**	87.65 (0.4)	**27.24 (0.4)**
	PMET	51.57	98.43 (0.1)	88.45 (0.4)	**27.24 (0.4)**
	$\text{MEMIT}_{\text{DNE}}$	**52.59**	99.10 (0.1)	**97.85 (0.2)**	27.22 (0.4)
$1e^1$	MEMIT	52.20	**99.70 (0.1)**	92.99 (0.3)	**27.26 (0.4)**
	PMET	51.86	97.22 (0.2)	92.17 (0.3)	27.24 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	**52.69**	99.26 (0.1)	**98.37 (0.1)**	27.24 (0.4)
$1e^2$	MEMIT	52.63	**99.56 (0.1)**	93.45 (0.3)	**27.58 (0.4)**
	PMET	52.22	96.82 (0.2)	92.21 (0.3)	27.57 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	**52.66**	98.97 (0.1)	**96.94 (0.2)**	27.36 (0.4)
$1e^3$	MEMIT	**53.37**	**98.81 (0.1)**	93.38 (0.3)	**28.26 (0.4)**
	PMET	52.72	95.46 (0.2)	90.73 (0.3)	28.24 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	52.23	98.49 (0.1)	**94.19 (0.3)**	27.27 (0.4)
$1e^4$	MEMIT	**51.01**	96.35 (0.2)	89.95 (0.4)	**26.80 (0.4)**
	PMET	49.42	90.47 (0.3)	84.36 (0.4)	26.46 (0.4)
	$\text{MEMIT}_{\text{DNE}}$	50.52	**96.45 (0.2)**	**90.01 (0.4)**	26.38 (0.4)

Table 9: Additional results of editing GPT-J on zsRE from $1e^0$ to $1e^4$ edits. Within the parentheses is the 95% confidence interval.

$\lvert\mathbb{M}\rvert$	Editor	S. ↑	Effi. ↑	Para.↑	Spec. ↑
$1e^0$	$\text{MEMIT}_{\text{DNE}}$	**44.48**	**80.31 (0.4)**	**72.17 (0.5)**	24.31 (0.4)
	MEMIT_{NT}	39.30	65.83 (0.5)	49.91 (0.5)	**24.33 (0.4)**
	MEMIT_{NE}	37.07	58.15 (0.5)	44.19 (0.5)	**24.33 (0.4)**
	$\text{MEMIT}_{\text{SNE}}$	41.88	74.43 (0.5)	58.48 (0.5)	**24.33 (0.4)**
	MEMIT_{UN}	39.34	62.80 (0.5)	51.98 (0.5)	**24.33 (0.4)**
	$\text{MEMIT}_{\text{RNP}}$	37.88	57.28 (0.5)	48.44 (0.5)	**24.33 (0.4)**
$1e^2$	$\text{MEMIT}_{\text{DNE}}$	**47.31**	**89.31 (0.3)**	**84.30 (0.4)**	**24.78 (0.4)**
	MEMIT_{NT}	45.34	82.88 (0.4)	74.02 (0.5)	24.64 (0.4)
	MEMIT_{NE}	44.52	79.69 (0.5)	70.65 (0.5)	24.58 (0.4)
	$\text{MEMIT}_{\text{SNE}}$	46.09	86.02 (0.4)	77.63 (0.5)	24.64 (0.4)
	MEMIT_{UN}	43.52	74.60 (0.5)	68.10 (0.5)	24.48 (0.4)
	$\text{MEMIT}_{\text{RNP}}$	45.88	84.39 (0.3)	77.59 (0.5)	24.60 (0.4)
$1e^4$	$\text{MEMIT}_{\text{DNE}}$	41.57	63.47 (0.5)	58.59 (0.6)	25.42 (0.4)
	MEMIT_{NT}	41.58	62.62 (0.5)	57.95 (0.6)	25.69 (0.5)
	MEMIT_{NE}	41.74	63.02 (0.5)	57.94 (0.6)	**25.81 (0.4)**
	$\text{MEMIT}_{\text{SNE}}$	41.89	63.26 (0.5)	**58.68 (0.6)**	25.80 (0.4)
	MEMIT_{UN}	29.55	35.92 (0.5)	32.36 (0.5)	23.38 (0.3)
	$\text{MEMIT}_{\text{RNP}}$	**41.93**	62.99 (0.5)	58.51 (0.6)	25.92 (0.4)

Table 10: Detailed experimental results of editing GPT2-xl on zsRE. Within the parentheses is the 95% confidence interval.

ments in Section 5.2 and 5.3. In all cases, editing with DNE returns higher generalization and in the most cases return the highest Score. As we have pointed out in our main experiments' discussions, although DNE makes the 'Fluency' lower, this metrics only considers the diversity of the generation texts and can not well reflect the classical text fluency in our common sense. And as the RS remains rather high, DNE will not cause the generation degeneration to stupidly repeat nonsense words.

C Detailed Experimental Results

Table 10 and 11, in integrate, report the detailed results of: Section 5.5 comparing with other methods of adding noises (NT and NE), and Section 5.6: the three ablation studies (SNE, UN, and RNP). In the most cases, DNE achieves the highest scores and the best generalization. While in some cases, DNE can be beaten by other methods, DNE still achieve the most robust performance gains on two models in all the cases. For example, in Table 11, MEMIT_{UN} gets higher Score in editing $1e^0$ case but its performance becomes dramatically degener-

$\|\mathbb{M}\|$	Editor	Score	Efficacy	Generalization		Specificity	Fluency	Consistency
		S ↑	ES ↑	PS ↑	PA ↑	NS ↑	GE ↑	RS ↑
$1e^0$	$\text{MEMIT}_{\text{DNE}}$	92.47	99.75 (0.1)	**99.08 (0.1)**	**87.40 (0.4)**	81.14 (0.4)	614.80 (0.4)	**42.40 (0.2)**
	MEMIT_{NT}	91.70	**99.86 (0.1)**	95.19 (0.3)	67.28 (0.6)	82.00 (0.4)	621.95 (0.2)	41.63 (0.2)
	MEMIT_{NE}	91.25	99.76 (0.1)	93.14 (0.3)	63.53 (0.5)	**82.53 (0.3)**	**622.09 (0.2)**	41.69 (0.2)
	$\text{MEMIT}_{\text{SNE}}$	92.38	**99.86 (0.1)**	97.83 (0.2)	77.24 (0.5)	81.71 (0.4)	620.84 (0.2)	42.73 (0.2)
	MEMIT_{UN}	**92.60**	99.77 (0.1)	98.37 (0.2)	81.85 (0.4)	81.90 (0.4)	621.11 (0.2)	42.71 (0.2)
	$\text{MEMIT}_{\text{RNP}}$	92.15	99.71 (0.1)	96.24 (0.2)	80.75 (0.4)	82.41 (0.3)	616.84 (0.4)	41.37 (0.2)
$1e^2$	$\text{MEMIT}_{\text{DNE}}$	**92.64**	99.79 (0.1)	**97.96 (0.2)**	**81.21 (0.4)**	82.27 (0.4)	620.35 (0.2)	**42.70 (0.2)**
	MEMIT_{NT}	91.70	99.85 (0.1)	94.89 (0.3)	66.61 (0.5)	82.23 (0.3)	622.02 (0.2)	41.65 (0.2)
	MEMIT_{NE}	91.31	99.76 (0.1)	92.90 (0.3)	63.37 (0.5)	**82.88 (0.3)**	621.75 (0.2)	41.55 (0.2)
	$\text{MEMIT}_{\text{SNE}}$	92.12	**99.87 (0.1)**	96.46 (0.2)	71.87 (0.5)	82.05 (0.3)	621.61 (0.2)	42.15 (0.2)
	MEMIT_{UN}	91.06	99.12 (0.1)	94.00 (0.3)	70.44 (0.5)	81.84 (0.3)	**622.67 (0.2)**	41.31 (0.2)
	$\text{MEMIT}_{\text{RNP}}$	91.97	99.81 (0.1)	95.48 (0.2)	72.95 (0.5)	82.46 (0.3)	621.45 (0.2)	41.83 (0.2)
$1e^4$	$\text{MEMIT}_{\text{DNE}}$	**85.87**	**99.26 (0.1)**	**89.82 (0.4)**	**58.43 (0.6)**	72.83 (0.4)	618.10 (0.2)	**40.33 (0.2)**
	MEMIT_{NT}	85.83	99.09 (0.1)	88.42 (0.4)	55.68 (0.6)	**73.79 (0.4)**	619.47 (0.2)	40.13 (0.2)
	MEMIT_{NE}	82.29	96.54 (0.3)	80.70 (0.4)	36.46 (0.5)	72.97 (0.4)	572.75 (0.3)	36.97 (0.2)
	$\text{MEMIT}_{\text{SNE}}$	85.84	99.14 (0.1)	88.82 (0.4)	56.49 (0.6)	73.51 (0.4)	**619.59 (0.2)**	40.32 (0.2)
	MEMIT_{UN}	70.94	79.53 (0.6)	68.52 (0.5)	20.30 (0.5)	66.13 (0.4)	527.55 (0.5)	24.85 (0.2)
	$\text{MEMIT}_{\text{RNP}}$	85.85	99.12 (0.1)	88.56 (0.4)	56.32 (0.6)	73.73 (0.4)	618.90 (0.2)	40.05 (0.2)

Table 11: Detailed experimental results on GPT-J (6B). Within parentheses is the 95% confidence interval.

ated when editing $1e^4$ cases. And also in Table 10, the results of NT and NE largely falls behind on editing $1e^0$ case. And NE shows significant lower generation qualities when applied on GPT-J in editing $1e^4$ cases in Table 11, because its generation fluency GE and consistency RS both get lower.

D Details about the Experiments

D.1 Datasets Details

In knowledge-editing, each data has one editing context for updating (training) LLMs and several applied contexts for evaluating the editing metrics, including contexts for the edited knowledge that evaluates effectiveness and generalization and contexts for an pre-selected arbitrary unrelated knowledge that evaluates specificity. For the two datasets we use, zsRE contains 19,087 data and Counterfacts contains 20,878 data.

D.2 Computing Resources

We run experiments on LLMs with three sizes, including GPT2-xl (1.5B), GPT-J (6B), and LLaMA-2 (7B). And we use one NVIDIA A800 80GB GPU to run our all experiments. The running time varies from several hours to 3 days.

D.3 Settings to the Hyper-parameters

All our experiments are based on the two open-sources in MEMIT and EasyEdit. The two open-sources provide pre-selected hyper-parameters and we all follow their settings. We have one hyper-parameter that the two open-source do not contain, i.e. the α. And we have written the chosen values and provided ablation studies on its values choices.

D.4 Results Statistics

One iteration of the datasets contains serveral editing experiments. For example, if the $|\mathbb{M}|$ is 1e0 on zsRE, one iteration of the datasets can have 19,087 editing experiments. We set the seeds for all experiments when starting the iteration, only iterate the datasets once, and report the results. The reported metrics contains the mean and the 95% confidence interval. As for example, the reported value 60.00 (0.15) denotes the result has a mean of 60.00 and the 95% confidence interval is (59.85, 60.15).

D.5 Existing Packages

We use the transformers' evaluation packages to calculate the BLEU and ROUGE. And we follow the default parameter and model settings.