Beyond Single Edits: Rethinking Knowledge Editing Evaluation with Multi-Edit Entities and Undirected Probing

Anonymous ACL submission

Abstract

Large Language Models (LLMs) acquire extensive factual knowledge during computationally expensive pretraining. Knowledge editing aims to efficiently update specific facts in a trained model without retraining the entire network. While many editing methods have recently been proposed and show strong performance on existing benchmarks, these benchmarks exhibit several limitations. In particular, they allow only a single edit per entity, rely on highly templated prompts, and evaluate edits using relative probability changes rather than ground-truth correctness. As a result, the capabilities and limitations of current knowledge editing methods remain insufficiently understood. In this work, we introduce the Subject-centric Entity Editing Dataset (SEED), a dataset and evaluation framework designed to address these limitations. SEED organizes edits into sets of related and unrelated facts rather than isolated instances, enabling evaluation when multiple facts about the same subject entity are updated. It also introduces undirected knowledge probes that test whether edited facts can be recalled in general contexts, and ground-truth evaluation metrics that directly measure whether the edited fact appears in the model's output. Using SEED, we test 9 state-of-the-art knowledge editing methods over 10 metrics. These studies find that editing performance decreases by up to 36.9% as the number of edits per entity increases and edit recall decreases by 36.6% with prompts that increase token-wise distance between the edit subject and relation.

1 Introduction

Large Language Models (LLMs) have been widely adopted in commercial and interactive applications (Thirunavukarasu et al., 2023; Kim et al., 2025; Liang et al., 2025). However, factual knowledge learned during training (Petroni et al., 2019; Roberts et al., 2020; Chang et al., 2024) can become outdated as real-world information changes over time. It is therefore important that LLMs have access to up-to-date information when responding to user queries. Knowledge editing methods address this challenge by making targeted updates to model parameters, avoiding the computational (Li et al., 2025a) and financial (Maslej et al., 2025) cost of retraining.

A broad range of knowledge editing approaches have recently been proposed, including parameter-based editors (Mitchell et al., 2022; Hu et al., 2024; Li et al., 2024; Jiang et al., 2025), which modify model weights to apply edits, and memory-based editors (Gu et al., 2024; Wang et al., 2024; Simon and Ewetz, 2025), which store edits in an external database for retrieval at inference time. These methods report excellent performance on benchmark datasets such as CounterFact (Meng et al., 2022), MQuAKE (Zhong et al., 2023), and UnKEBench (Deng et al., 2025). However, these datasets exhibit several limitations in their edit structure, prompting strategies, and evaluation metrics. Edits are treated as isolated instances, making it difficult to evaluate whether editors can successfully apply multiple edits to the same subject entity. Evaluation prompts are typically paraphrases of the edited facts, limiting the contexts in which edit recall is tested. Existing metrics rely on relative probability changes or text similarity, rather than directly measuring whether the edited fact appears in the model's output.

To address these limitations, we introduce the **S**ubject-centric **E**ntity **E**diting **D**ataset (SEED), a benchmark for evaluating Knowledge Editing methods. Our main contributions are as follows:

- Unlike prior benchmarks that focus on one-off edits, SEED supports multiple edits associated with the same subject, allowing evaluation of editing methods under multi-edit settings.
- We introduce a novel probing strategy that tests whether edited facts can be recalled

Category	Edit Structure		Probing Strategy			Evaluation Metrics		
Features	1-to-1 Edits	1-to-n Edits	Directed Probing	Undirected Probing	Compound Probing	Next Token Pred.	Machine Translation	String Matching
ZSRE	✓	∅	✓	∅	∅	✓	∅	∅
CounterFact	✓	∼	✓	∅	∅	✓	✓	∅
MQuAKE	✓	∼	✓	∅	∅	✓	∅	∅
KEBench	✓	∅	✓	∅	∅	✓	∅	∅
RippleEdit	✓	∅	✓	∅	∅	✓	∅	∅
UnKEBench	✓	∅	✓	∅	∅	∅	✓	∼
EveryEdit	✓	∅	✓	∅	∅	∅	✓	∅
SEED (ours)	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Key features included in Knowledge Editing datasets. ✓indicates feature inclusion, ∼ indicates similar features, and ∅ indicates feature absence.

in a different format than the inserted edits, measuring generalization to broader conversational contexts.

- SEED provides metrics that directly measure edited fact presence in model outputs, eliminating reliance on similarity-based metrics.
- Experiments with SEED show that editing performance drops by up to 36.9% as the number of edits per subject increases, and that undirected probes reduce edit recall by 36.6% compared to standard paraphrase prompts, indicating issues with subject-level representations and a lack of knowledge integration robustness.

2 Background

The Knowledge Editing problem formulation is described in Section 2.1. Related datasets are discussed in Section 2.2.

2.1 Problem Formulation

Knowledge can be concisely represented as relational triples $t = (s, r, o)$, a 3-tuple containing a subject s and object o entity that are connected by some relation r. For example, *Tokyo is the capital of Japan* is represented as $t = (\textit{Tokyo}, \textit{capital}, \textit{Japan})$. Editing is then framed as updating t with a new object o^*, such that $t^* = (s, r, o^*)$. This allows questions like *What is the capital of France* to be asked where the expected answer is *Tokyo*.

2.2 Limitations of Existing Datasets

In this section, we compare existing datasets based on their *edit structure*, *probing strategy*, and *evaluation metrics*, as shown in Table 1.

All existing datasets provide 1-to-1 edits, where each subject is assigned a single edit. CounterFact (Meng et al., 2022) and MQuAKE (Zhong et al., 2023) are the only datasets to contain 1-to-n edits, where a single subject is assigned multiple edits. However, only 0.2% and 1.8% of subjects in the respective datasets receive 3 or more edits, demonstrating a lack of systematic evaluations over subjects that received multiple edits.

All existing datasets provide *directed* evaluation prompts based on paraphrases of edits that closely group the edit subject and relation. ZSRE (Levy et al., 2017) formats edits as single-hop questions, KEBench (Wei et al., 2024) chains edits together to form multi-hop questions, and EveryEdit (Jiang et al., 2025) uses the exact edit as the evaluation prompt. No existing datasets provide any *undirected* prompts to assess a model's ability to recall edits when the subject and relation contain many intervening tokens between them. Further, no datasets evaluate whether edited models can recall edited knowledge under the *compounding* effects of multiple edits at the input.

The evaluation metrics used are broadly split by the format that datasets provide edits in. Datasets that format edits as relational triples tend to use next token prediction metrics like Efficacy and Neighborhood Score (Meng et al., 2022), while datasets that format edits as unstructured text use machine translation metrics like BLEU (Papineni et al., 2002) and BERTScore (Zhang et al., 2020). Current datasets neglect any form of *string matching* to measure the direct inclusion of edits in generated text. UnKEBench (Deng et al., 2025) attempts edit inclusion quantification, but relies on generat-

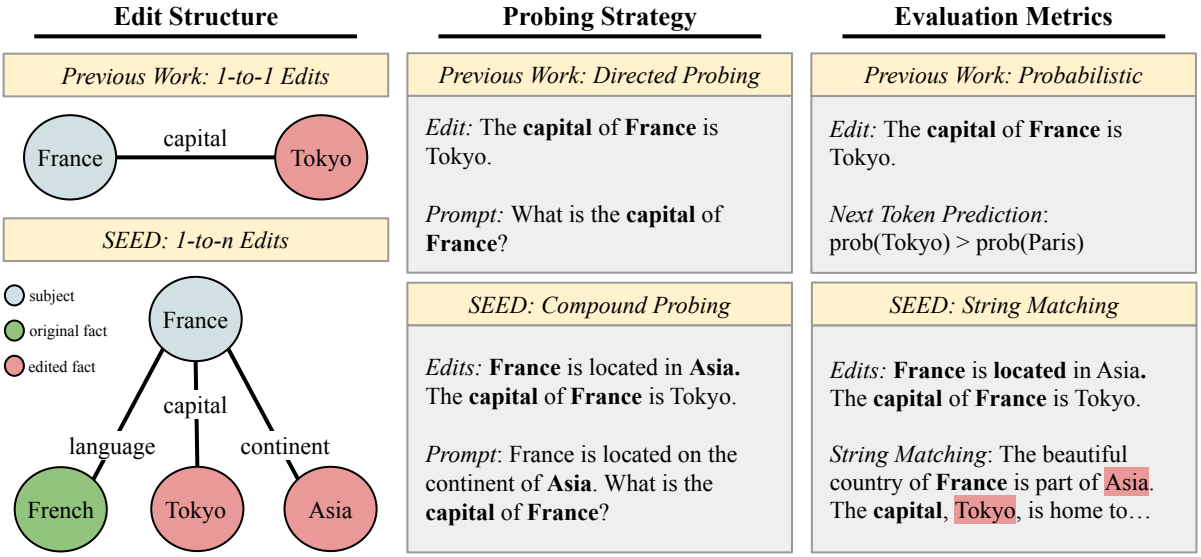

Figure 1: **Edit Structure**: 1-to-n edits enable complex evaluations of editing performance due to containing multiple edited and non-edited relations. **Probing Strategy**: Undirected probing separates the edited subject-relation pair, breaking patterns learned through editing. **Evaluation Metrics**: Evaluation over compound edits in the same input and direct edit matching in generated text provide insights into editing performance under varied conditions.

ing atomic subquestions and abstracting measurements to a language model.

Our proposed SEED dataset overcomes these limitations by supporting multiple edits per subjects, adopting prompting strategies that do not directly reference the edited fact, and incorporates metrics that directly measure whether edits appear in model outputs.

3 Dataset Innovations

In this section, we describe how the innovations in SEED lead to new challenges with regard to the 3 main components of Knowledge Editing datasets, illustrated in Figure 1. Innovations to the structure of edits is discussed in Section 3.1, the prompting strategies used for evaluation in Section 3.2, and the evaluation metrics in Section 3.3.

3.1 Edit Structure

The 1-to-n setting poses a fundamental challenge for current editors. Both parameter- and memory-based editors rely on subject entities as lookup keys (Geva et al., 2021), creating ambiguity when multiple edits target the same subject. Consider editing both the *capital* and *president* of *France*. A memory-based editor must distinguish between the sentence embeddings of (*France*, *capital*) and (*France*, *president*) using similarity metrics, which becomes increasingly difficult when subjects are identical and relations are semantically similar. As the number of edits per subject grows, it becomes increasingly likely to infer related knowledge as edits. For example, the similarity between *The president of France is* and *The prime minister of France is* is 0.802, where the threshold for an edit is commonly in the 0.7 to 0.8 range.

3.2 Probing Strategy

Current probing strategies provide edit subjects and relations close together. For example, in *The capital of France is*, the subject and relation are separated by 1 word. This creates an identifiable subject-relation pattern for edit retrieval, allowing a model to easily answer with *Tokyo*. In contrast, the prompt *Tell me about France* only contains the edit subject. The model can then visit other relations before discussing the *capital*, leading to the subject-relation pattern being broken. The model will now have difficulty retrieving the answer *Tokyo*.

3.3 Evaluation Metrics

Current evaluations rely on measuring output token probability shifts over a single edit, but fail to assess whether the distribution shifts hold over sequential edits in the input. For example, it is unclear how the next token probabilities of *France is located in Asia. The capital of France is* will behave because the edit t = (*France*, *continent*, *Asia*) appears before another edit to the same subject is queried. Further, no

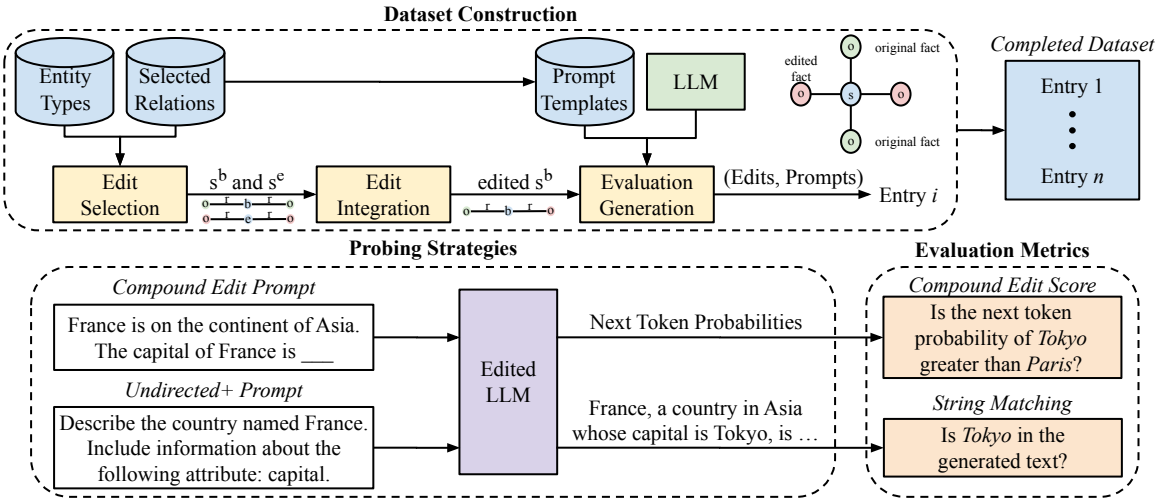

Figure 2: **Dataset Construction**: Dataset entries are created by applying objects of one subject as the edits to another. **Probing Strategies**: Directed and Undirected prompts are passed to the edited LLM to generate next token probabilities and full text outputs. **Evaluation Metrics**: The new probing strategies are introduced to evaluate model performance with edits in the input and to evaluate direct inclusion of edits in generated text.

metrics provide evaluations of whether the edits are directly present in text generations. An editing method that successfully shifts token probabilities but fails to incorporate edited facts into text generation has limited utility for downstream applications.

4 SEED

SEED is composed of edits, prompts, and metrics. The dataset construction pipeline is explained in Section 4.1. The proposed probing strategies are presented in Section 4.2. The proposed evaluation metrics are described in Section 4.3. An example of the full SEED pipeline is provided in Figure 2.

4.1 Dataset Construction Pipeline

Each entry in SEED is formatted as $(\mathcal{T}, \mathcal{P})$, where $\mathcal{T}$ is a 1-to-n edit scenario and $\mathcal{P}$ contains evaluation questions. Construction of a single entry involves three steps: *Entity Selection* extracts two entities, a base subject s^b and edit subject s^e, and their relations from Wikidata, discussed in Section 4.1.1. *Edit Integration* transfers objects from overlapping s^e relations to corresponding s^b relations, creating edited triples $t^* = (s^b, r, o^e)$, discussed in Section 4.1.2. *Evaluation Generation* produces evaluation prompts, discussed in Section 4.1.3. Dataset statistics are given in Section 4.1.4.

4.1.1 Entity Selection

The input to *Entity Selection* is Wikidata (Vrandečić and Krötzsch, 2014), a set of entity types, and a set of relations. The output of this step is then the two pruned subjects.

We manually curate of a set of entity types $\mathcal{E}$, based on Wikidata $\mathcal{W}$ entity types, for extracting pairs of similar subjects. The types $\mathcal{E}$ are used to filter $\mathcal{W}$ to $\mathcal{W}_\mathcal{E} \subseteq \mathcal{W}$, which contains only subjects with types in $\mathcal{E}$. The base subject s^b and edit subject s^e are sampled from $\mathcal{W}_\mathcal{E}$.

We manually curate a set of relations $\mathcal{R}_\mathcal{E}$ from the top 25% of relations in 5000 entities from $\mathcal{W}_\mathcal{E}$, ensuring LLMs have pre-edit knowledge of all selected relations. For both the base subject relations $\mathcal{R}_b$ and edit subject relations $\mathcal{R}_e$, we only retain relations in $\mathcal{R}_\mathcal{E}$ and discard the rest, such that $\mathcal{R}_{b,\text{pruned}} = \mathcal{R}_b \cap \mathcal{R}_\mathcal{E}$ and $\mathcal{R}_{e,\text{pruned}} = \mathcal{R}_e \cap \mathcal{R}_\mathcal{E}$. Subjects with $|\mathcal{R}_s| < 2$ are replaced since subjects with only a single relation cannot be used to evaluate 1-to-n edits. When relations map to multiple objects, we randomly select one as the primary object and store the others as aliases. The output is then the pruned s^b and s^e, with knowledge stored in tuples as $t = (s, r, o, o_{\text{alias}})$.

4.1.2 Edit Integration

The input to *Edit Integration* is s^b and s^e. We find the intersection of relations between $\mathcal{R}_b$ and $\mathcal{R}_e$, such that $\mathcal{R}_{\text{inter}} = \mathcal{R}_b \cap \mathcal{R}_e$. If $|\mathcal{R}_{\text{inter}}| < 2$, the edit subject s^e is replaced with another from $\mathcal{W}_\mathcal{E}$. For each $r \in \mathcal{R}_{\text{inter}}$, the objects o^e of corresponding edit subject relations $\mathcal{R}_e$ become edits for corresponding base subject relations $\mathcal{R}_b$, such that edited triples within s^b are $t^* = (s^b, r, o^e)$.

Subset	Art	Business	Person
# Subject	1000	1000	1000
# Relations	3855	3696	11267
# Edits	2733	2527	5594
Avg # Edits	2.7	2.5	5.6

Table 2: Dataset statistics for the SEED subsets.

Each knowledge tuple within s^b becomes $t = (s, r, o, o_{\text{alias}}, o^*, o^*_{\text{alias}})$, where $o^* = \textit{null}$ for non-edited relations.

4.1.3 Evaluation Generation

The input to *Evaluation Generation* is the base subject s^b after receiving edits. We manually construct a set of cloze-style $\mathcal{T}_c$ and single-hop question $\mathcal{T}_q$ templates per selected relation $r \in \mathcal{R}_\mathcal{E}$ and ensure that $|\mathcal{T}_c| \geq 4$ and $|\mathcal{T}_q| \geq 1$. For each triple in the base subject s^b, one sentence and questions template are randomly selected to represent the knowledge. For edited triples, this sentence is used by editing frameworks to apply edits. Also, three templates are randomly selected to become paraphrase prompts. A model text generation a subject not in the dataset is prepended to each paraphrase prompt to test the generalization of inserted edits. For each subject, we also include a Compound Probing Prompt, described in Section 4.1.3.

4.1.4 Dataset Statistics

We generate 3 unique datasets grouped by Art, Business, and People, chosen for their prevalence in Wikidata and coverage of a variety of relations. The dataset statistics are provided in Table 2. Further information on dataset construction and statistics are provided in Section C.

4.2 Probing Strategies

We introduce *undirected* probing strategies to probe model recall of edited information that separates subjects and relations, preventing models from relying on learned subject-relation patterns. Specifically, we propose the *Undirected* and *Undirected+* prompting strategies. *Undirected* prompts only ask for information about the base subject. The entity type is also given to help with subject disambiguation. *Undirected+* prompts additionally specify which relations to include in a separate sentence to provide additional input guidance while still keeping subject-relation separation. The prompts used in this work are given in Appendix D.

We also introduce a new directed probing strategy, the Compound Probing Prompt (CPP), to evaluate edit recall when edits exist in the model input. An LLM is used to generate a sentence for each triple in s^b. To ensure consistency, we only prompt for the creation of 2 sentences based on 2 triples at a time, combining all sentences at the end. If an object from a triple does not appear in a pair of sentences, we instead sample relevant sentences from the template sentences. The prompt can be succinctly represented as CPP $= \bigoplus_i^n(t)$ for each $t \in s^b$ when t is a sentence and $\bigoplus$ represents string concatenation.

4.3 Metrics

SEED uses Efficacy Score (ES) and Generality Score (GS) (Meng et al., 2022) for next-token prediction, the machine translation metrics BLEU (Papineni et al., 2002), chrF (Popović, 2015), and BERTScore (Zhang et al., 2020) for text generation, and per-question (Q-Acc) and per-case (C-acc) (Zhong et al., 2023) for multi-hop question-answering. SEED also introduces a modification to 1 metric and presents 2 new metrics.

Modified Neighborhood Score Neighborhood Score (NS) (Meng et al., 2022) measures edit locality on non-edited knowledge. The original NS compares the output token probability $\mathbb{P}(\cdot)$ of 1 true object o against that of 1 edit object o^*, such that NS $= \mathbb{P}(o) > \mathbb{P}(o^*)$. However, subjects in SEED are composed of both edited and non-edited triples, and the non-edited triples are treated as neighborhood knowledge rather than providing separate neighborhood prompts. To measure NS in this context, each neighborhood object token o^n must be compare against each edit object o^*_i token. For a single o^n, the new NS* can be represented as:

$$\text{NS}^* = \mathbb{P}(o^n) > \max_i^n(\mathbb{P}(o^*_i)), \qquad (1)$$

where $\max_i^n(\cdot)$ indicates the maximum value over n edit object o^*_i.

Compound Edit Score The Compound Probing Score (CPS) is based on the Compound Probing Prompt (CEP), described in Section 4.1.3. The prompt is string split on each o to create cloze-style sentences $\mathcal{C} = \{c_0, c_1, ..., c_n\}$ for each subject-relation pair. Starting with $c_{\text{input}} = \{c_0\}$, we evaluate using Efficacy Score, prepend c_0 to c_1 such that $c_{\text{input}} = \{c_0, c_1\}$ and repeat until all sentences in

the prompt are used. For a single prompt, CPS is formulated as:

$$CPS = \frac{1}{n}\sum_{i=0}^{n}[\mathbb{P}(o_i|c_{\text{input}}) < \mathbb{P}(o_i^*|c_{\text{input}})], \quad (2)$$

where n is the number of triples in s^b. We also track the number of perfectly answered Compound Probing Prompts with the Compound Probing Perfect Score, such that $CPPS = \sum_i^m (CPS == 1)$, where m is the total number of edited base subjects s^b in the dataset.

String Matching String Matching (SM) measures whether edits are directly present in text generated by an edited model. An evaluation prompt is passed to a pre-edit model $f_{\text{pre}}(\cdot)$ to generate text $f_{\text{pre}}(p)$. Outputs that directly contain the true object o or its aliases o_{alias} are kept in a set $\mathcal{H}$ while the rest are discarded. The post-edit model $f_{\text{post}}(\cdot)$ is then passed each $h \in \mathcal{H}$. The post-edit model outputs are checked for the corresponding edit object o^* and its aliases o^*_{alias}. SM for a single prompt is defined as:

$$SM = \frac{o^* \in f_{\text{post}}(p) \vee o^*_{\text{alias}} \in f_{\text{post}}(p)}{|\mathcal{H}|}, \quad (3)$$

where $\in$ represents substring occurrence.

5 Experimental Results

Editing Baselines We evaluate 5 parameter- and 3 memory-based editors on SEED. Specifically, we use the GRACE (Hartvigsen et al., 2023), ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), PMET (Li et al., 2024), and SWEA (Li et al., 2025b) parameter-based editors and the PokeMQA (Gu et al., 2024), DeepEdit (Wang et al., 2024), and CHECK (Simon and Ewetz, 2025) memory-based editors. We also provide evaluations using layer-wise fine-tuning (FT), a commonly used as a baseline editing method. Parameters used by each framework are given in Appendix Section E.

Models We provide evaluations using the GPT-J (Wang and Komatsuzaki, 2021) and Llama 3.1 8B (Grattafiori et al., 2024) causal language models. GPT-J is commonly used in prior knowledge editing work, making it optimal for direct comparison against previous works. Llama 3.1 is chosen since it is a newer and larger model.

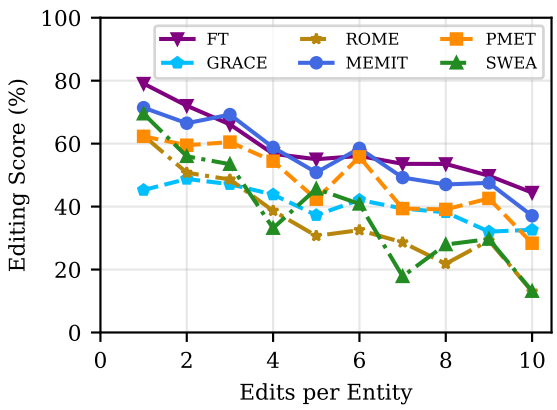

Figure 3: Editing Score (S) achieved by the parameter-based frameworks over 200 edits as the number of edits per subject entity increases.

Extended Results This results section is focused on editing GPT-J with the SEED Person dataset. Studies with different models and SEED datasets are provided in Appendix Section F.

5.1 Scaling Edits per Entity

To analyze how editing performance scales with the number of edits applied to a single subject, we evaluate models under increasing edits applied to unique subjects, shown in Figure 3. We vary the number of edits applied to a single subject from 1 to 10 while keeping the total number of edits constant at 200. The subjects are randomly sampled from the SEED Person dataset. The model used is GPT-J. Editing Score (S) is the harmonic mean over ES, GS, and NS, used to evaluate the edit success and edit locality tradeoff.

We observe consistent degradation in edit score as the number of edits per subject entity increases, suggesting that existing editing methods struggle to maintain multiple edits to the same subject. Between 1 and 3 edits per subject, the editing score decreases by an average of 7.5%. By 10 edits per subject, the editing score drops by an average of 36.9%. This effect highlights the importance of evaluating editing methods under multi-edit conditions rather than isolated edits.

5.2 Undirected Probing

We compare directed and undirected probing methods by generating text from the model using the edit sentence and the Undirected+ prompt with 1 included relation as inputs, respectively, as shown in Figure 4. This comparison is performed on SEED Person with edits applied to GPT-J.

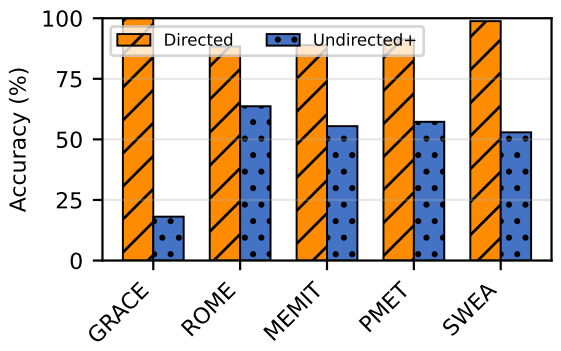

Figure 4: Comparison of frameworks using Directed and Undirected+ prompts.

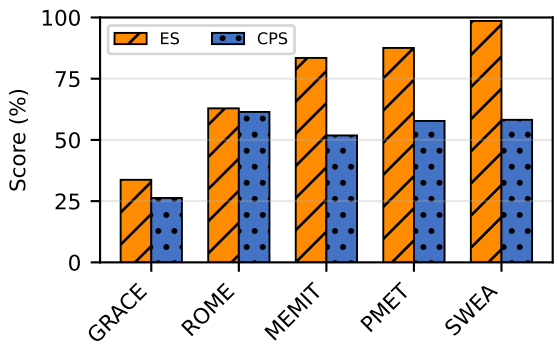

Figure 5: Comparison of framework performance on Efficacy Score (ES) and Compound Probing Score (CPS).

We first evaluate pre-edit model performance and observe that the model achieves a higher string matching accuracy on the Undirected+ prompts, correctly answering 46.8% of Undirected+ prompts versus 34.0% of directed prompts. This gap can be attributed to the structural differences between the two prompt types. Directed prompts require immediate correct responses because the model quickly moves on to discussing other topics. In contrast, the Undirected+ prompts allow the model more flexibility on when the target relation is discussed, which likely accounts for the better performance.

The results for the editing frameworks use the pre-edit results as an upper bound on accuracy. After editing, the performance pattern reverses, with the directed prompts becoming becoming easier to answer by an average of 36.6%. This suggests that current editing frameworks establish strong subject-relation patterns during training. When these learned patterns are disrupted by the Undirected+ prompt structure, the model fails to successfully recall edits, despite having correctly answered the same factual prompt prior to editing.

5.3 Compound Probing

We compare framework performance on the Efficacy Score (ES) and Compound Probing Score (CPS) on the SEED Person dataset using GPT-J in Figure 5. All editing frameworks demonstrate substantially higher performance on ES by an average of 22.1%. This performance gap suggests that current editing frameworks face challenges in inserting edits that can be reliably recalled by models when processing multiple edits in sequence. We believe this limitation stems from the training structure of these editors, which optimize for model recall over related non-edit knowledge rather than incorporating other existing edits.

5.4 Full Dataset Evaluation

We present parameter-based editing framework results on the SEED Person dataset in Table 3. Consistent with prior works, all frameworks achieve high Efficacy Scores (ES), indicating that frameworks are still able to insert edits and directly recall edits with multiple edits per subject. Neighborhood Score (NS) is notably low, with an average of 43.9%. Frameworks are easily maintaining the exact edit knowledge representation at the cost of non-edit subject knowledge. Future frameworks need better mechanisms for improving edit locality. Performance on the Compound Probing Score is also lower, dropping below ES by 22.1% and Generality Score (GS) by 9.8% across both models. This pattern indicates that while frameworks successfully recall edits both directly and under paraphrase, their performance degrades when multiple edited facts are present in the input prompt. The Compound Probing Perfect Score (CPPS) reveals an even sharper decline, demonstrating the difficulty of maintaining correctness across all edited knowledge of a subject in a single prompt.

We evaluate editing performance for text generation using the Undirected+ prompt with one relation included at the input. For the machine translation metrics, we treat the pre-edit model output as the ground truth and the post-edit model output as the prediction. Applying edits greatly reduces lexical score. Ideally, successful edits would preserve text structure while modifying specific factual content, yielding lexical similarities in the 80s to 90s range. Instead, we observe substantial divergence from pre-edit generations. Some frameworks also achieve similarity scores of 100, indicating that

Metric	Next Token Prediction				Compound Probing Score		Lexical Similarity		Semantic Similarity	String Matching
	S	ES	GS	NS	CPS	CPPS	BLEU	chrF	BERTScore	SM
Model						GPT-J				Size: 6B
FT	66.0	96.9	77.1	45.1	66.8	28.0	7.3	20.6	53.5	44.9
GRACE	37.6	33.7	29.0	63.8	26.3	3.0	100.0	100.0	100.0	18.1
ROME	32.2	62.9	56.6	16.8	61.4	22.5	2.8	18.1	51.9	63.7
MEMIT	58.9	83.5	60.5	44.6	51.8	15.4	25.0	40.3	65.7	55.5
PMET	54.8	87.6	71.0	34.2	57.8	19.3	13.4	29.1	60.5	57.3
SWEA	39.4	98.6	96.2	18.0	58.2	21.2	19.9	35.7	62.1	52.9
Model						Llama 3.1				Size: 8B
FT	54.4	89.5	72.8	33.1	65.1	24.0	1.2	13.0	39.9	27.3
GRACE	38.5	31.8	29.8	77.9	19.2	1.8	100.0	100.0	100.0	15.2
ROME	31.8	50.7	50.7	18.2	52.0	14.0	0.0	0.0	24.4	0.1
MEMIT	45.4	41.3	35.5	73.0	26.5	2.9	32.5	50.7	74.8	19.9
PMET	56.1	68.7	49.4	53.7	42.6	9.5	15.2	35.7	66.5	32.2
SWEA	43.4	66.5	29.8	48.8	19.2	1.8	100.0	100.0	100.0	15.2

Table 3: Results of parameter-based editors on the SEED Person dataset, using the GPT-J and Llama 3.1 models.

model has been insufficiently modified to induce edit-related changes. Despite low lexical overlap, the editors fare better on semantic similarity. After excluding frameworks that failed to elicit edits, semantic similarity averages to 56.3%, suggesting that part of the text meaning has been preserved and editing has not fully degraded model text generation capabilities. Frameworks also demonstrate limited success with string matching, achieving an average of 48.7% on GPT-J and 9.2% on Llama 3.1. This difference suggests that larger models present greater challenges for evoking edits in text generation scenarios. Interestingly, GRACE and SWEA exhibit lexical and semantic similarity scores of 100 despite non-zero string matching, indicating that these perfect scores reflect rounding artifacts rather than complete editing failure. The frameworks induced minor changes, but remained insufficient for robust edit integration.

We provide memory-based editing framework results on 1000 edits and corresponding questions from the SEED Person dataset in Table 4. While memory-based editors are primarily designed for multi-hop reasoning, they should retain the ability to answer single-hop questions. The frameworks achieve an average 38.0% per-question (Q-Acc) and 90.2% per-case (C-Acc) accuracy. The differences in accuracies indicates that multi-hop editing frameworks are able to answer at least one edit correctly from most of the subjects in the dataset, but fail to consistently answer all edit questions for individual subjects. This suggests that the frameworks may be assigning disproportionately high similarity scores to a small subset of edits per subject, leading to selective recall. Rather than retrieving all relevant edits for a given subject, the retrieval process favors specific edit instances, resulting in incomplete coverage across the full set of edits per subject.

Metric	Q-Acc	C-Acc
Model	GPT-J	6B
PokeMQA	25.0	81.5
DeepEdit	39.8	92.7
CHECK	38.0	91.6
Model	Llama 3.1	8B
PokeMQA	26.1	82.6
DeepEdit	49.8	97.2
CHECK	49.3	95.5

Table 4: Results of memory-based editors on the the SEED Person dataset using GPT-J and Llama 3.1.

6 Conclusion

In this paper, we introduce the SEED dataset, which is of composed 1-to-n edits, undirected and compound probing prompts, and string matching metrics. Evaluations show editing performance decreases by 36.9% as the edits per subject scales.

7 Limitations

There are two main limitations of the SEED dataset:

1. Edits are provided in English only. This limits discussion of editing performance to English-based models. Future works should consider approaches for other languages and possibly multilingual editing.

2. Our dataset only covers evaluations over structured editing, where knowledge is formatted as relational triples. Some recent work has investigated unstructured editing, where knowledge is presented as text paragraphs. However, unstructured knowledge editors can be evaluated on SEED by using the Compound Probing Prompt as the set of edits.

It should also be pointed out that, like other knowledge editing works, all results are empirical. The specific size and configuration of data used directly impacts model performance.

Knowledge Editing can be potentially harmful if a malicious actor gains control of a commercial model (i.e. Claude) and changes the model weights to output malicious text. However, that would require breaking through enterprise cybersecurity solutions, which we image would be very difficult to do.

References

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. 2024. How do large language models acquire factual knowledge during pretraining? In *Advances in Neural Information Processing Systems*, volume 37, pages 60626–60668. Curran Associates, Inc.

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. Everything is editable: Extend knowledge editing to unstructured data in large language models. *Preprint*, arXiv:2405.15349.

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.

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*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2024. PokeMQA: Programmable knowledge editing for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8069–8083, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: lifelong model editing with discrete key-value adaptors. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. WilKE: Wise-layer knowledge editor for lifelong knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3476–3503, Bangkok, Thailand. Association for Computational Linguistics.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Mingyang Wan, Guojun Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Anyedit: Edit any knowledge encoded in language models. In *Forty-second International Conference on Machine Learning*.

Byungjun Kim, Minju Kim, Dayeon Seo, and Bugeun Kim. 2025. Leveraging large language models for active merchant non-player characters. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 10108–10116. International Joint Conferences on Artificial Intelligence Organization. AI, Arts & Creativity.

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)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Xuying Meng, Siqi Fan, Peng Han, Jing Li, Li Du, Bowen Qin, Zheng Zhang, Aixin Sun, and Yequan Wang. 2025a. Flm-101b: An open llm and how to train it with $100kbudget$. *Preprint*, arXiv:2309.03852.

Xiaopeng Li, Shasha Li, Shezheng Song, Huijun Liu, Bin Ji, Xi Wang, Jun Ma, Jie Yu, Xiaodong Liu,

Jing Wang, and Weimin Zhang. 2025b. Swea: Updating factual knowledge in large language models via subject word embedding altering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24494–24502.

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

Weixin Liang, Yaohui Zhang, Mihai Codreanu, Jiayu Wang, Hancheng Cao, and James Zou. 2025. The widespread adoption of large language model-assisted writing across society. *Patterns*, 6(12):101366.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Toby Walsh, Armin Hamrah, Lapo Santarlasci, and 4 others. 2025. Artificial intelligence index report 2025. *Preprint*, arXiv:2504.07139.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Dominic Simon and Rickard Ewetz. 2025. Knowledge editing for multi-hop question answering using semantic analysis. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 8241–8249. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature Medicine*, 29:1930–1940.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

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

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

Zihao Wei, Liang Pang, Hanxing Ding, Jingcheng Deng, Huawei Shen, and Xueqi Cheng. 2024. Stable knowledge editing in large language models. *Preprint*, arXiv:2402.13048.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zexuan Zhong, Zhengxuan Wu, Christopher 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*, pages 15686–15702, Singapore. Association for Computational Linguistics.

A Overview

This Appendix is composed of 5 additional sections: a statement on AI usage in the writing of the paper is given in Sectoin B, the prompts used during dataset construction and additional statistics are given in Section C, exact Undirected probing prompts used during evaluations are given in Section D, parameters for each framework used in evaluation and the hardware setup are given in Section E, and results over additional ablations and the remaining SEED datasets are provided in Section F.

B AI Usage

During the research and manuscript writing process, AI was only used in two ways: code for visualization generation and rephrasing excerpts of text for writing clarity. We used AI to generate skeleton code for `matplotlib` scripts, and modified the code for our own purposes. We also used AI for assistance in improving language presentation and removing grammatical mistakes. However, all ideas presented in this manuscript are our own.

C Dataset Construction Details

In this section, we provide additional information on how the SEED dataset was constructed. We provide all LLM prompts and other templates used in Section C.1. Additional dataset statistics are provided in Section C.2.

C.1 Construction Templates

We provide examples of some of the sentence templates used for formatting edits in Figure 6. SUBJ refers to where the edit subject needs to be substituted in. OBJ refers to where the edit object needs to be substituted in. The majority of templates were handcrafted by the authors, however some sentence templates from the ParaRel (Elazar et al., 2021) dataset were also used when selected relations overlapped between datasets.

We provide all of the entity types used to pull subjects from Wikidata in Figure 7. The entity types were chosen to provide a variety of different subjects that all relate back to the theme of the respective dataset. The Person dataset only uses a single entity types due to the prevalence of human subjects in Wikidata.

We provide an example of the Compound Probing Prompt in Figure 8. SUBJ is the formatting specifier for the edit subject to be inserted. The prompt is formatted with a predetermined number of triples and passed to an LLM, which formats each of the triples as a sentence. The number of triples used in model call can be varied. The final prompt is a paragraph containing a sentence on each of the triples.

Relation: place of birth
Edit Sentence: The location of SUBJ's birth was OBJ.

Relation: occupation
Edit Sentence: The occupation of SUBJ is OBJ.

Relation: founded by
Edit Sentence: One of the founders of SUBJ is OBJ.

Relation: location of formation
Edit Sentence: The location that SUBJ was founded in is OBJ.

Relation: genre
Edit Sentence: The genre that the work SUBJ belongs to is OBJ.

Relation: creator
Edit Sentence: SUBJ was created by OBJ.

Figure 6: Examples of edit sentences created for use during dataset construction. The edit sentences are used for both edit insertion and evaluation. *SUBJ* refers to the sentence subject and *OBJ* refers to the sentence object.

C.2 Statistics

The top 7 relations for each dataset subset are given in Figure 9. We selected from the top 25% of relations for the respective entity types. Human entities in Wikidata exhibit substantially more shared descriptor relations compared to business- and art-related entities, which show greater relational diversity. This pattern is reflected in the overall distribution of relation occurrences across the dataset.

The number of edits per subject for each dataset subset are shown in Figure 10. The numberof edits per subject directly correlates with the prevalence of commonly shared relations. Subsets with higher relational overlap contain more subjects with multiple edits, while those with greater relational diversity show fewer edits per entity.

Art	Business	Person
video game	business	human
film	public company	
television series	fast food restaurant chain	
literary work	enterprise	
manga series	retail chain	
animated television series	technology company	
anime television series	restaurant chain	
anime film	video game developer	
painting	video game publisher	
musical group	publishing company	
	book publisher	

Figure 7: All of the entity types used to filter entities for the **Art**, **Business**, and **Person** subsets of SEED.

Triples:
(Camp Lazlo, publisher, Crave Entertainment)
(Camp Lazlo, creator, Joe Murray)
(Camp Lazlo, country of origin, Uniated States)
(Camp Lazlo, composer, Toshihiko Takahashi)
(Camp Lazlo, production company, Cartoon Network Studios)
(Camp Lazlo, genre, adventure television series)

LLM Prompt:
Provide information about SUBJ. Make the response seem slightly natural. In the response, only include information from the following relational triples as a single sentence each:
(Camp Lazlo, publisher, Crave Entertainment)
(Camp Lazlo, creator, Joe Murray)
...
(Camp Lazlo, genre, adventure television series)

Compounding Edit Prompt:
Camp Lazlo is a popular animated television series that was developed by Joe Murray and published by Crave Entertainment. Camp Lazlo is an American animated television series. The show's score was composed by Toshihiko Takahashi. Camp Lazlo is an animated television show that originated from the production company Cartoon Network Studios. It's an adventure television series.

Figure 8: An example of the Compound Probing Prompt used during evaluation, based on the subject *Camp Lazlo*.

D Undirected Evaluation Prompts

We provide an example of the *Undirected* and *Undirected+* evaluation prompts in Figure 11. The Undirected prompt asks for information about an entity without specifying what information should be provided. The entity type is also provided to aid in entity disambiguation. The Undirected+ prompt builds off of the Undirected prompt by also asking for specific information to be provided. The number of relations included in the prompt can be varied. In both prompts, we also highlight the model prompt with *Prompt:* and the point where the model needs to answer with *Response:* to try to help the model understand this is a question-answering scenario so that it will provide relevant information.

E Evaluation Parameters

For reproducibility purposes, we provide the framework parameters used by parameter-based editors in Section E.1 and memory-based editors in Section E.2. We also provide the hardware setup in Section E.3.

E.1 Parameter-based Frameworks

We provide evaluations over 6 parameter-based edits that follow the same parameter style file setup used in ROME (Meng et al., 2022). The code for the frameworks are directly taken from the SWEA (Li et al., 2025b) Github repository linked

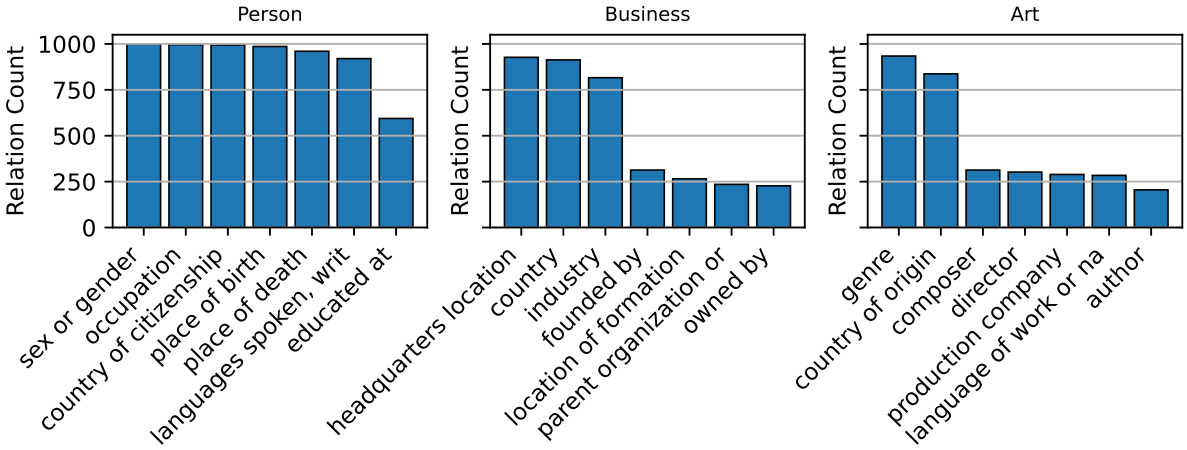

Figure 9: The top 7 relations used in each of the SEED subsets.

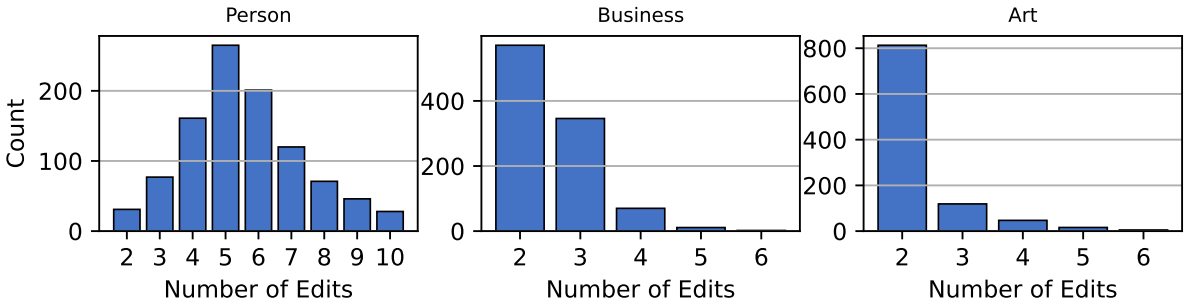

Figure 10: The number of subject entities categorized by the number of inserted edits.

here. We provide the parameters for each framework based on the ROME file setup below:

- **Layer-wise Fine-Tuning (FT)**: *GPT-J* - [Layers: 21, num_steps: 25, lr: $5e^{-4}$, weight_decay: 0, kl_factor: 0, norm_constraint: false]. *Llama 3.1 8B* - [Layers: 21, num_steps: 30, lr: $5e^{-4}$, weight_decay: 0, kl_factor: 0, norm_constraint: false].

- **General Retrieval Adaptor for Continual Editing (GRACE)**: *GPT-J* - [edit_lr: 1.0, n_iter: 25, eps: 1.0, dist_fn: euc, val_init: cold, val_train: sgd, val_reg: null, reg: early_stop, replacement: replace_last, eps_expand: coverage, num_pert: 8, dropout: 0.0]. *Llama 3.1 8B* - [edit_lr: 1.0, n_iter: 30, eps: 1.0, dist_fn: euc, val_init: cold, val_train: sgd, val_reg: null, reg: early_stop, replacement: replace_last, eps_expand: coverage, num_pert: 8, dropout: 0.0].

- **Rank One Model Editing (ROME)**: *GPT-J* - [layers: 5, fact_token: subject_last, v_num_grad_steps: 25, v_lr: $5e^{-1}$, v_loss_layer: 27, v_weight_decay: 0.5, clamp_norm_factor: 4, kl_factor: 0.0625, mom2_adjustment: true, context_template_length_params: [[5, 2], [10, 2]], mom2_n_samples: 100000]. *Llama 3.1 8B* - [layers: 5, fact_token: subject_last, v_num_grad_steps: 30, v_lr: $5e^{-1}$, v_loss_layer: 31, v_weight_decay: $1e^{-3}$, clamp_norm_factor: 4, kl_factor: 0.0625, mom2_adjustment: false, context_template_length_params: [[5, 10], [10, 10]], mom2_n_samples: 100000]

- **Mass Editing Memory in a Transformer (MEMIT)**: *GPT-J* - [layers: [3, 4, 5, 6, 7, 8], clamp_norm_factor: 0.75, layer_selection: all, fact_token: subject_last, v_num_grad_steps: 25, v_lr: $5e^{-1}$, v_loss_layer: 27, v_weight_decay: 0.5, kl_factor: 0.0625, mom2_adjustment: true, mom2_update_weight: 15000, mom2_n_samples: 100000]. *Llama 3.1 8B* -

Given knowledge: **(Spirited Away, director, Hayao Miyazaki)**.

Undirected

Prompt: Provide information about the animated movie named Spirited Away.
Response:

Undirected+

Prompt: Provide information about the animated movie named Spirited Away. In the response, only include information about Spirited Away for the following factual attributes:
director
Response:

Figure 11: The *Undirected* and *Undirected+* prompts used to evaluate unstructured text with SEED.

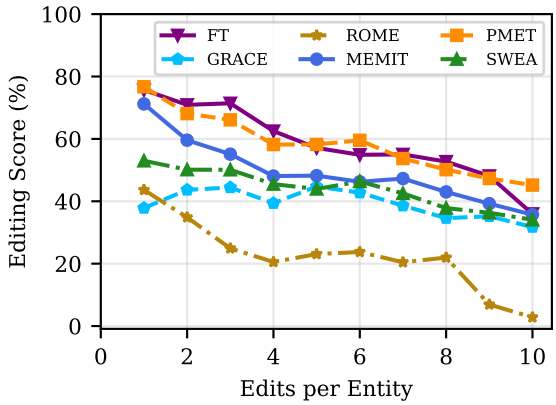

Figure 12: Editing Score, the harmonic mean over ES, GS, and NS, achieved by the FT, ROME, GRACE, MEMIT, PMET, and SWEA editing frameworks over 100 edits as the number of edits per entity is increased using the Llama 3.1 8B model.

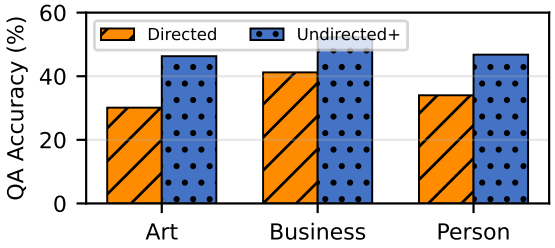

Figure 13: Comparison of GPT-J using Directed and Undirected probing to elicit targeted knowledge on the SEED datasets, measured with String Matching.

[layers: [4, 5, 6, 7, 8], clamp_norm_factor: 0.75, layer_selection: all, fact_token: subject_last, v_num_grad_steps: 30, v_lr: $1e^{-1}$, v_loss_layer: 31, v_weight_decay: 0.5, kl_factor: 0.0625, mom2_adjustment: true, mom2_update_weight: 15000, mom2_n_samples: 100000].

- **Precise Model Editing in a Transformer (PMET)**: *GPT-J* - [layers: [3, 4, 5, 6, 7, 8], clamp_norm_factor: 0.75, layer_selection: all, fact_token: subject_last, v_num_grad_steps: 25, v_lr: $2e^{-1}$, v_loss_layer: 27, v_weight_decay: 0.5, kl_factor: 1.0, mom2_adjustment: true, mom2_update_weight: 6000, mom2_n_samples: 100000, nll_loss_factor: 1.0]. *Llama 3.1 8B* - [layers: [4, 5, 6, 7, 8], clamp_norm_factor: 0.75, layer_selection: all, fact_token: subject_last, v_num_grad_steps: 30, v_lr: $1e^{-1}$, v_loss_layer: 31, v_weight_decay: 0.5, kl_factor: 0.0625, mom2_adjustment: true, mom2_update_weight: 6000, mom2_n_samples: 100000, nll_loss_factor: 1.0].

- **Subject Word Embedding Altering (SWEA)**: *GPT-J* - [clamp_norm_factor: 1.0, fact_token: subject_last, v_num_grad_steps: 25, v_lr: $2e^{-2}$, v_loss_layer: 27, v_weight_decay: 0.3, kl_factor: 0.2, nll_loss_factor: 1.0, mode: kn+optimize, kn_params: [cat, 0.35, 0]]. *Llama 3.1 8B* - [clamp_norm_factor: 1.0, fact_token: subject_last, v_num_grad_steps: 30, v_lr: $1e^{-2}$, v_loss_layer: 31, v_weight_decay: 0.5, kl_factor: 0.0625, nll_loss_factor: 1.0, mode: kn+optimize, kn_params: [cat, 0.35, 0]].

E.2 Memory-based Frameworks

We provide evaluations over 3 memory-based editors. The frameworks are taken from their respective author's Github repository: PokeMQA (Gu et al., 2024) from the Github repository linked

Dataset	Art	Business	Person
# Edits	1121	1420	2935
% Edits	41.0	56.3	52.4

Table 5: Filtered dataset statistics after removing all edits that the pre-edit model cannot correctly answer.

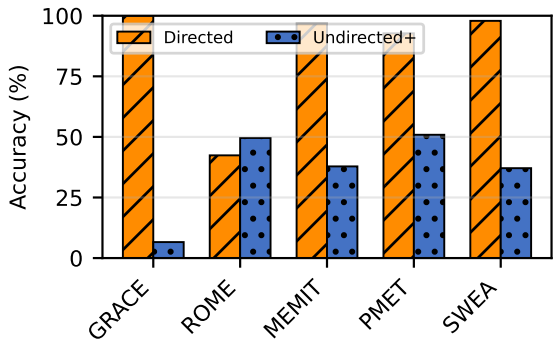

Figure 14: Comparison of frameworks using Directed and Undirected+ prompts. Accuracy is measured using String Matching on GPT-J generated text, with edits from the SEED Art dataset.

here, DeepEdit (Wang et al., 2024) from the Github repository linked here, and CHECK (Simon and Ewetz, 2025) from the Github repository linked here. For each model, we set the `maximum new tokens` generated from each model call to be 200. For PokeMQA and DeepEdit, we set the `maximum number of extracted questions hops` to be 5, so the framework will report a failure of the 5 question hop limit is exceeded. CHECK uses a `cosine similarity threshold` of 0.8 for retrieving relevant edits from the edit embeddings.

E.3 Hardware Setup

All code was run on a High Performance Computing cluster. Dataset construction used 1 NVIDIA B200, 16 CPU cores, and 256GB of RAM. Experimental evaluations were run using 1 NVIDIA B200, 8 CPU cores, and 64GB of RAM.

F Results, Continued

We provide 4 additional results sections: scaling the number of edits per subject using Llama 3.1 8B (Grattafiori et al., 2024) is given in Section F.1, further experiments comparing directed and undirected probing across the SEED datasets is given in Section F.2, a comparison of compound probing using Llama 3.1 8B is provided in Section F.3, and full dataset evaluations over SEED Art and Business are provided in Section F.4.

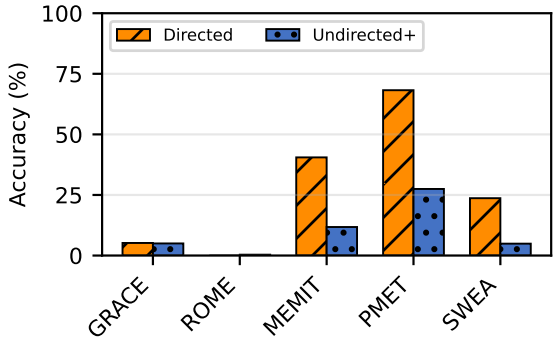

Figure 15: Comparison of frameworks using Directed and Undirected+ prompts. Accuracy is measured using String Matching on Llama 3.1 8B generated text, with edits from the SEED Art dataset.

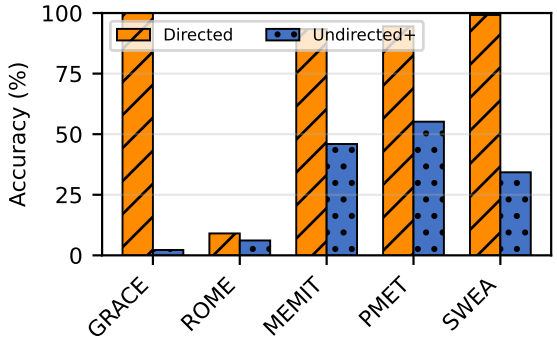

Figure 16: Comparison of frameworks using Directed and Undirected+ prompts. Accuracy is measured using String Matching on GPT-J generated text, with edits from the SEED Business dataset.

F.1 Scaling Edits per Subject

In this section, we provide another analysis of how editing performance scales with the number of edits applied to a single subject. The setup is the same as in Section 5.1, where we vary the number of edits applied to a single subject from 1 to 10 while keeping the total number of edits constant at 200. The subjects are still sampled from the SEED Person dataset. However, in this experiment, we are now using the Llama 3.1 (Grattafiori et al., 2024) instead. The results are shown in Figure 12.

The pattern of edit score degradation holds on the larger and newer Llama 3.1. As the number of edits per entity increases, Editing Score (S) continues to decrease. Between 1 and 3 edits per entity, the editing score decreases by an average of 7.7%. By 10 edits per entity, the editing score drops by an average of 28.8%.

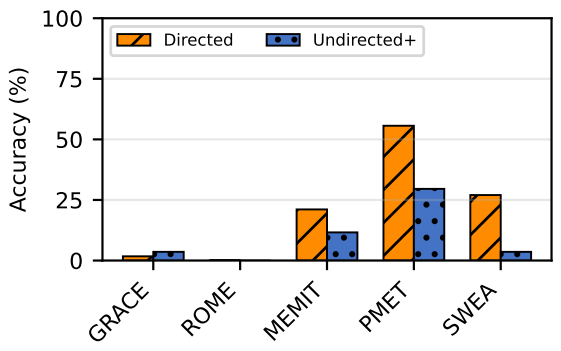

Figure 17: Comparison of frameworks using Directed and Undirected+ prompts. Accuracy is measured using String Matching on Llama 3.1 8B generated text, with edits from the SEED Business dataset.

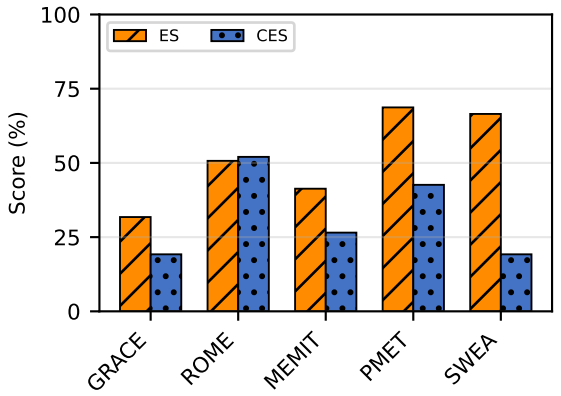

Figure 18: Comparison of framework performance on Efficacy Score (ES) and Compound Probing Score (CPS) on the SEED Person dataset using Llama 3.1.

F.2 Undirected Probing

Following the setup in Section 5.2, we evaluate GPT-J performance across all 3 SEED datasets prior to editing, using the original objects as the ground truth. All triples, both edited and unedited, are used in this evaluation because we are searching for inclusion of original facts. The results are given in Figure 13. With directed probing, GPT-J includes 30.1%, 41.21%, and 34.0% of relevant facts on the Art, Business, and Person datasets, respectively. With undirected probing, GPT-J includes 46.3%, 52.3%, and 46.8% on the Art, Business, and Person datasets, respectively. With undirected probing, GPT-J achieves a 13.3% higher string matching score based on fact inclusion, indicating that this prompting strategy allows language models to more easily include the desired information than directed prompting.

The exact number of edited relations that the model was able to include the correct original object for are included in Table 5. This number is also provided as a percentage of total edits in the datasets. We gather the edited relations correctly answered by the pre-edit model and distill them into a smaller dataset to evaluate post-editing model performance on the Directed and Undirected Probing strategies.

We now provide comparisons of Directed probing and Undirected probing with String Matching accuracy. Results using the SEED Art dataset with GPT-J are given in Figure 14 and with Llama 3.1 in Figure 15. Results using the SEED Business dataset with GPT-J are given in Figure 16 and with Llama 3.1 in Figure 17. When using GPT-J across both the SEED Art and Business datasets, the results mimic those found in Figure 4. The edited model is able to include the edit objects more often with directed prompts compared to undirected prompts, with a 49.6% accuracy differential on Art and 50.5% on Business. This is the opposite of what is expected given that the pre-edit model includes targeted objects in the generated text more easily when using the undirected prompt. This provides further evidence that current editors are teaching models a subject-relation pattern rather than a new robust knowledge representation.

The results using Llama 3.1 also demonstrate that edited models struggle significantly with undirected prompts. Notably, these results are greatly diminished compared to using GPT-J. However, this is in line with results in Table 6 and Table 7, which shows that the editors struggle to successfully apply edits to the newer and larger Llama 3.1.

F.3 Compound Probing

Following the setup in Section 5.3, we compare framework performance on the Efficacy Score (ES) and Compound Probing Score (CPS) on the SEED Person dataset. In this experiment, we use Llama 3.1. Results are presented in Figure 18. Similar to the results shown in Section 5.3, frameworks perform better on the Compound Probing Score (CPS) compared to the Efficacy Score (ES) by 19.9%.

F.4 Full Dataset Evaluations

We provide results of parameter-based editing frameworks on the SEED Art dataset on Table 6 and the SEED Business dataset in Table 7. These results follow the same trends found in Table 3. The Efficacy Score (ES) and Generality Score (GS)

Metric	Next Token Prediction				Compound Probing Score		Lexical Similarity		Semantic Similarity	String Matching
	S	ES	GS	NS	CPS	CPPS	BLEU	chrF	BERTScore	SM
Model						GPT-J				Size: 6B
FT	72.6	98.4	72.7	57.4	63.0	43.1	5.2	17.4	49.7	32.1
GRACE	43.7	38.8	33.0	79.4	19.7	7.0	100.0	100.0	100.0	6.9
ROME	46.0	51.9	50.2	38.5	48.0	26.4	0.6	12.7	40.2	49.5
MEMIT	69.4	95.3	73.3	52.4	51.5	32.3	13.7	32.5	59.1	37.8
PMET	65.7	97.8	79.2	43.9	56.3	36.4	20.4	38.2	63.7	50.9
SWEA	45.8	99.0	97.8	22.1	61.3	43.8	20.6	37.8	62.6	37.1
Model						Llama 3.1				Size: 8B
FT	71.9	94.7	76.8	55.1	69.0	50.4	1.9	11.8	36.5	33.9
GRACE	36.2	26.6	29.4	88.9	13.7	5.3	100.0	100.0	100.0	5.0
ROME	40.1	50.2	50.9	28.4	47.9	28.2	0.0	7.1	27.4	0.4
MEMIT	55.5	57.1	41.0	82.3	23.5	9.3	31.5	51.2	74.3	11.8
PMET	69.2	84.3	62.4	64.6	42.7	22.6	19.6	41.5	68.6	27.5
SWEA	43.8	51.8	30.6	60.5	16.7	7.8	97.9	98.2	98.9	4.9

Table 6: Results of parameter-based editors on the SEED Art dataset, using the GPT-J and Llama 3.1 models.

are high, with an average of 70.5% and 58.1% on Art and 69.5% and 55.9% on Business, confirming that the framework can apply multiple edits to the same subject and then directly recalling those edits. However, applying so many edits to the same subject negatively impacts model ability to recall related non-edit knowledge, with an average Neighborhood Score (NS) of 56.1% on Art and 48.3% on Business. Clearly, current parameter-based frameworks need to improve the locality of their edits. The Compound Probing Scores are still low, with an average score of 42.8% on Art and 45.7%. The lexical and semantic similarity scores also follow the same trends described before. The edited model generates text that significantly diverges from the pre-edit model. However, the average semantic similarity score of 53.6 on Art and 58.3% on Business indicate that the generated text is not fully diverging in content. Editors struggle to insert edits in such a way that causes edits to be generated by models during text generation, with an average string matching accuracy of 31.2% on Art and 26.6% on Business for editors that made enough of a change in model parameters to cause generated text to change.

We provide results for memory-based editors on the SEED Art and Business datasets in Table 8. The frameworks achieve an average per-question accuracy of 40.3% on Art and 36.9% on Business, and an average per-case accuracy of 68.8% on Art and 69.5% on Business. Similar to results on the SEED Person dataset in Table 4, these results demonstrate that multi-hop editing frameworks can successfully answer at least one question related to an edited subject correctly, but are largely unable to answer every edit related to subject. The likely cause is from editors inferring similar non-edit knowledge as edits.

Metric	Next Token Prediction				Compound Probing Score		Lexical Similarity		Semantic Similarity	String Matching
	S	ES	GS	NS	CPS	CPPS	BLEU	chrF	BERTScore	SM
Model						GPT-J				Size: 6B
FT	65.9	99.1	69.5	47.5	70.1	49.0	18.5	31.6	64.5	27.6
GRACE	40.6	37.4	29.6	74.5	23.6	7.4	100.0	100.0	100.0	2.2
ROME	37.6	54.3	49.4	24.3	51.1	28.3	0.4	10.8	41.4	6.1
MEMIT	57.6	93.7	71.8	36.4	58.5	37.5	14.6	34.0	62.5	45.9
PMET	52.4	96.3	77.0	29.5	64.5	43.5	27.9	45.1	68.5	55.2
SWEA	43.3	99.3	96.8	20.4	62.7	41.9	19.9	37.0	64.8	34.3
Model						Llama 3.1				Size: 8B
FT	66.8	91.8	68.8	51.4	65.6	43.9	2.5	16.9	44.0	28.6
GRACE	40.5	32.5	31.4	86.2	17.1	4.6	99.9	100.0	100.0	3.6
ROME	40.3	48.5	48.6	30.0	47.6	26.5	0.0	0.0	29.7	0.0
MEMIT	50.5	48.3	39.6	74.4	27.5	10.8	36.4	57.6	77.6	11.6
PMET	61.5	78.6	57.0	53.9	43.2	22.0	21.7	46.4	72.1	29.6
SWEA	43.1	54.1	31.6	51.1	17.3	4.8	99.4	99.6	99.8	3.6

Table 7: Results of parameter-based editors on the SEED Business dataset, using the GPT-J and Llama 3.1 models.

Dataset	Art		Business	
Metric	Q-Acc	C-Acc	Q-Acc	C-Acc
Model	GPT-J			6B
PokeMQA	39.3	66.3	34.2	67.8
DeepEdit	47.2	81.4	38.0	75.9
CHECK	39.8	70.1	31.7	59.9
Model	Llama 3.1			8B
PokeMQA	29.7	56.3	28.0	58.6
DeepEdit	58.8	88.9	58.5	92.3
CHECK	26.8	49.5	30.9	62.4

Table 8: Results of memory-based editors on the the SEED Art and Business datasets using GPT-J and Llama 3.1.