REACT: Representation Extraction And Controllable Tuning to Overcome Overfitting in LLM Knowledge Editing

Anonymous ACL submission

Abstract

Large language model editing methods frequently encounter overfitting, wherein factual updates disproportionately influence the model's broader behavior, causing it to adhere rigidly to the edited target regardless of the query context. To address this challenge, we introduce **REACT** (Representation Extraction And Controllable Tuning), a dual-phase framework designed for precise and scalable knowledge editing. In the initial phase, we utilize tailored stimuli with Principal Component Analysis to extract latent factual representations and derive a directional "belief shift" vector. In the subsequent phase, a pre-trained classifier guides the selective perturbation of hidden states via a learned scalar, ensuring that modifications remain confined to relevant regions of the latent space. This strategy is further refined through a composite loss function that balances editing and localization objectives, ultimately integrating new information effectively while preserving unrelated knowledge. Empirical evaluations on COUNTERFACT, MQuAKE, and EVOKE benchmarks demonstrate that **REACT** significantly mitigates overfitting and enhances reliability, portability, and generality across diverse editing scenarios.

1 Introduction

Large language models (LLMs) have become indispensable in modern applications, powering a wide array of systems from chatbots to content generators(Zhao et al., 2023; Xu et al., 2024). Despite their widespread utility, ensuring that these models maintain up-to-date and accurate factual information remains a critical challenge, particularly when extensive retraining is impractical(Zhang et al., 2024b). This necessity has spurred interest in the field of knowledge editing, where targeted updates to a model's internal knowledge base are pursued without compromising overall performance(Wang et al., 2023; Yao et al., 2023; Cheng et al., 2023).

Recent advances in knowledge editing have sought to address these issues by incrementally incorporating new facts into LLMs(De Cao et al., 2021a). However, many existing approaches encounter significant challenges, like *overfitting during editing process*(Zhang et al., 2024a). Concretely, this occurs when a model, after being updated with new knowledge, becomes excessively specialized to the edited samples. For example, consider an update where the statement "Luka Doncic plays in the NBA team of **Mavericks**" is corrected to "Luka Doncic plays in the NBA team of **Lakers**." In an overfit scenario, when queried with "Who does Luka Doncic play with?", the model may still disproportionately favor the edit target but not the correct answer—assigning a high probability to "Mavericks"—while the probabilities for more contextually appropriate responses, such as teammates like Austin Reaves or LeBron James, remain undesirably low. These limitations not only compromise the reliability of the updates but also hinder the practical deployment of such techniques in real-world systems, highlighting a crucial gap in current methodologies.

In response to these challenges, we propose a novel framework that leverages a dual-phase representation pipeline to perform targeted knowledge edits via representation engineering(Andy Zou, 2023). In the first phase—*Extracting Latent Knowledge Representations*—we employ tailored input stimuli, Principal Component Analysis (PCA), and learnable linear transformations to distill the model's internal factual representations. For each factual instance, we generate a stimulus pair and compute a directional vector that encapsulates the latent "belief" shift associated with the edit. In the subsequent phase—*Selective Perturbing Representations Controllably*—a pre-trained classifier determines which hidden states require modification, and a learned scalar governs the magnitude of the update based on the alignment between the

hidden state and the directional vector. This selective mechanism ensures that only the relevant knowledge of the latent space are perturbed while preserving unrelated knowledge.

Our contributions can be summarized as follows:

- **Mitigating Overfitting in Knowledge Editing**: We overcome a critical limitation of existing knowledge editing approaches—overfitting—by designing a method that prevents excessive specialization to edited facts while preserving broader model generalization.
- **A Controllable Hidden-State Editing Framework**: We propose a novel dual-phase editing pipeline that operates directly on the model's internal representations, enabling precise and scalable knowledge updates through selective perturbation of hidden states.
- **Empirical Validation Across Diverse Benchmarks**: We extensively evaluate our approach on multiple knowledge editing datasets (COUNTERFACT, MQuAKE, and EVOKE), demonstrating superior factual accuracy, reduced overfitting, and improved robustness compared to state-of-the-art baselines.

2 REACT: Representation Extraction And Controllable Tuning to overcome overfitting

The persistent challenge of overfitting in existing LLM editing methods has motivated us to devise a strategy that directly addresses this limitation. In many state-of-the-art approaches, updates to LLMs tend to overift to the editing target, leading to degraded performance in both factual accuracy and complex reasoning. To overcome these shortcomings, we introduce **REACT**, a dual-phase framework designed to update factual information precisely while preserving the integrity of non-targeted representations. Our method achieves this by decoupling the editing process into two complementary stages: (i) *representation extraction* from latent knowledge to isolate the essential factual components, and (ii) *selective perturbation* to refine internal representations in a controlled manner. This separation not only enables targeted updates but also significantly mitigates the risk of overfitting, thereby ensuring robust and reliable model performance.

2.1 Phase I: Extracting Latent Knowledge Representations

In this phase, the model's internal representations of factual knowledge are extracted using tailored input prompts, or *stimuli*(Andy Zou, 2023). For each factual instance, we generate a stimulus pair—a positive instance and a negative instance which only differs from each other by the subject—using an identical template. Each stimulus is passed through the model to obtain layer-wise hidden representations, denoted as $\mathbf{h}^{(l)}$ for a chosen layer l.

To capture a comprehensive picture, we collect $N = 512$ stimulus pairs $\{(\mathbf{h}^{(l)}_{+,i}, \mathbf{h}^{(l)}_{-,i})\}_{i=1}^{N}$. We then apply Principal Component Analysis (PCA) to these representations to extract the principal component $\{(\mathbf{h}^{(l)}_{+}, \mathbf{h}^{(l)}_{-}\}$, which summarizes the key directional change in the latent space. Instead of directly subtracting the negative from the positive representation, we process the difference through a linear transformation:

$$\mathbf{v} = \mathbf{W}\left[\mathbf{h}^{(l)}_{+}; \mathbf{h}^{(l)}_{-}\right] + \mathbf{b}, \tag{1}$$

where $\left[\mathbf{h}^{(l)}_{+}; \mathbf{h}^{(l)}_{-}\right]$ denotes the concatenation of $\mathbf{h}^{(l)}_{+}$ and $\mathbf{h}^{(l)}_{-}$, $\mathbf{W} \in \mathbb{R}^{2d\times d}$ is the learnable weight matrix, and $\mathbf{b} \in \mathbb{R}^{d}$ is the bias vector. The vector $\mathbf{v}$ thus encapsulates the latent "belief" shift before and after an edit.

2.2 Phase II: Selective Perturbing Representations Controllably

Once the directional vector $\mathbf{v}$ is obtained, we proceed with a controllable editing phase. Here a *pretrained classifier* (denoted Φ) produces a probability $\Phi(\mathbf{h}) \in [0, 1]$ indicating whether a hidden state $\mathbf{h}$ from the Transformer decoder block(Andy Zou, 2023) should be edited or not. A *learned scalar* α then determines the magnitude of the update, and the sign of the update is based on the dot-product. Concretely, we apply:

$$\mathbf{h}' = \begin{cases} \mathbf{h} + \alpha \times \text{sign}(\mathbf{h}^{\mathrm{T}} v) \times \mathbf{v}, & \text{if } \Phi(\mathbf{h}) > 0.5, \\ \mathbf{h}, & \text{otherwise.} \end{cases} \tag{2}$$

Thus, only when $\Phi(\mathbf{h}) > 0.5$ do we add the perturbation $\alpha \times \text{sign}(\mathbf{h}^{\mathrm{T}} v) \times \mathbf{v}$ to the original hidden state $\mathbf{h}$. Otherwise, $\mathbf{h}$ remains unchanged. This selective mechanism confines edits to the relevant region of the latent space while preventing unnecessary alterations.

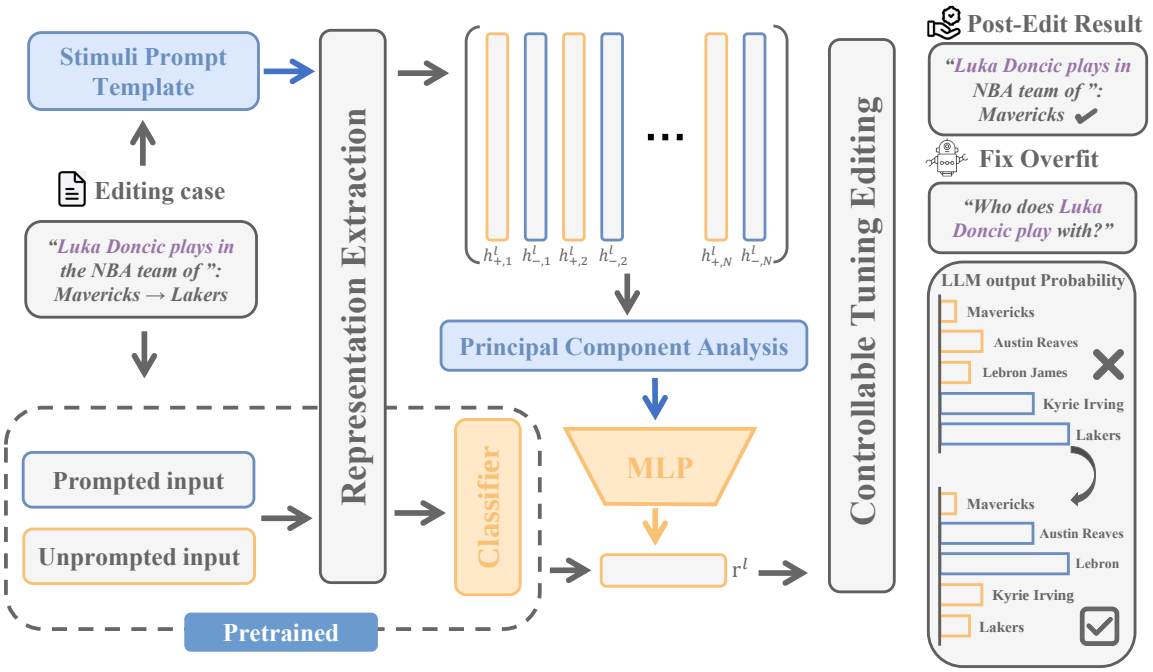

Figure 1: An overview of our REACT pipeline for controllable knowledge editing. We First construct stimuli prompts and feed them into a LLM to extract layer-wise representations, which are then processed via PCA and an MLP to isolate the key "belief shift" vector. Therafter, we apply a selective perturbation (using learned scalar factors) to the model's hidden states. A pre-trained classifier manages where and how the edits occur.

Editing Loss We aim to ensure the edited process genuinely updates the new factual knowledge. Let $\mathcal{D}$ be the dataset we train on, where we want the model G^* (with **REACT** applied) to output the updated fact o^*. Formally,

$$\mathcal{L}_{\text{edit}} = \mathbb{E}_{j\sim\mathcal{D}}\Big[-\log \mathbb{P}_{G^*}\big(o^* \mid x_j + p\big)\Big], \quad (3)$$

where p denotes any prompt or stimulus appended to x_j to trigger the newly inserted knowledge.

Localization Loss While the edit must be reflected in the model's outputs for relevant prompts, it should minimally affect unrelated inputs. Hence, we impose a KL-divergence penalty between the edited output distribution and the original output distribution:

$$\mathcal{L}_{\text{loc}} = D_{\text{KL}}\Big(\mathbb{P}_{G^*}[x \mid p'] \,\Big\|\, \mathbb{P}_{G}[x \mid p']\Big), \quad (4)$$

where p' indicates a prompt unrelated to the edited fact. By keeping the distance between these distributions low, we restrict the scope of the change to the intended knowledge only.

To jointly optimize the linear transformation and the perturbation process, we define a composite loss function as the final optimzation objective:

$$\mathcal{L}_{total} = c_{edit} \times \mathcal{L}_{edit} + c_{loc} \times \mathcal{L}_{loc}, \quad (5)$$

where c_{edit} and c_{loc} are hyperparameters balancing the two loss terms.

2.3 Details of Pre-trained classifier

Before the two phases, **REACT** employs a *multi-stage* procedure to pre-train a classifier that evaluates whether (and how) a hidden-state transformation should be applied to preserve semantic integrity. Specifically, let $\mathbf{h}^p$ and $\mathbf{h}^u$ denote the final-token embeddings of a *prompted* input (for a target fact) and an *unprompted* input (for a generic context), respectively. For each training instance, the language model produces these representations from a designated layer:

$$\mathbf{h}^p = \text{LM}\big(\text{prompted\_input}\big), \quad (6)$$
$$\mathbf{h}^u = \text{LM}\big(\text{unprompted\_input}\big). \quad (7)$$

Our attention-based classifier $\Psi(\cdot)$ learns distinct transformations for these two representatations. Specifically, we define learnable parameters $\mathbf{W}_Q$ and $\mathbf{W}_K$, which map each representation into $\mathbf{v}_q$ and $\mathbf{v}_k$ respectively:

$$\mathbf{v}_q = \mathbf{W}_Q\, \mathbf{h}_p, \quad (8)$$
$$\mathbf{v}_k = \mathbf{W}_K\, \mathbf{h}_u. \quad (9)$$

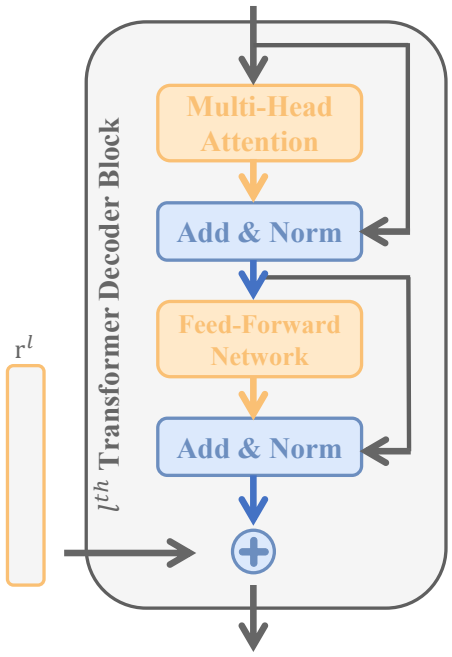

Figure 2: Illustration of the controllable editing mechanism within each Transformer decoder layer. The computed vector r_l is integrated into the model by adding it to the output of the l^{th} Transformer decoder block, enabling precise modifications to the model's knowledge representation.

A dot product between $\mathbf{v}_q$ and $\mathbf{v}_k$ yields a scalar measure of similarity:

$$\text{score} = \sum_d \left(\mathbf{v}_q^{(d)}\, \mathbf{v}_k^{(d)}\right). \tag{10}$$

To obtain a *final* similarity value $\gamma \in [0, 1]$, we apply normalization:

$$\gamma = \frac{\text{score}}{\|\mathbf{v}_q\|\, \|\mathbf{v}_k\| + 10^{-8}}$$

where $\|\cdot\|$ denotes the ℓ_2 norm. We then *threshold* γ at 0.5 to produce a binary decision:

$$\Psi(\mathbf{v}_p, \mathbf{v}_t) = \begin{cases} 1, & \text{if } \gamma > 0.5, \\ 0, & \text{otherwise.} \end{cases}$$

In this way, the classifier determines whether the fact-specific embedding $\mathbf{v}_p$ is sufficiently close to (or coherent with) the unprompted embedding $\mathbf{v}_t$, completing the ojective of applying **REACT** during the appropriate contents.

To encourage correct classification of edited vs. unedited representations, we incorporate *two* main loss components: an *editing loss* $\mathcal{L}_{\text{edit,cls}}$ and a *locality loss* $\mathcal{L}_{\text{loc,cls}}$. First, let $\Delta\mathbf{h}_i = \mathbf{h}_i^p - \mathbf{h}_i^u$ be the difference in embeddings for the i-th instance. We define:

$$\mathcal{L}_{\text{edit,class}} = \frac{1}{N}\sum_{i=1}^{N} \left\| \gamma_i\, \Delta\mathbf{h}_i \right\|_2^2, \tag{11}$$

$$\mathcal{L}_{\text{loc,class}} = \frac{1}{N}\sum_{i=1}^{N} \left\| (1-\gamma_i)\, \Delta\mathbf{h}_i \right\|_2^2. \tag{12}$$

Intuitively, $\mathcal{L}_{\text{edit,class}}$ encourages large $\Delta\mathbf{h}_i$ (i.e., *fact-specific* shifts) when γ_i is high (the model "believes" an edit is relevant), whereas $\mathcal{L}_{\text{loc,class}}$ penalizes such shifts when γ_i is low (i.e., for *unrelated* or unprompted contexts).

We then combine these losses:

$$\mathcal{L}_{\text{class}} = \lambda_{\text{edit}}\, \mathcal{L}_{\text{edit,class}} + \lambda_{\text{loc}}\, \mathcal{L}_{\text{loc,class}}, \tag{13}$$

where λ_{edit} and λ_{loc} are hyperparameters. We backpropagate $\mathcal{L}_{\text{class}}$ using *Adam optimizers* (Kingma and Ba, 2017), accompanied by *learning rate schedulers* and *gradient clipping* to maintain stable updates:

$$\mathbf{W}_\Psi \leftarrow \mathbf{W}_\Psi - \eta\, \nabla_{\mathbf{W}_\Psi} \mathcal{L}_{\text{class}}, \tag{14}$$

where η is the learning rate. Mini-batch processing and gradient accumulation are applied iteratively to refine Ψ over many epochs. This staged training regime enables the classifier to differentiate subtle semantic shifts while preserving contextual coherence, ultimately guiding the fact-editing process to stay *localized* and *accurate*.

3 Experimental Settings

3.1 Large Language Models

We evaluate our approach using two prominent language models: **Llama3.1-8B-instruct** (Grattafiori et al., 2024) and **Qwen2.5-7B-instruct** (Qwen et al., 2025). These models were selected for their robust ability to follow complex instructions and generate contextually coherent responses. Their open-source nature—providing full access to model weights—ensures transparency, reproducibility, and the opportunity for further customization. In addition, their strong performance across both standard benchmarks and real-world tasks makes them well-suited for rigorous experimental evaluation.

3.2 Knowledge Editing Baselines

Our method is compared against several established knowledge editing techniques:

Fine-Tuning (FT) Adapts pretrained LLMs to specific tasks by training on task-specific datasets. Fine-tuning updates model parameters to better align predictions with target outcomes by optimizing a loss function that minimizes the gap between predictions and ground truth.

MEND (Model Editor Networks using Gradient Decomposition) (Mitchell et al., 2022a) Employs auxiliary networks to facilitate fast, localized changes without full retraining. By applying low-rank decomposition to the gradients obtained during fine-tuning, MEND efficiently adjusts parameters.

MEMIT (Mass-Editing Memory in a Transformer) (Meng et al., 2023) Builds on the ROME framework to efficiently update LLMs with multiple factual associations. MEMIT targets neuron activations in middle-layer feed-forward modules to adjust weights directly, thereby modifying factual recall.

MELO (Model Editing with Neuron-Indexed Dynamic LoRA) (Zhong et al., 2023) Utilizes dynamically activated LoRA blocks—indexed through an internal vector database—to provide targeted and efficient updates.

3.3 Benchmarks

3.3.1 COUNTERFACT

COUNTERFACT(Meng et al., 2022a) comprises 21,919 records that cover a diverse range of subjects, relations, and linguistic variations. This dataset evaluates the model's ability to incorporate counterfactual edits by assessing whether it can store and retrieve new facts, focusing on substantive factual associations rather than superficial word changes.

Reliability Assesses how accurately the model performs on a given edit, focusing on its ability to maintain basic factual correctness for each specific modification:

$$\mathcal{M}_{\text{reliability}} = \mathop{\mathbb{E}}_{(i_e,x_e,y_e,y_e')\sim\mathcal{D}_{\text{edit}}} \mathbb{1}\left\{f\left(i_e,x_e\right)=y_e\right\}$$

Generality Evaluates the model's capacity to apply the edit correctly to in-scope data, ensuring that the model maintains generalization capabilities:

$$\mathcal{M}_{\text{generality}} = \mathop{\mathbb{E}}_{\substack{(i_e,x_e,y_e,y_e')\sim\mathcal{D}_{\text{edit}} \\ x_r\sim\mathcal{N}(x_e)}} \mathbb{1}\left\{f\left(i_e,x_r\right)=y_e\right\}$$

where the $\mathcal{N}(x_e)$ stands for the rephrased neighborhood of input text.

Locality Examines whether data outside the scope of the edit remains unaffected, preserving the model's performance on unrelated information.

$$\mathcal{M}_{\text{locality}} = \mathop{\mathbb{E}}_{(x_l,y_l)\sim\mathcal{D}_{\text{loc}}} \mathbb{1}\left\{f^*\left(x_l\right)=f(x_1)\right\}$$

3.3.2 MQuAKE

MQuAKE(Zhong et al., 2023) is a multi-hop benchmark designed to test knowledge editing in language models. By requiring the model to adjust related knowledge when updating individual facts, MQuAKE provides a comprehensive measure of the model's reasoning and adaptability following modifications.

Portability Evaluates the robustness of the generalization of the edit, evaluating whether the modified knowledge can be applied effectively to related content.

$$\mathcal{M}_{\text{port}} = \mathop{\mathbb{E}}_{\substack{(i_e,x_e,y_e,y_e')\sim\mathcal{D}_{\text{edit}} \\ (x_p,y_p)\sim\mathcal{P}(i_e,x_e,y_e,y_e')}} \mathbb{1}\left\{f\left(i_e,x_p\right)=y_p\right\}$$

where $\mathcal{P}\left(i_e,x_e,y_e,y_e'\right)$ denotes the Portability scope given input i_e, x_e, y_e and target output y_e'.

3.3.3 EVOKE

To rigorously assess overfitting tendencies in knowledge editing methods, we employ the **EVOKE** (EValuation of Editing Overfit in Knowledge Editing) benchmark(Zhang et al., 2024a). EVOKE is designed to analyze both the efficacy and generalization properties of edited models and comprises four overfit tasks:

Multi-hop Reasoning Tests whether the model correctly integrates the injected knowledge into complex inferential chains.

Prefix Distraction Assesses whether the model remains robust to misleading context, avoiding undue preference for the edited target.

Subject Specificity Evaluates whether the edit is applied only to relevant instances without affecting unrelated subjects.

Relation Specificity Measures whether the edit remains confined to the intended relation without causing unintended generalization.

We now define the key probability-based metrics to quantify the effectiveness of Overfit tasks editing.

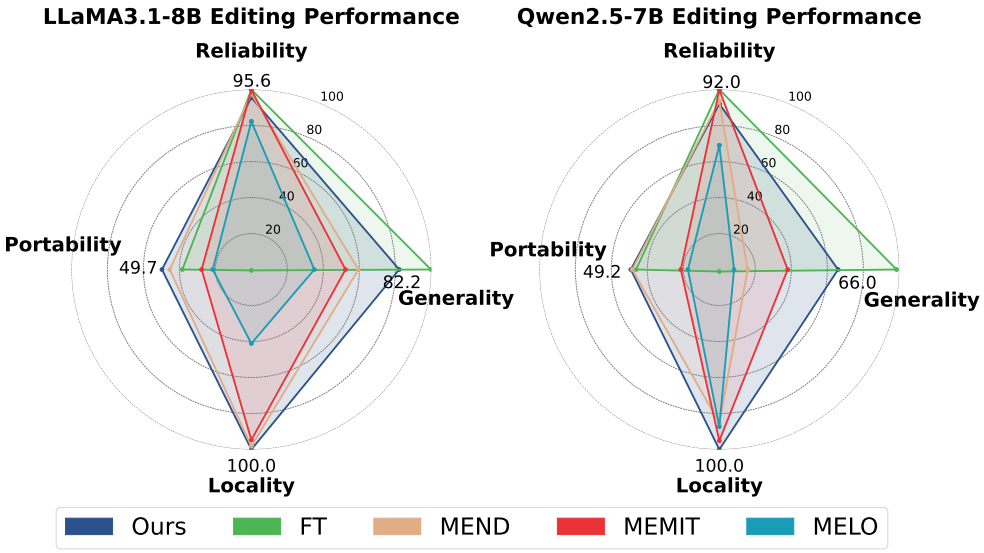

Figure 3: Editing results comparison across different editing methods on COUNTERFACT and MQuAKE-CF-v2 in a radar chart. The chart presents overall scores across four metrics: Reliability, Generality, Locality, and Portability, for the Llama 3.1 and Qwen2.5 models, respectively. Detailed results may be found in Appendix B.3.

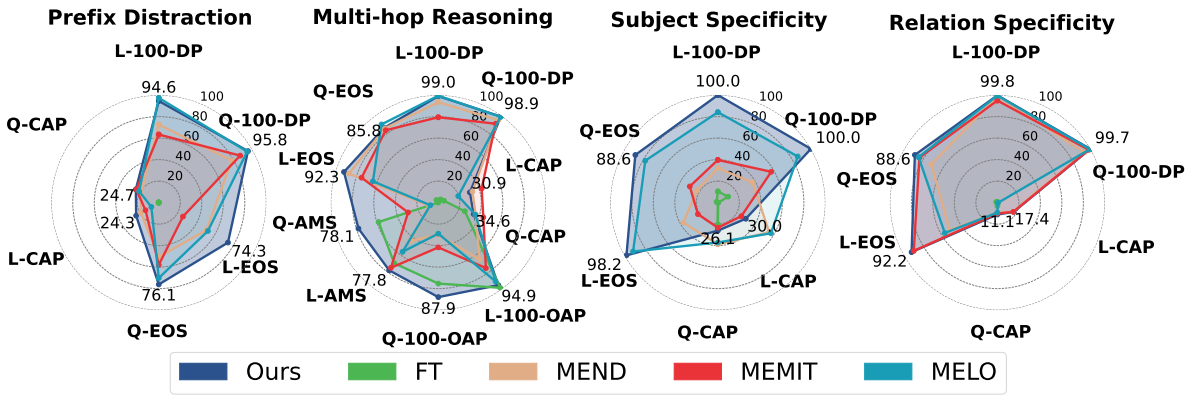

Figure 4: Editing Performance across different knowledge editing methods on EVOKE as depicted in the radar chart. All evaluation metrics range from 0 to 100. In the chart, values prefixed with "100-" denote the difference between the original metric value and 100. Results labeled with "L-" correspond to the Llama 3.1 model, and "Q-" to the Qwen 2.5 model. Detailed results can be found in Appendix B.3.

Correct Answer Probability (CAP) Measures the probability that the model generates the correct answer ans given a prompt. We define the CAP metric as:

$$\mathcal{M}_{\text{CAP}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{eval}}} \mathbb{P}(\text{ans} \mid \text{prompt})$$

where $\mathcal{D}_{\text{eval}}$ is the evaluation dataset.

Original Answer Probability (OAP) Evaluates the likelihood that the model continues to output the pre-edit answer ori, indicating potential resistance to modification. The metric is defined as:

$$\mathcal{M}_{\text{OAP}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{eval}}} \mathbb{P}(\text{ori} \mid \text{prompt})$$

Direct Probability (DP) Assesses the model's likelihood of producing the edited knowledge o^* when prompted, capturing its direct recall capability:

$$\mathcal{M}_{\text{DP}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{eval}}} \mathbb{P}(o^* \mid \text{prompt})$$

Editing Overfit Score (EOS) Evaluates whether the model overfits by favoring the edit target o^* over the correct answer ans. Formally, we define:

$$\mathcal{M}_{\text{EOS}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{eval}}} \mathbb{1}\left\{\mathbb{P}(\text{ans} \mid \text{p}) > \mathbb{P}(o^* \mid \text{p})\right\}$$

Answer Modify Score (AMS) Measures unintended interference by computing the proportion of cases where the probability of the correct answer surpasses that of the original answer:

$$\mathcal{M}_{\text{AMS}} = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{eval}}} \mathbb{1}\left\{\mathbb{P}(\text{ans} \mid \text{p}) > \mathbb{P}(\text{ori} \mid \text{p})\right\}$$

4 Experimental Results

We pre-trained the classifier on the COUNTERFACT-train dataset, as COUNTERFACT encompasses a wide range of knowledge triples (s, r, o) under various editing scenarios. This diversity ensures the classifier's generalizability to other datasets, thereby eliminating the need for retraining. Then, we trained **REACT** using the classifier on COUNTERFACT-train for the same reason. The details of the training process, including hyperparameter selection, optimizer configuration, and other relevant settings, are provided in Appendix B.2. The resulting weights were then employed to evaluate performance on the COUNTERFACT-edit, MQuAKE-v2, and EVOKE datasets, with the corresponding results presented in Tables 1 and 2.

4.1 COUNTERFACT and MQuAKE Results

Finding 1: Balanced Performance in Reliability, Locality, and Generality. Our method consistently demonstrates a well-balanced performance across the dimensions of reliability, locality, and generality. As evidenced by the radar chart 3 and high arithmetic average (*Score*) reported in Table 1, our approach effectively updates factual knowledge while maintaining uniform performance across these key metrics. This balance ensures that the model not only adapts to new information but also preserves the integrity of existing, unrelated knowledge.

Finding 2: Superior Portability Reflecting Robust Knowledge Editing. In addition to excelling in reliability, locality, and generality, our approach achieves notably high portability scores. Portability, which gauges the ability of the model to retain unaffected behavior following an edit, is a critical indicator of robust performance and resilience against overfitting. Compared to baseline methods, our framework shows significantly better portability, underscoring its capacity to implement targeted edits without compromising overall model functionality.

4.2 EVOKE Results

Finding 1: Significant Reduction in Overfitting. Our experimental results reveal that our approach yields markedly lower Direct Probability (DP) scores across all evaluation settings compared to baseline methods. In tasks such as Prefix Distraction, Multi-hop Reasoning, Subject Specificity, and Relation Specificity, the consistently reduced DP scores indicate that our method effectively avoids overfitting—i.e., it minimizes the inadvertent propagation of the edit target. Moreover, the corresponding high End-of-Sentence (EOS) and Answer Matching Scores (AMS) confirm that the overall output quality is preserved, reinforcing that our approach maintains a precise and targeted update without compromising the model's broader knowledge base.

Finding 2: Balanced Calibration Evident in CAP Scores. While our Correct Answer Probability (CAP) values are moderate relative to some baselines, this is not a shortcoming but rather a deliberate reflection of a cautious editing strategy. The moderate CAP scores indicate that our method deliberately refrains from overconfident updates, ensuring that only edits with sufficient certainty are applied. This balanced calibration is critical for preventing overfitting and for maintaining the stability of non-targeted knowledge, ultimately contributing to the robustness of our overall editing performance.

Finding 3: Superior Generalization Across Benchmarks. Despite being trained solely on the COUNTERFACT dataset, our method demonstrates exceptional generalization, consistently outperforming alternative approaches across diverse evaluation benchmarks. The robustness of our results—characterized by low DP scores paired with strong EOS and AMS metrics in multi-hop reasoning, subject specificity, and relation specificity tasks—provides compelling evidence that our approach generalizes effectively to various knowledge editing scenarios. This superior generalization underscores the potential of our method as a scalable and reliable solution for knowledge editing in large language models.

5 Related Work

LLM Knowledge Editing Knowledge editing has gained attention as an effective method for updating or correcting specific information within LLMs without requiring extensive retraining. Existing approaches can be broadly classified into two categories: parameter-preserving and parameter-modifying techniques. Parameter-preserving methods, such as SERAC (Mitchell et al., 2022b), maintain the model's existing parameters and instead leverage external memory or retrieval mech-

anisms to refine responses dynamically. In contrast, parameter-modifying methods directly adjust the internal weights of the model to embed new or corrected information. This category includes fine-tuning-based strategies like FT-L (Zhu et al., 2020), meta-learning approaches such as KE (De Cao et al., 2021b) and MEND (Mitchell et al., 2021), as well as structured intervention techniques that first localize and then edit knowledge representations, exemplified by MEMIT (Meng et al., 2022b). These methods provide varying levels of efficiency and precision, with locate-then-edit approaches offering more targeted modifications while preserving broader model behavior. The emergence of knowledge editing frameworks underscores the growing need for controllability and adaptability in modern LLMs, ensuring that their responses remain accurate and up-to-date without extensive retraining.

Representation Engineering Representation Engineering(Andy Zou, 2023) is derived as a novel approach that shifts the focus from neurons and circuits to high-level representations, enabling both monitoring and manipulation of cognitive functions in deep neural networks. Their work demonstrates that knowledge editing, along with other interventions such as truthfulness enforcement and memorization reduction, can be effectively implemented through representation control. Methods such as Linear Artificial Tomography (LAT) and Contrast Vectors allow for precise identification and modification of knowledge representations, aligning with prior efforts in mechanistic interpretability and concept erasure (Meng et al., 2023; Hernandez et al., 2023). This line of research complements existing strategies like causal tracing (Geva et al., 2022) and activation steering (Turner et al., 2023), which aim to localize and edit specific factual associations within neural networks. The emergence of RepE suggests that transparency-focused representation-based interventions can serve as an alternative to parameter-based fine-tuning, offering a more targeted and interpretable means of modifying LLM behavior.

6 Discussion and Conclusions

In this work, we present REACT, a dual-phase framework that overcomes overfitting in large language model editing by separating the process into (i) representation extraction and (ii) selective perturbation. It achieves balanced improvements in reliability, locality, and generality while preserving unrelated model behavior. By isolating a concise "belief shift" vector and applying controlled perturbations, REACT minimizes unintended side effects. Although the method requires careful parameter tuning and introduces extra computation, it offers a precise and interpretable approach for updating factual knowledge in LLMs while effectively overcoming overfitting.

7 Limitations

Though experiments prove that textbfREACT has great generalization ability from COUNTERFACT to other datasets, but the best way is to train the model on respective dataset to complete the task

References

Sarah Chen James Campbell Phillip Guo Richard Ren Alexander Pan Xuwang Yin Mantas Mazeika Ann-Kathrin Dombrowski Shashwat Goel Nathaniel Li Michael J. Byun Zifan Wang Alex Mallen Steven Basart Sanmi Koyejo Dawn Song Matt Fredrikson Zico Kolter Dan Hendrycks Andy Zou, Long Phan. 2023. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. 2023. Can we edit multimodal large language models? *arXiv preprint arXiv:2310.08475*.

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

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2022. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2203.14465*.

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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc

Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2306.04542*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36.

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

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

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

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

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

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.

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

Ziyang Xu, Haitian Zhong, Bingrui He, Xueying Wang, and Tianchi Lu. 2024. Ptransips: Identification of phosphorylation sites enhanced by protein plm embeddings. *IEEE Journal of Biomedical and Health Informatics*.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. 2024a. Uncovering overfitting in large language model editing. *Preprint*, arXiv:2410.07819.

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

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

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.

Chengrun Zhu, Hieu Pham, Zihang Dai, Chris Cundy, Sean Welleck, and Kyunghyun Cho. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Dataset Details

A.1 COUNTERFACT

The COUNTERFACT dataset is divided into three distinct subsets: a training set, a validation set, and an edit set (serving as an independent test set). The training set, validation set, and edit set contain 10,000 samples, 1,919 samples, and 10,000 samples, respectively. Each sample includes an original factual statement alongside its counterfactually revised variant, enabling systematic evaluation of models' sensitivity to subtle factual perturbations.

A.2 MQuAKE

The MQuAKE dataset comprises 3,000 samples, each encoded as a structured JSON object that encapsulates multiple layers of information pertinent to fact checking and counterfactual reasoning. Every sample contains detailed rewrite instructions, diverse composite questions, original and counterfactual answers (with aliases), concise single-hop Q&A pairs, and structured knowledge triples that document the factual revisions.

A.3 EVOKE

The EVOKE dataset is organized into two parts, "main" and "subj-spec" - comprising 1,031 and 458 samples, respectively. Each sample is represented as a JSON object containing detailed rewrite instructions with multiple prompt variations, portability information for alternative fact verifications, and prefix distractions, all designed to support rigorous evaluation of fact-checking and counterfactual reasoning tasks.

B Experiment Details

B.1 Version of edited LLMs

B.2 Experiment Resources and Parameters

In this study, we utilize an internal cluster equipped with the following resources: AMD EPYC 7763 CPUs, NVIDIA A100 80GB GPUs, and 512GB of RAM. The operating system is Ubuntu 20.04.6, and we employ PyTorch in our experiments.

The training of classifier took 12 GPU hours for each model on a single NVIDIA A100 80GB GPU, with total parameter number of 7.6B for Qwen-2.5 and 8.03B for Llama3.1.

The training of **REACT** took 40 GPU hours for each model on a single NVIDIA A100 80GB GPU, with total parameter number of 719M for Qwen-2.5 and 1.04B for Llama3.1.

B.2.1 REACT

Models	Iters	Edit Layer	Optimizer	LR
Llama3.1	20000	all layer of Transformer Module	Adam	$1e-5$
Qwen2.5	20000	all layer of Transformer Module	Adam	$1e-5$

$c_{edit} = 1, c_{loc} = 0.1$ for all models.

B.2.2 FT

Models	Steps	Edit Layer	Optimizer	LR
Llama3.1	25	layer $29, 30, 31$ of Transformer Module	Adam	$5e-4$
Qwen2.5	25	layer 27 of Transformer Module	Adam	$5e-4$

B.2.3 MEND

Models	MaxIter	Edit Layer	Optimizer	LR
Llama3.1	100000	layer $29, 30, 31$ of Transformer Module	Adam	$1e-6$
Qwen2.5	100000	layer $25, 26, 27$ of Transformer Module	Adam	$1e-6$

B.2.4 MEMIT

Models	mom sample	Edit Layer	kl factor
Llama3.1	3000	layer $4, 5, 6, 7, 8$ of Transformer Module	0.0625
Qwen2.5	3000	layer $4, 5, 6, 7, 8$ of Transformer Module	0.0625

B.2.5 MELO

Models	Radius	Edit Layer	edit per block	number of block
Llama3.1	75	layer $30, 31$ of Transformer Module	4	1500
Qwen2.5	75	layer $26, 27$ of Transformer Module	4	1500

B.3 Original experiment results

Model	Method	CounterFact-edit			MQuake-CF-V2 single-edit	Score
		Reliability↑	**Generality↑**	**Locality↑**	**Portability↑**	
Llama3.1 8B	Ours	95.58	82.17	**100.0**	**49.68**	**81.86**
	FT	**100**	**99.8**	0.49	38.38	59.67
	MEND	97.6	59.5	98.2	45.36	75.17
	MEMIT	99.8	52.3	94.7	27.63	68.61
	MELO	82.3	35.0	41.1	21.49	44.97
Qwen2.5 7B	Ours	92.0	66.0	**100**	**49.17**	
	FT	**100**	**98.5**	1.1	46.26	61.47
	MEND	93.7	15.8	85.3	48.38	60.80
	MEMIT	99.8	38.0	95.1	21.4	**63.58**
	MELO	69.0	8.2	87.3	17.45	45.49

Table 1: Editing results comparison across different knowledge-editing methods on Counterfact and MQuAKE-CF-v2. The best result for each metric is in **bold**, and the second best is underlined. The final "Score" column is the arithmetic mean of all metrics for that row.

	Prefix Distraction			Multi-hop Reasoning					Subject Specificity			Relation Specificity		
Editor	**DP↓**	**EOS↑**	**CAP↑**	**DP↓**	**CAP↑**	**OAP↓**	**AMS↑**	**EOS↑**	**DP↓**	**CAP↑**	**EOS↑**	**DP↓**	**CAP↑**	**EOS↑**
Llama3.1	5.44	**74.32**	**24.32**	0.96	30.87	5.06	**77.78**	**92.28**	**0**	30.02	**98.15**	**0.22**	**17.42**	**92.16**
FT	99.78	0	0	99.08	5.56	**2.03**	69.71	0.12	89.62	0.35	0	99.76	0	0
MEND	27.46	51.13	19.24	6.61	33.39	34.68	44.28	87.35	67.95	55.16	37.12	1.07	16.95	51.13
MEMIT	36.67	25.97	14.25	20.62	**42.42**	24.73	74.94	75.06	60.30	25.26	21.40	5.08	17.12	89.79
MELO	**2.57**	52.76	7.97	**0.58**	19.53	9.29	56.57	63.99	15.91	**57.04**	91.05	0.52	0.54	56.48
Qwen2.5	**4.19**	**76.11**	**24.66**	**1.11**	34.60	**12.09**	**78.09**	85.80	**0**	26.08	**88.64**	**0.26**	**11.06**	**88.64**
FT	99.73	0.15	0.33	96.28	25.94	24.69	58.39	2.92	88.94	20.26	1.31	99.25	3.05	1.22
MEND	21.64	50.49	18.87	5.17	36.53	70.03	9.00	85.16	62.62	**38.78**	22.49	6.47	9.42	71.03
MEMIT	12.57	57.93	24.16	9.29	**44.02**	58.33	29.56	83.21	42.65	23.33	30.13	1.81	10.14	83.70
MELO	5.02	70.18	21.12	1.35	36.29	71.13	7.79	**89.90**	14.17	37.06	77.95	0.69	9.30	84.65

Table 2: Editing results comparison across different knowledge-editing methods on EVOKE. The best result for each metric is in **bold**, and the second best is underlined.