Causal Path Alignment: Anchoring the Optimization Trajectory for Controllable In-Parameter Knowledge Editing

Xiyu Liu[1,2], **Zhengxiao Liu**[1,2*], **Naibin Gu**[1,2], **Zheng Lin**[1,2*] and **Weiping Wang**[1]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{liuxiyu, liuzhengxiao, gunaibin, linzheng, wangweiping}@iie.ac.cn

Abstract

Knowledge editing is pivotal for efficiently updating the parametric memory of Large Language Models (LLMs), enabling them to function as evolving agents in dynamic environments. However, mainstream in-parameter knowledge editing approaches suffer from Subject-Dominant Memory Interference: modifying a specific fact inadvertently corrupts the broader structural knowledge associated with the same subject within LLMs. We diagnose the root cause as a shortcut learning pathology, where the optimization objective overfits subject representations while bypassing the essential relational context. To rectify this, we propose Causal Path Alignment (CPA), a principled framework designed to anchor the optimization trajectory to valid causal pathways. CPA enforces parameter updates to route through relation-aware intermediate states, thereby preventing the erasure of contextual dependencies. Experimental results across diverse LLM backbones demonstrate that CPA consistently eliminates the shortcut, significantly improving relation specificity while exhibiting minimal side-effects. Moreover, CPA serves as a model-agnostic plug-in for existing editors, paving the way for reliable and trustworthy in-parameter knowledge editing.

1 Introduction

As Large Language Models increasingly serve as the cognitive core of autonomous agents [Minaee *et al.*, 2025; Luo *et al.*, 2025], the ability to maintain and update their parametric memory becomes paramount [Zhang *et al.*, 2024b; Wu *et al.*, 2025]. Unlike static models, an agent operating in a dynamic environment must continuously internalize new factual information [Hu *et al.*, 2026]. In this context, knowledge editing serves as the critical technique for the agent's parametric memory, aiming to efficiently update a small set of factual associations encoded in LLMs while exerting minimal side effects [Wang *et al.*, 2024b].

Mainstream in-parameter *locate-then-edit* methods have demonstrated impressive efficacy in rewriting specific factual associations [Zhang *et al.*, 2024a; Meng *et al.*, 2022a; Meng *et al.*, 2022b; Fang *et al.*, 2024; Li and Chu, 2025]. However, they suffer from a severe side effect that we term **Subject-Dominant Memory Interference**: modifying a specific attribute of a subject inadvertently corrupts the LLM's structural knowledge of the same subject [Liu *et al.*, 2025]. For instance, an edit targeting the fact that *Lionel Messi is a citizen of Germany* can lead to the severe corruption of his associated knowledge, causing the model to incorrectly assign him a hallucinatory nationality or alter his family relations (Fig. 1). Such uncontrollable over-generalization makes the edited model untrustworthy and unreliable.

While recent studies have begun to acknowledge this issue, existing solutions remain superficial. Prior attempts diagnose the issue statistically and primarily rely on one-sided or heuristic adjustments without diving into the root cause, leading to limited improvement or compromised performance [Liu *et al.*, 2025; Zhang *et al.*, 2025]. In this work, we attribute Subject-Dominant Memory Interference to a fundamental optimization pathology within the editing process. Through gradient saliency analysis and causal tracing, we reveal that standard editing objectives drive the model towards a form of shortcut learning [Geirhos *et al.*, 2020]: **Subject-Dominant Shortcut**. Specifically, during the optimization of the edited weights, the model exhibits a strong bias towards overfitting the high-magnitude representation of the *subject* (e.g., "Messi") while neglecting the *relational context* (e.g., "is a citizen of"). Consequently, the optimization trajectory bypasses the internal causal mechanism that gates knowledge retrieval based on relations, establishing a direct, unconditional mapping from the subject to the new target.

To rectify this, we propose **Causal Path Alignment (CPA)**, a principled framework designed to anchor the optimization trajectory to the model's valid causal pathways. Unlike previous heuristic patches, CPA reframes knowledge editing not merely as minimizing a loss function, but as navigating a constrained optimization landscape. Our approach operates in two stages: First, we anchor the relational context by identifying an intermediate hidden state that encodes the specific relation, independent of the target change. Second, we steer the parameter updates to strictly route through this anchored state. By forcing the information flow to traverse

*Zhengxiao Liu and Zheng Lin are corresponding authors.

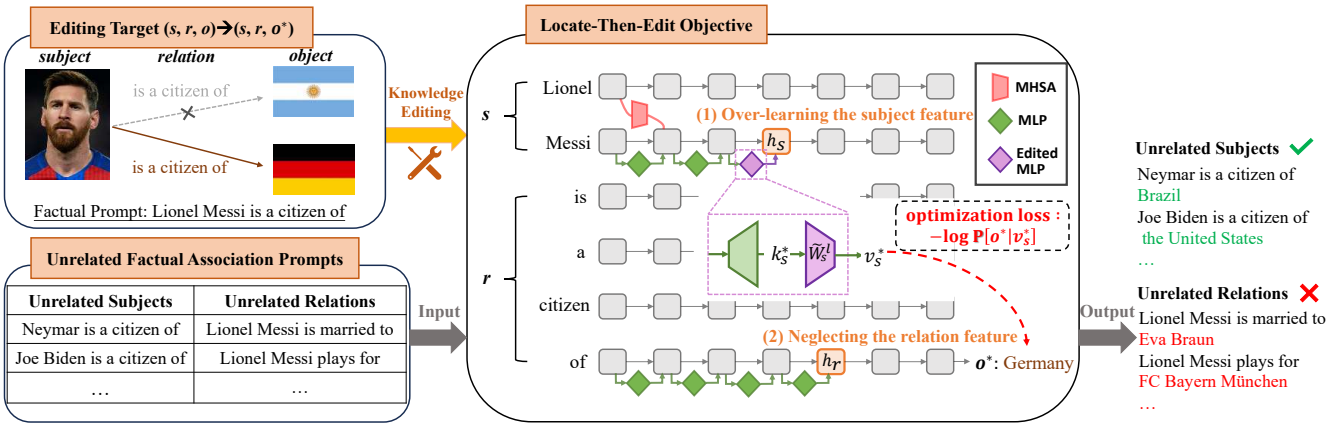

Figure 1: Subject-Dominant Memory Interference in Locate-then-Edit Knowledge Editing.

the correct relational feature, CPA re-engages the model's contextual gating mechanisms, ensuring that the update is specific to both the subject and the relation (Subject, Relation → Target) rather than the subject alone.

Our contributions are summarized as follows:

- We identify shortcut learning as the root cause of memory interference in knowledge editing, framing the problem as a misalignment between the optimization objective and the internal causal structure of LLMs.
- We introduce Causal Path Alignment (CPA), a two-phase optimization framework that explicitly anchors the editing process to relational features to eliminate the shortcut path.
- Comprehensive experiments on diverse LLM backbones and benchmarks demonstrate that CPA achieves optimal performance in relation specificity while exhibiting minimal side-effects. Crucially, CPA acts as a model-agnostic plug-in, seamlessly enhancing the reliability of existing methods, paving the way for controllable and trustworthy knowledge editing.

2 Related Works

2.1 Memory-based and Meta-learning Editing

Early approaches to knowledge editing often circumvent direct modification of the internal parameters of LLMs. One line of work adopts a *memory-based* strategy [Mitchell *et al.*, 2022b; Hartvigsen *et al.*, 2023; Wang *et al.*, 2024a], utilizing external caches to route inputs to edited counterparts. Another line employs *meta-learning* networks to predict weight updates via a trained hyper-network [De Cao *et al.*, 2021; Mitchell *et al.*, 2022a; Li *et al.*, 2025], enabling rapid adaptation. While effective at altering behavior, these methods function as external patches rather than updating the LLM's internal parametric worldview. This precludes deep integration and inevitably incurs overhead in inference latency and memory growth.

2.2 Locate-then-Edit Knowledge Editing

Locate-then-edit works achieve knowledge editing through knowledge neurons localization and the direct modification of located parameters. ROME [Meng *et al.*, 2022a] pioneers this by employing Causal Mediation Analysis to identify the decisive neuron activations for processing factual associations. They reveal that the Multi-Layer Perceptron (MLP) modules at the *last-subject token position* (i.e., the subject aggregation site) play a pivotal role in mediating the prediction of factual prompts. To scale this capability, MEMIT [Meng *et al.*, 2022b] distributes the update across multiple layers to enable batch editing, while subsequent works have extended this paradigm to lifelong editing scenarios [Fang *et al.*, 2024; Cai and Cao, 2024; Li and Chu, 2025], aiming to update parameters sequentially without catastrophic forgetting.

Despite their efficacy, these subject-focused editing methods suffer from a critical Subject-Dominant Memory Interference flaw. As mentioned by recent studies [Liu *et al.*, 2025; Zhang *et al.*, 2025], editing the subject aggregation site often inadvertently changes every fact linked to that subject, rendering the editing process uncontrollable. Current mitigation strategies primarily rely on heuristic modifications. For instance, RETS [Liu *et al.*, 2025] attempts to alleviate interference by relocating the edit target to the last-relation token with subject constraints, while LTI [Zhang *et al.*, 2025] introduces a data-augmentation approach, constraining the optimization neighborhood guided with enhanced samples. However, these approaches still omit the root cause. RETS suffers from a drastic degradation in generation fluency due to excessively stringent constraints, while LTI's reliance on data augmentation obtains limited improvement. In this work, we provide an in-depth exploration of the **optimization dynamics** that causes subject-focused editing to fail, and propose a principled optimization trajectory alignment to address this issue from the root cause.

3 Deconstructing the Optimization Pathology: The Subject-Dominant Shortcut

To understand the root cause of the uncontrollable over-generalization observed in locate-then-edit methods, we conduct a rigorous investigation into the optimization dynamics of ROME-like approaches. We propose that the failure is not due to the efficacy of the parameter update itself, but rather

a *structural pathology* in how the target representation is derived.

3.1 Preliminaries: The Locate-then-Edit Objective

Knowledge editing aims to alter factual associations within an auto-regressive language model $\mathcal{M}$. Given a fact triplet $\langle s, r, o \rangle$, where s is the subject, r is the relation, and o is the object, the goal is to update the model's prediction to a new target $\langle s, r, o^* \rangle$ while minimizing interference with unrelated knowledge $\langle s, r', o' \rangle$ or $\langle s', r, o' \rangle$.

Locate-then-edit methods operate under the *Linear Associative Memory* hypothesis [Geva *et al.*, 2022; Geva *et al.*, 2023]. They identify specific mid-to-early MLP layers (e.g., layer l) at the last subject token index as the decisive location for recalling subject attributes. The down-projection matrix W_s^l of layer l can be treated as a set of associative key-value memories that store the map between input key vectors $K = [\ k_1 \mid k_2 \mid k_3 \mid ...\ ]$ to output value vectors $V = [\ v_1 \mid v_2 \mid v_3 \mid ...\ ]$. A specific fact is edited by injecting the corresponding (k_s^*, v_s^*) pair into the associative memories. The editing process is decomposed into two distinct steps: (1) determining the *optimal target vector* v_s^* that encodes the new fact, and (2) updating the weight matrix W_s^l to map the subject's key k_s^* to this new value v_s^*.

While the second step (i.e., the weight update) is a solved constrained least-squares problem (often yielding a rank-one update $\tilde{W}_s^l = W_s^l + \Delta W$), our investigation identifies the first step as the source of the subject-dominant memory interference issue. The target vector v_s^* is obtained by optimizing the latent representation v to maximize the probability of the target object o^* given the prompt. Formally, $v_s^* = argmin_v \mathcal{L}(v)$ is obtained as the minimizer of the following loss function:

$$\mathcal{L}(v) = -\frac{1}{N}\sum_{t=1}^{N} \log P_{\mathcal{M}}(o^* \mid h_L(v; p_t)) + \mathrm{KL}(v, v_{old}) \quad (1)$$

Here, $P_{\mathcal{M}}(o^* \mid h_L(v; p_t))$ denotes the prediction probability for o^* when the hidden state at the edit layer is intervened with vector v, given a set of prefixed prompts p_t. The KL divergence term $\mathrm{KL}(v, v_{old}) = \mathcal{D}_{\mathrm{KL}}(P_{\mathcal{M}}(\cdot|v) \| P_{\mathcal{M}}(\cdot|v_{old}))$ acts as a regularizer to constrain the distribution shift of v from the original output vector of W_s^l.

Crucially, this optimization process assumes that finding a vector v that satisfies the likelihood of o^* is sufficient for robust knowledge editing. However, as we demonstrate in the subsequent analysis, this objective function creates a **Subject-Dominant Shortcut**: it encourages the vector v_s^* to over-encode the mapping to o^* directly, effectively bypassing the relational computational path inherent in the inference mechanism of LLMs.

3.2 Gradient Analysis: The Divergence of Information Pathways

To characterize the feature prioritization inherent in the optimization of the target vector v_s^*, we analyze the gradient landscape of the loss function $\mathcal{L}(v)$ with respect to the model's internal representations. Since ROME introduces perturbations at a specific mid-early layer l, we track the back-propagated gradients through the hidden states of subsequent layers. Formally, we define the *Gradient Saliency* S for a hidden state $h_{i,\ell}$ at token position i and layer $\ell > l$ as the L_2-norm of the gradient vector:

$$S(h_{i,\ell}) = \left\| \nabla_{h_{i,\ell}} \mathcal{L}(v) \right\|_2 \quad (2)$$

We focus particularly on the *last subject token* (p_{ls}) and the *last relation token* (p_{lr})[1], as prior interpretability studies suggest these positions serve as aggregation hubs for subject attributes [Geva *et al.*, 2023] and relational attributes [Liu *et al.*, 2025], respectively.

Figure 2 visualizes the average gradient saliency over 100 factual edits on GPT2-XL 1.5B (48 layers) [Radford *et al.*, 2019] and Qwen2.5 7B (28 layers) [Qwen *et al.*, 2025], excluding the final output token to highlight internal dynamics. The visualization reveals a striking **Dual-Peak Pattern** where gradients are significantly active at two distinct loci across the layers. Specifically, we observe high gradient norms at the **Subject Pathway (p_{ls})**, indicating the model leverages entity representations to adjust the prediction, alongside strong gradient flows at the **Relation Pathway (p_{lr})**, which confirms that the pre-trained weights initially attempt to utilize the relational context to mediate the prediction of the target object. **This empirical evidence suggests that the optimization landscape naturally presents two competing pathways for minimizing the loss: modifying the subject representation or leveraging the relation representation**. The existence of the relation pathway in the gradient map proves that the model *can* theoretically attend to the relation. However, as we unveil in the subsequent section, the standard optimization objective fails to sustain this balance, causing the optimization trajectory to collapse onto a shortcut.

3.3 The Shortcut Mechanism: Causal Path Collapse

While the gradient analysis in Section 3.2 reveals that the optimization landscape *initially* offers two viable pathways (subject and relation), it remains unclear how the edited model actually utilizes these features for inference. To investigate the functional shift in the model's internal mechanism, we employ Causal Tracing [Meng *et al.*, 2022a], a technique that measures the contribution of specific hidden states to the final prediction by intervening on corrupted runs. Specifically, we calculate the Indirect Effect (**IE**) of an MLP output $m_i^{l_j}$ (at token position i and layer l_j) by restoring its clean activation into a noise-corrupted forward pass. Formally, the IE for the target object o is defined as $\mathbf{IE}(o, \theta, m_i^{l_j}) = P_\theta(o \mid p', m_i^{l_j}) - P_\theta(o \mid p')$, where p' is the prompt with noise-corrupted embeddings.

To quantify the structural change induced by editing, we introduce the **Ratio of Indirect Effect (RIE)**, which measures

[1]The token positions are categorized into relation prefix ("rp"), first subject ("fs"), middle subject ("ms"), last subject ("ls"), first relation ("fr"), middle relation ("mr") and last relation ("lr") according to their positions in the tokenized subject or relation of a factual association.

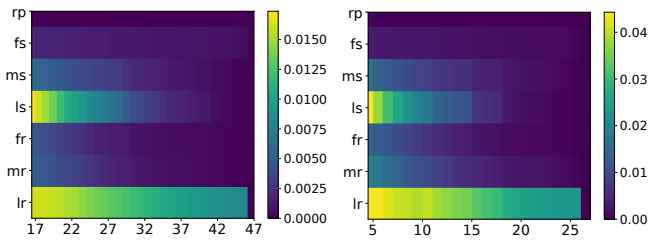

Figure 2: The average gradient saliency map during the first epoch of optimization for v_s^*. The heatmap visualizes gradient magnitudes across layers (x-axis) and token positions (y-axis). A distinct **Dual-Peak Pattern** is observed at the last subject token (ls) and the last relation token (lr). Left: GPT2-XL (1.5B). Right: Qwen2.5 (7B).

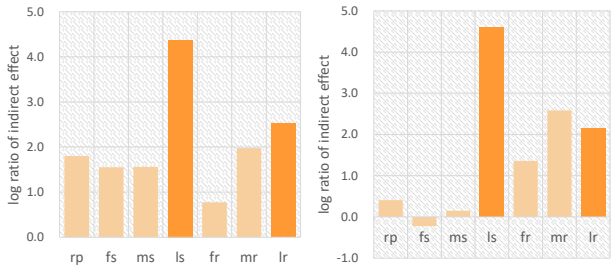

Figure 3: Maximum RIE of MLP outputs across layers. The x-axis represents token positions. A dramatic surge in RIE is observed at the last subject position (ls), contrasting with the raise at the last relation position (lr). Left: GPT2-XL (1.5B). Right: Qwen2.5 (7B).

the logarithmic shift in causal reliance between the original model θ_o and the edited model θ_{edited}. The RIE for a specific activation is calculated as:

$$\textbf{RIE}(m_i^{l_j}) = \log \frac{\textbf{IE}(o^*, \theta_{\text{edited}}, m_i^{l_j})}{\textbf{IE}(o, \theta_o, m_i^{l_j})} \tag{3}$$

This metric isolates the relative gain or loss in causal significance for each component. We analyze the maximum RIE of MLP outputs across all layers at each token position over 100 factual edits. Figure 3 presents these results, with a detailed case study provided in the Appendix.

The empirical results reveal a critical pathology: the causal contribution of the *subject representation* (h_s) increases exponentially after editing, while the *relation representation* (h_r) shows much less gain. This suggests that the optimization process has decoupled the prediction from the relational context. We formalize this phenomenon as follows:

Definition 1 (Subject-Dominant Shortcut). *Given a factual association $\langle s, r \rangle \rightarrow o^*$, an editing process converges to a Subject-Dominant Shortcut if the learned conditional probability distribution P_{θ^*} exhibits approximate conditional independence of the object o^* from the relation r, given the subject s:*

$$P_{\theta^*}(o^* \mid s, r) \approx P_{\theta^*}(o^* \mid s) \tag{4}$$

This degeneracy manifests structurally when the causal influence of the subject representation dominates the computation, effectively marginalizing the relational operator such that $\textbf{IE}(h_s) \gg \textbf{IE}(h_r)$.

This definition implies a causal path collapse. In a healthy model, the prediction follows a compositional path $f(s, r)$. **However, existing optimization forces the subject vector v_s to absorb the target information o^* directly**. Mathematically, the optimizer finds a solution where the mapping approximates $v_s \rightarrow o^*$, bypassing the relational gate. Consequently, when queried with the same subject but a different relation (e.g., $\langle s, r' \rangle$), the dominant subject pathway triggers the retrieval of o^*, resulting in the severe subject-dominant memory interference (i.e., poor relation specificity) observed in existing methods.

4 Eliminating Shortcut Learning via Causal Path Alignment

Having identified the collapse of the causal path as the root cause of subject-dominant memory interference, we propose a principled optimization framework to rectify this pathology. The standard editing objective, which minimizes the negative log-likelihood $-\log P(o^*|v_s)$, is fundamentally *under-constrained*: it demands only that the target o^* be predicted, without specifying *how*. Given the pre-existing dominance of subject features, the optimizer inevitably exploits the direct shortcut $v_s \rightarrow o^*$. To dismantle this shortcut, we must explicitly regularize the optimization trajectory, forcing the information flow to traverse the relational computational path. We term this approach **Causal Path Alignment (CPA)**, which effectively realigns the model's internal causal mechanism.

4.1 Theoretical Framework: Causal Path Alignment via Anchoring

Our strategy hinges on anchoring the prediction to the *relation feature*. Let h_r denote the hidden state at the last relation token position in a layer l_a ($l_a > l$) subsequent to the edit layer. In the standard forward pass of a LLM, the subject injection v_s propagates through intermediate layers to form h_r, which then dictates the final prediction [Geva *et al.*, 2023]. Formally, this dependency can be modeled as a Markov chain: $v_s \rightarrow \cdots \rightarrow h_r \rightarrow \cdots \rightarrow o^*$. Ideally, to ensure the relation r mediates the prediction, we should maximize the joint likelihood $P(o^*, h_r \mid v_s)$. By applying the chain rule of probability, we decompose this objective into:

$$P(o^*, h_r \mid v_s) = \underbrace{P(o^* \mid h_r, v_s)}_{\text{Prediction Head}} \cdot \underbrace{P(h_r \mid v_s)}_{\text{Feature Construction}} \tag{5}$$

The first term, $P(o^* \mid h_r, v_s)$, represents the probability of the target object given both the relation feature and the subject injection. The structural flaw in current locate-then-edit editing is that the model learns to rely on v_s directly to predict o^*, rendering h_r redundant (Sec. 3). To eliminate this shortcut, we introduce a crucial **Conditional Independence Constraint** that we enforce the prediction of the object o^* should depend *solely* on the relation feature h_r, making it independent of the specific subject injection v_s given h_r. Mathematically, this imposes $o^* \perp v_s \mid h_r$, which simplifies the first Prediction Head term in Eq. 5 to $P(o^* \mid h_r)$.

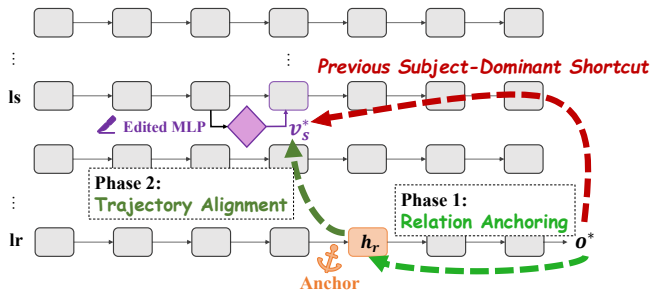

Figure 4: Our Causal Path Alignment (CPA) eliminates subject-dominant shortcut with a two-phase optimization framework.

Consequently, our refined optimization objective maximizes the decomposed likelihood:

$$\mathcal{L}_{\text{CPA}} \approx \underbrace{\log P(o^* \mid h_r)}_{\text{Relation Anchoring}} + \underbrace{\log P(h_r \mid v_s)}_{\text{Trajectory Alignment}} \tag{6}$$

This decomposition provides the theoretical foundation for our two-stage optimization process. The first term, representing *Relation Anchoring*, establishes a valid mapping from the relation feature to the target object, creating a "virtual anchor" in the activation space that encodes relation semantics decoupled from the subject. The second term, representing *Trajectory Alignment*, ensures that the edited subject vector v_s correctly reconstructs this pivotal relation feature. By optimizing these terms sequentially, we force the optimization path to be anchored to the intermediate state h_r, thereby strictly prohibiting the bypass of relational semantics.

4.2 Implementation: Two-Phase Optimization

We implement CPA as a sequential optimization process that anchors the update trajectory to the causal structure derived in Eq. 6. As shown in Fig. 4, we sequentially optimize the decomposed likelihood terms to enforce relational mediation.

Phase 1: Relation Anchoring. We first optimize a relation anchor h_r^* by maximizing $\log P(o^* \mid h_r)$ at the bottleneck layer l_a. This shifts the prediction to o^* while operating solely in the activation space to avoid parameter overfitting. We solve $h_r^* = \text{argmin}_{h_r} \mathcal{L}_1(h_r)$ via:

$$\mathcal{L}_1(h_r) = -\frac{1}{N}\sum_{t=1}^{N} \log P[o^*|h_r, p_t] + \mathcal{D}_{\text{KL}}[P(\cdot|h_r)||P(\cdot|h_r^o)] \tag{7}$$

The KL divergence against the pre-edit state h_r^o confines the anchor to the local semantic manifold, ensuring the edited relation remains distinguishable from unrelated facts.

Phase 2: Trajectory Alignment. Fixing h_r^*, we maximize $\log P(h_r \mid v_s)$ by optimizing the subject injection vector v_s (the output of the MLP down-projection at the editing layer l) to accurately reconstruct the relation anchor. The updated vector v_s^* is obtained by minimizing the discrepancy in the feature space:

$$\mathcal{L}_2(v_s) = ||\mathcal{F}[v_s, p] - h_r^*||_F \tag{8}$$

where $\mathcal{F}[v_s, p]$ is the forward propagation result in h_r given v_s and the prompt p. By enforcing v_s to reconstruct h_r^*, this step compels the vector update to traverse the established causal path, effectively eliminating the shortcut learning pathology. All weight decays are omitted for simplicity.

5 Experiments

5.1 Experimental Setup

Datasets and Evaluation Metrics. We evaluate CPA under the standard single-edit protocol, where each run rewrites one factual association $\langle s, r, o\rangle \rightarrow \langle s, r, o^*\rangle$ and quantifies both the success of the rewrite and the unintended spillover to other memories.

Our primary benchmark is **COUNTERFACT_RS** [Liu *et al.*, 2025], a relation-sensitive evaluation suite derived from COUNTERFACT, on which we sample 2,000 edits and repeat evaluation five times, reporting mean and standard deviation. COUNTERFACT_RS provides a controlled decomposition of editing quality into *Efficacy* (whether the rewritten object is preferred under rewriting prompts), *Subject Specificity* (S-SPEC., whether the new object spuriously transfers to neighborhood subjects $\langle s', r\rangle$), and *Relation Specificity* (R-SPEC., whether the new object leaks to other relations of the same subject $\langle s, r'\rangle$). Because our goal is to eliminate Subject-Dominant Memory Interference, we treat R-SPEC. as the key controllability metric, while also reporting a Fluency score that captures the quality of generated continuations under the benchmark protocol.

To complement the evaluation with a broader subject-dominant memory interference assessment, we also evaluate on the **EVOKE** [Zhang *et al.*, 2025] benchmark and report corresponding metrics: DP (Deterioration Probability) to measure how often unrelated answers are disrupted after editing, CAP (Correct Answer Probability) to track overall correctness, and EOS (Editing Overfit Score) to capture over-confident collapse, serving as a primary indicator of the model's specificity and overall performance; lower DP and higher EOS indicate stronger resistance to interference.

Baselines and Model Architectures. We compare against representative locate-then-edit editors, including **ROME** [Meng *et al.*, 2022a], **MEMIT** [Meng *et al.*, 2022b], and **AlphaEdit** [Fang *et al.*, 2024], as well as recent mitigation-oriented variants such as **RETS** [Liu *et al.*, 2025] and **ROME-LTI** [Zhang *et al.*, 2025]. Experiments span multiple LLM families and scales, including GPT2-XL (1.5B), GPT-J (6B) [Wang and Komatsuzaki, 2021], and Qwen2.5 (7B/14B). In particular, we report the main results for **ROME-CPA** because ROME is the representative and most widely adopted single-edit backbone, and it cleanly decomposes editing into (i) optimizing the target value representation and (ii) a closed-form rank-one weight update at the located MLP module. This decomposition makes the shortcut pathology and our trajectory-alignment intervention directly comparable. We nevertheless verify portability by transplanting CPA to other locate-then-edit methods in subsequent experiments.

Implementation Details. For each backbone, we follow the commonly adopted layer choices for locate-then-edit updates and select a later bottleneck layer $l_a > l$ for relation

Method	COUNTERFACT_RS				EVOKE		
	Eff.↑	S-Spec.↑	R-Spec.↑	Flue.↑	DP↓	CAP↑	EOS↑
GPT2-XL (1.5B)							
ROME	99.8(±0.0)	75.4(±0.0)	45.7(±0.2)	623.4(±0.0)	39.1(±0.3)	2.8(±0.1)	21.6(±0.2)
MEMIT	94.0(±0.1)	76.4(±0.0)	61.5(±0.5)	**627.4**(±0.2)	38.0(±1.0)	1.9(±0.1)	10.9(±0.3)
AlphaEdit	99.8(±0.0)	76.2(±0.0)	41.2(±0.3)	623.4(±0.1)	34.5(±0.2)	2.1(±0.0)	15.5(±0.2)
RETS	**100.0**(±0.0)	67.9(±0.4)	**78.7**(±0.5)	585.1(±5.9)	**5.2**(±0.8)	**4.8**(±0.0)	**61.4**(±2.0)
ROME-LTI	99.9(±0.0)	75.7(±0.1)	57.2(±0.3)	622.0(±0.2)	17.7(±0.2)	3.6(±0.1)	31.4(±0.4)
ROME-CPA (Ours)	98.7(±0.1)	**76.5**(±0.0)	72.2(±0.7)	619.7(±0.3)	5.7(±0.1)	2.5(±0.1)	36.5(±0.1)
GPT-J (6B)							
ROME	**100.0**(±0.0)	79.3(±0.0)	52.8(±0.4)	621.0(±0.2)	53.1(±0.4)	1.9(±0.0)	6.1(±0.1)
MEMIT	**100.0**(±0.0)	81.1(±0.0)	66.8(±0.3)	**621.5**(±0.1)	38.8(±0.3)	2.2(±0.1)	12.3(±0.2)
AlphaEdit	99.8(±0.1)	81.9(±0.0)	60.4(±0.2)	621.4(±0.1)	38.2(±0.2)	2.5(±0.0)	14.8(±0.1)
RETS	**100.0**(±0.0)	61.4(±1.9)	81.5(±0.7)	530.6(±17.1)	30.1(±0.6)	**4.2**(±0.2)	30.7(±0.5)
ROME-LTI	99.8(±0.0)	80.7(±0.1)	82.5(±0.3)	618.8(±0.1)	9.0(±0.5)	3.5(±0.1)	32.5(±0.5)
ROME-CPA (Ours)	99.6(±0.0)	**82.0**(±0.0)	**83.5**(±0.3)	618.6(±0.2)	**7.8**(±0.7)	3.1(±0.1)	**37.3**(±1.3)
Qwen2.5 (7B)							
ROME	99.8(±0.0)	82.2(±0.0)	70.7(±0.2)	625.5(±0.1)	16.6(±0.2)	9.3(±0.1)	42.6(±0.3)
MEMIT	**100.0**(±0.0)	67.3(±0.2)	64.7(±0.5)	624.4(±0.1)	12.2(±0.3)	11.3(±0.1)	52.6(±0.4)
AlphaEdit	**100.0**(±0.0)	83.6(±0.0)	65.0(±0.7)	**626.3**(±0.1)	20.8(±0.2)	8.0(±0.1)	33.6(±0.2)
RETS	99.5(±0.1)	42.7(±0.4)	52.9(±0.3)	540.3(±7.4)	10.2(±0.1)	**14.1**(±0.2)	70.5(±0.6)
ROME-LTI	99.5(±0.1)	83.7(±0.3)	84.6(±0.1)	624.5(±0.3)	4.5(±0.1)	6.6(±0.1)	52.8(±0.5)
ROME-CPA (Ours)	96.2(±0.1)	**84.0**(±0.5)	**92.2**(±0.4)	623.9(±0.2)	**0.8**(±0.0)	8.5(±0.1)	**76.2**(±0.1)
Qwen2.5 (14B)							
ROME	**100.0**(±0.0)	61.3(±0.1)	56.9(±0.3)	624.0(±0.1)	18.1(±0.4)	9.4(±1.1)	45.3(±0.6)
MEMIT	**100.0**(±0.0)	79.8(±0.0)	66.7(±0.2)	**625.4**(±0.1)	16.9(±0.2)	**15.3**(±0.1)	48.9(±0.3)
RETS	92.0(±0.0)	73.6(±1.1)	83.2(±0.5)	591.5(±6.3)	33.8(±0.4)	7.9(±0.1)	29.0(±0.2)
ROME-LTI	**100.0**(±0.0)	70.5(±0.2)	78.1(±0.1)	623.4(±0.3)	16.4(±0.3)	11.6(±0.1)	59.8(±0.4)
ROME-CPA (Ours)	99.5(±0.1)	**83.4**(±0.5)	**84.5**(±0.4)	620.5(±0.2)	**6.9**(±0.2)	12.7(±0.6)	**62.8**(±0.3)

Table 1: Performance comparison on COUNTERFACT_RS and EVOKE datasets. We explicitly focus on the Subject-Dominant Memory Interference of locate-then-edit knowledge editing, which is mainly measured by R-Specificity (R-Spec.) on COUNTERFACT_RS, and by DP, CAP, and EOS on EVOKE. Red values indicate severe low values for COUNTERFACT_RS. Standard deviations are in parentheses. **Bold** indicates the best result, and underline indicates the second-best result. Our CPA method effectively mitigates interference while achieving least side-effects across benchmarks. All metrics except Fluency are reported in percentage (%).

anchoring. Concretely, for 48-layer models we edit the 18th and 19th MLP layers and anchor at $l_a \in \{19, 20\}$ (GPT2-XL) or $l_a = 28$ (Qwen2.5 14B), while for 28-layer models we edit layer 6 and anchor at $l_a = 11$ (GPT-J and Qwen2.5 7B). More details can be referred to the Appendix.

5.2 Main Results

Table 1 summarizes the results on COUNTERFACT_RS and EVOKE across multiple LLM backbones.

CounterFact_RS. Across all evaluated models, CPA consistently improves relation specificity (R-Spec.), indicating a substantial reduction of Subject-Dominant Memory Interference. Concretely, compared with ROME, CPA raises R-Spec. from 45.7% to 72.2% on GPT2-XL, from 52.8% to 83.5% on GPT-J, from 70.7% to 92.2% on Qwen2.5 7B, and from 56.9% to 84.5% on Qwen2.5 14B, while maintaining high editing success (Eff.) and strong subject specificity (S-Spec.). Notably, this gain does not come from overly restrictive constraints: fluency remains comparable to ROME/MEMIT, whereas methods with aggressive constraints (e.g., RETS) can exhibit pronounced fluency degradation and less stable cross-model behavior.

EVOKE. EVOKE directly probes collateral corruption and shortcut-induced overfitting. CPA achieves the lowest (or second-lowest) damage proxy DP and the best EOS on most settings, suggesting that the edited behavior is less dominated by a subject-only shortcut and induces fewer unintended changes. While RETS can reach strong CAP/EOS on some settings, it often does so with a notable trade-off in fluency and/or robustness; in contrast, CPA provides a more balanced improvement profile across architectures.

Overall, the main results demonstrate that CPA provides a reliable remedy for Subject-Dominant Memory Interference across both benchmarks and model families. CPA consistently yields large gains in relation specificity and further reduces detailed damage indicators on EVOKE, while preserving (and often saturating) editing efficacy and maintaining competitive fluency. These findings support our central claim: anchoring the optimization to a relation-aware causal

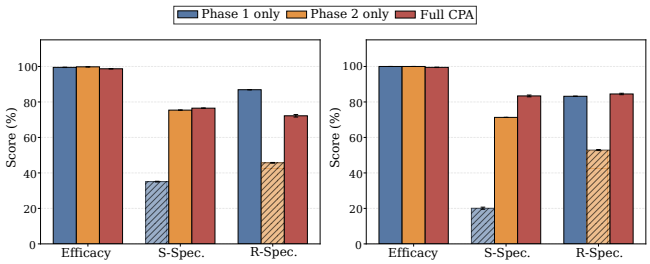

Figure 5: Ablation study of ROME-CPA on COUNTERFACT_RS. Left: GPT2-XL (1.5B). Right: Qwen2.5 (14B).

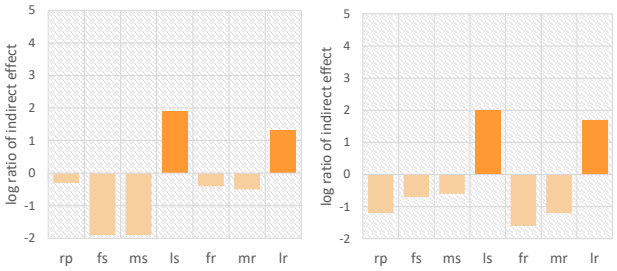

Figure 6: Mechanistic validation of eliminating the Subject Dominant Shortcut. Left: GPT2-XL (1.5B). Right: Qwen2.5 (7B).

bottleneck and aligning the update trajectory to reconstruct this anchor prevents causal path collapse, enabling edits that are both effective and controllable.

5.3 Mechanistic Validation

Beyond aggregate metrics, we verify that CPA indeed restores the relational causal pathway rather than merely suppressing outputs. Following Section 3.3, we re-run causal tracing and compute the maximum Ratio of Indirect Effect (RIE) over layers for each token position (Fig. 6). The key signature of shortcut elimination is that the relative causal gain at the last relation position becomes closer to that at the last subject position, indicating that the edited prediction is mediated by relation-aware features instead of a subject-only bypass. This mechanistic evidence aligns with the consistent improvements in relation specificity and the reductions in EVOKE side-effect indicators.

5.4 Ablation Study

We conduct ablation experiments on 2,000 edits from COUNTERFACT_RS to disentangle the contribution of each phase in **Causal Path Alignment (CPA)**. We compare three configurations under identical prompts, stopping criteria, and hyperparameters: **(1) Phase 1 only** performs *relation anchoring* by optimizing the bottleneck activation h_r at the last-relation token position to minimize $-\log P(o^* \mid h_r)$, but omits the subsequent trajectory alignment; **(2) Phase 2 only** reduces to the conventional single-phase target construction by directly optimizing the injected subject value vector v_s at the last-subject token position with the standard objective $-\log P(o^* \mid v_s)$; and **(3) Full CPA** executes the complete two-phase procedure (relation anchoring followed by trajectory alignment).

Model	Method	Eff.↑	S-Spec.↑	R-Spec.↑
GPT2-XL (1.5B)	MEMIT	94.0	76.4	61.5
	+ **CPA**	92.8	76.3	**79.7** (+18.2)
	AlphaEdit	99.8	76.2	41.2
	+ **CPA**	97.7	76.2	**68.4** (+27.2)
Qwen2.5 (7B)	MEMIT	100.0	67.3	64.7
	+ **CPA**	93.8	82.7	**90.6** (+25.9)
	AlphaEdit	100.0	83.6	65.0
	+ **CPA**	92.0	82.5	**88.9** (+23.9)

Table 2: Plug-and-Play generalization of CPA on MEMIT and AlphaEdit. All metrics are reported in percentage (%).

Results in Fig. 5 confirm the synergistic necessity of both phases. We observe a distinct trade-off: **Phase 1-only** secures relation specificity but suffers from catastrophic subject over-generalization (S-Spec. drops to ∼20–35%), whereas **Phase 2-only** restores subject focus but reverts to the shortcut pathology, degrading relation specificity to ∼46%. Only the full CPA reconciles these conflicting objectives, achieving an optimal balance (e.g., Qwen2.5: 83.4% S-Spec., 84.5% R-Spec.) while maintaining near-saturated efficacy.

5.5 Plug-and-Play Generalization to Other Editors

CPA is editor-agnostic: it constrains *how* the target representation is optimized, and thus can be transplanted to other locate-then-edit pipelines without altering their localization or weight-update solvers. To test this, we integrate CPA into MEMIT and AlphaEdit by replacing their original target-vector optimization with CPA while keeping their original layer selections and update rules. As shown in Table 2, CPA substantially improves R-Spec. for both editors on GPT2-XL and Qwen2.5 7B (e.g., MEMIT: 61.5%→79.7% and 64.7%→90.6%; AlphaEdit: 41.2%→68.4% and 65.0%→88.9%), while largely preserving efficacy and S-Spec. This indicates that the shortcut pathology stems from the unconstrained target-vector optimization, and CPA provides a general remedy by anchoring the optimization trajectory to relation-aware causal states.

6 Conclusion

In this work, we diagnose the *Subject-Dominant Shortcut*—a pathology in locate-then-edit methods where optimization decouples predictions from relational context, causing severe subject-dominant memory interference. To rectify this, we propose Causal Path Alignment (CPA), a framework that anchors parameter updates to valid internal causal pathways. By enforcing a conditional independence constraint via a two-stage process, CPA ensures updates are mediated by relational features rather than unconditional subject mappings. Experiments across diverse LLMs demonstrate that CPA significantly boosts relational specificity without compromising efficacy. As a model-agnostic plug-in, CPA effectively resolves the plasticity-stability trade-off, paving the way for trustworthy parametric memory updates in evolving AI agents.

Ethical Statement

Our research focuses on addressing the challenge of outdated or erroneous knowledge encoded in large language models by developing methods to detect and correct such inaccuracies. At the same time, we are acutely aware of the ethical risks this technology entails, particularly the possibility of misuse for disseminating harmful or misleading content. To mitigate these risks, we underscore the critical importance of two key principles: first, language models must be obtained from rigorously vetted and reliable sources; second, users must exercise utmost caution and critical judgment when utilizing outputs generated by these models.

In the preparation of this manuscript, AI-assisted tools (including large language models) are employed strictly for the purpose of polishing and textual refinement to enhance clarity and organization. The results have been further controlled and edited. We retain full responsibility for the core scientific questions, methodological design, theoretical innovation, experimental analysis, and all conclusions presented. AI plays no role in the creative scientific conceptualization process.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 62472419, 62472420) and the Enterprise Project (No. E4V06811F3).

References

[Cai and Cao, 2024] Yuchen Cai and Ding Cao. O-edit: Orthogonal subspace editing for language model sequential editing, 2024.

[De Cao *et al.*, 2021] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[Fang *et al.*, 2024] Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Xiang Wang, Xiangnan He, and Tat seng Chua. Alphaedit: Null-space constrained knowledge editing for language models, 2024.

[Geirhos *et al.*, 2020] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020.

[Geva *et al.*, 2022] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[Geva *et al.*, 2023] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore, December 2023. Association for Computational Linguistics.

[Hartvigsen *et al.*, 2023] Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors, 2023.

[Hu *et al.*, 2026] Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, Zhenrong Cheng, Xuanbo Fan, Jiaxin Guo, Xinlei Yu, Zhenhong Zhou, Zewen Hu, Jiahao Huo, Junhao Wang, Yuwei Niu, Yu Wang, Zhenfei Yin, Xiaobin Hu, Yue Liao, Qiankun Li, Kun Wang, Wangchunshu Zhou, Yixin Liu, Dawei Cheng, Qi Zhang, Tao Gui, Shirui Pan, Yan Zhang, Philip Torr, Zhicheng Dou, Ji-Rong Wen, Xuanjing Huang, Yu-Gang Jiang, and Shuicheng Yan. Memory in the age of ai agents, 2026.

[Li and Chu, 2025] Qi Li and Xiaowen Chu. AdaEdit: Advancing continuous knowledge editing for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4127–4149, Vienna, Austria, July 2025. Association for Computational Linguistics.

[Li *et al.*, 2025] Zherui Li, Houcheng Jiang, Hao Chen, Baolong Bi, Zhenhong Zhou, Fei Sun, Junfeng Fang, and Xiang Wang. Reinforced lifelong editing for language models, 2025.

[Liu *et al.*, 2025] Xiyu Liu, Zhengxiao Liu, Naibin Gu, Zheng Lin, Wanli Ma, Ji Xiang, and Weiping Wang. Relation also knows: Rethinking the recall and editing of factual associations in auto-regressive transformer language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24659–24667, Apr. 2025.

[Luo *et al.*, 2025] Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, Yiqiao Jin, Fan Zhang, Xian Wu, Hanqing Zhao, Dacheng Tao, Philip S. Yu, and Ming Zhang. Large language model agent: A survey on methodology, applications and challenges, 2025.

[Meng *et al.*, 2022a] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35, 2022.

[Meng *et al.*, 2022b] Kevin Meng, Arnab Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *ArXiv*, abs/2210.07229, 2022.

[Minaee *et al.*, 2025] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2025.

[Mitchell *et al.*, 2022a] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale, 2022.

[Mitchell *et al.*, 2022b] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale, 2022.

[Qwen *et al.*, 2025] 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. Qwen2.5 technical report, 2025.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

[Wang *et al.*, 2024a] Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models, 2024.

[Wang *et al.*, 2024b] Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey, 2024.

[Wu *et al.*, 2025] Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. From human memory to ai memory: A survey on memory mechanisms in the era of llms, 2025.

[Zhang *et al.*, 2024a] Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models, 2024.

[Zhang *et al.*, 2024b] Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model based agents, 2024.

[Zhang *et al.*, 2025] Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. Uncovering overfitting in large language model editing, 2025.