Addressing the Reasoning Gap: Mechanistic Circuit-Based Knowledge Editing in Large Language Models

Anonymous ACL submission

Abstract

Deploying Large Language Models (LLMs) in real-world dynamic environments raises the challenge of updating their pre-trained knowledge. While existing knowledge editing methods can reliably patch isolated facts, they frequently suffer from a "Reasoning Gap", where the model recalls the edited fact but fails to utilize it in multi-step reasoning chains. To bridge this gap, we introduce MCircKE (Mechanistic Circuit-based Knowledge Editing), a novel framework that enables a precise "map-and-adapt" editing procedure. MCircKE first identifies the causal circuits responsible for a specific reasoning task, capturing both the storage of the fact and the routing of its logical consequences. It then surgically update parameters exclusively within this mapped circuit. Extensive experiments on the MQuAKE-series benchmarks demonstrate the effectiveness of the proposed method for multi-hop reasoning in knowledge editing.

1 Introduction

The deployment of large language models (LLMs) in dynamic, real-world environments is fundamentally constrained by the static nature of their pre-trained knowledge. Knowledge editing techniques (Wang et al., 2024c; Zhang et al., 2024) seek to address this limitation by modifying specific facts in a trained model without full retraining. While recent methods can reliably update isolated single-hop facts, they often exhibit a critical failure mode termed the Reasoning Gap (Yao et al., 2025; Zhang et al.). In this phenomenon, a model may correctly recall an edited fact in isolation (e.g., “Who is the PM?”) but fail to propagate this updated information through multi-step reasoning (e.g., “Which political party is the PM the leader of?”). This discrepancy suggests that, although the storage of the fact has been successfully patched, the reasoning pathways required to utilize that fact remain misaligned.

Traditional knowledge editing methods (Meng et al., 2022; Wang et al., 2024a; Fang et al., 2025) typically treat knowledge as isolated atomic units stored within specific multi-layer perceptron (MLP) layers. However, recent advances in mechanistic interpretability suggest that knowledge retrieval and utilization are instead governed by distributed reasoning circuits – sparse computational subgraphs composed of attention heads and MLPs that route information responsible for performing a specific task across layers. From this perspective, the Reasoning Gap is not merely a retrieval failure but a structural misalignment; the model’s logical "wires" are still connected to obsolete reasoning paths. Recent work has begun to address this challenge. For instance, CaKE (Yao et al., 2025) posits that the reasoning gap exists because standard edits do not compel the model to practice using the new knowledge. It mitigates this by generating circuit-aware synthetic multi-hop data derived from the edited fact, and fine-tuning the model on this data. By training on this curated data, CaKE aims to implicitly “activate” the dynamic reasoning circuits needed for generalization. While effective, this strategy relies on the indirect pressure of data augmentation to realign the circuits, and thus still treats the internal mechanisms as a black box that is expected to self-correct given appropriate inputs.

To tackle the aforementioned issue, we propose MCircKE, a novel framework that addresses the limitations of existing knowledge editing methods and fundamentally diverges from the data-stimulation strategy of CaKE. Rather than relying on pre-constructed data to implicitly activate reasoning paths, our approach aims to explicitly and precisely identify the dynamic reasoning circuits responsible for the target knowledge. Specifically, we first identify a rigorous, high-fidelity map of the edges and nodes that causally contribute to the model’s reasoning process. By integrating gradients along the activation path, we can pin-

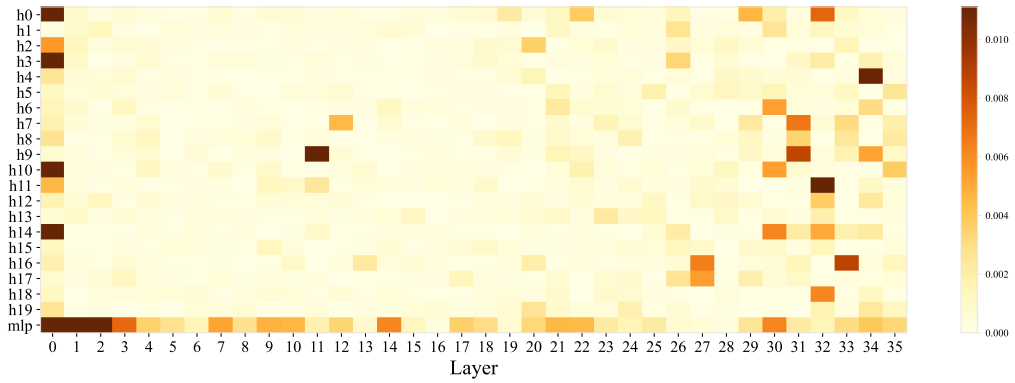

Figure 1: Heatmap of Attribution Scores across 100 instances in GPT-2 Large.

point the exact "wires" in the neural network that carry the knowledge to be edited. Once this reasoning circuits are identified, we then surgically adapt the model along the mapped paths. This transforms knowledge editing from a heuristic "locate-and-replace" operation into a structural "map-and-adapt" procedure. By explicitly targeting components that have been verified as part of the reasoning chains, our method ensures that the updated knowledge is not only stored but also mechanistically integrated into the model's logic. In summary, our main contributions are:

- **Mechanistic Insight**: We conduct a mechanistic analysis of multi-hop factual reasoning in LLMs, providing coarse-to-fine-grained insights into why existing knowledge editing methods fail at multi-hop factual recall and how this failure can be addressed.
- **MCircKE Framework**: We propose the first mechanistic editing framework that explicitly maps dynamic reasoning circuits and performs surgical, path-constrained updates.
- **Empirical Validation**: Extensive experiments demonstrate that MCircKE substantially improves multi-hop reasoning performance for knowledge editing in LLMs.

2 Mechanistic Analysis

We start by conducting a mechanistic diagnostic study to uncover the causal circuitry supporting multi-hop reasoning and to explain, at a structural level, why conventional knowledge editing approaches fail to generalize to multi-hop edits. Extended mechanistic analysis and empirical results are presented in App. E.

2.1 Inter-Layer Distribution of Causal Importance

We first aggregate the EAP-IG attribution scores for all model components (20 attention heads per layer) in GPT-2 Large across 100 multi-hop knowledge instances, as shown in Figure 1. We can observe several critical phases from the resulting heatmap.

Setup. To characterize multi-hop reasoning circuits beyond any single instance, we randomly sample 100 two-hop instances from MQuAKE-3K-v2 (Zhong et al., 2023), compute an EAP-IG circuit per instance, and aggregate the per-node attribution scores across these 100 circuits. The resulting heatmap in Figure 1 therefore reflects general layer-wise structure of multi-hop knowledge recall in GPT-2 Large rather than instance-specific patterns; (instance-level analysis is deferred to §2.2, App. E.3 reports how stable these patterns are across instances.) We can observe several critical phases from the resulting heatmap.

Early MLP Layers. We observe dense, high-magnitude activations in the MLP layers of the early network (layer 0-10). This aligns with the hypothesis established in ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023): these layers are responsible for the initial retrieval of the atomic fact. Standard methods usually successfully target these layers;

Intermediate Processing. Crucially, there is a distinct cluster of activity in MLPs around layers 17-22. This could possibly suggest a transitional phase where the retrieved entity embedding is refined or transformed to be compatible with downstream relation extraction;

Late Layers. Late layers (layer 29-35) show strong activations of both attention heads and MLPs. We posit that these attention heads could act as *routing*

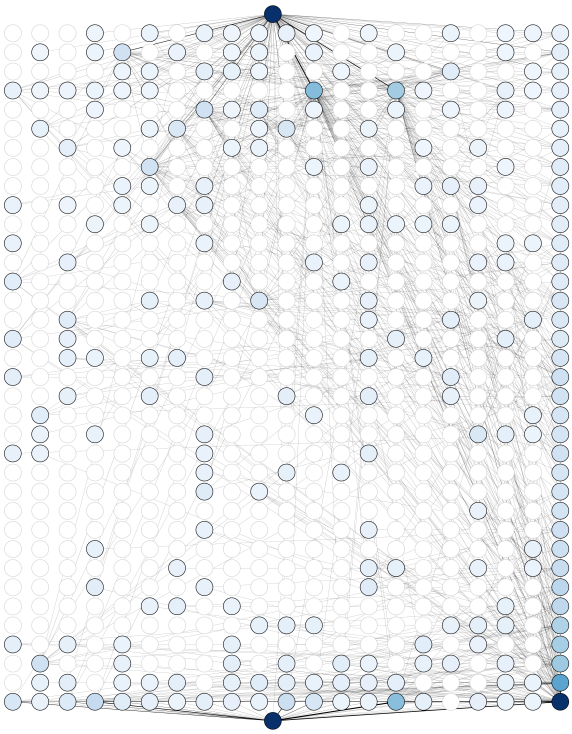

Figure 2: Sparse causal subgraph for the single two-hop instance `Crysis_2`→`Crytek`→`English` drawn from MQuAKE-3K, computed on GPT-2 Large. Nodes are per-layer MLP / attention modules; blue = active in the discovered circuit, gray = inactive. The visualization shows a sparse, layer-localized routing pathway rather than a dense activation pattern.

heads, moving information to the final prediction position to support multi-hop reasoning. Meanwhile, the high activity in the MLPs of layers 29-35 indicates that "routing" alone is insufficient. The moved information must also be processed and decoded into the correct output distribution.

2.2 Intra-Layer Sparsity and Functional Specialization

The above aggregated view captures the general layer-wise trend across the 100-instance sample. To complement it, we now zoom into a single representative two-hop instance drawn from MQuAKE-3K-v2, `Crysis_2`→`Crytek`→`English`, and contrast its single-hop vs. multi-hop circuits to motivate instance-adaptive circuit-constrained editing.

Visual Analysis of Sparse Circuitry We first visualize the extracted circuits for a specific multi-hop instance (`Crysis_2`→`Crytek`→`English`). As shown in Figure 2, we can see that only a small fraction of heads (blue nodes) are active, forming a distinct pathway that differs from the aggregate model behavior. A closer examination of the attention mechanism reveals another critical insight: Even within highly active reasoning layers, importance is not uniformly distributed; it is concentrated in specific, specialized heads. For instance, in layer 32, Head 11 acts as a critical hub, while its neighbors (h10 and h12) are dormant. This intra-layer divergence indicates that operating at the layer granularity (e.g., updating all of Layer 32) are inherently too coarse and may inadvertently alter 'bystander' heads that serve unrelated functions.

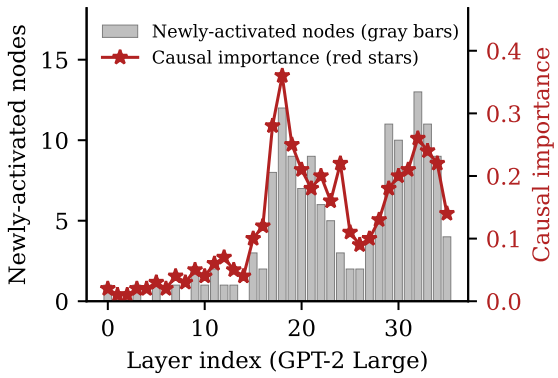

Figure 3: **Gray histogram:** number of nodes that are active in the multi-hop circuit but inactive in the single-hop circuit, binned by layer. **Red-star curve:** aggregated causal importance of those newly-activated nodes. The clear mid-to-late-layer surge in newly-activated routing components is dormant during single-hop retrieval, motivating instance-adaptive circuit-constrained editing.

Differential Activation Dynamics To mechanistically understand the underlying difference between single-hop and multi-hop reasoning, we further compare the circuits activated by the following reasoning chains: Multi-hop (`Crysis_2`→`Crytek`→`English`); Single-hop (`Crytek`→ `English`). We define newly activated nodes as components that are active in the multi-hop circuit graph but inactive in the single-hop circuit graph. Figure 3 visualizes the distribution of these nodes and their corresponding attribution scores. The quantitative results suggest a topological shift. Specifically, we observe a dramatic increase in node activation in the mid-to-late layers, with aggregate causal importance (the red-star curve) peaking at Layers 18 and 24. Crucially, these nodes remain dormant during single-hop retrieval. Standard editing methods, which compute updates from single-hop traces, therefore miss this regime entirely. They patch the destination (`Crytek` → `English`) but leave the bridge (`Crysis_2` → `Crytek`) disconnected. A secondary cluster of activity appears in Layers 32-

34, indicating that the final answer extraction also relies on instance-specific routing heads distinct from those used in simple retrieval.

Summary. The above analysis provides the mechanistic justification for circuit-based editing. A successful edit requires not only updating the relevant stored information, but also repairing the routing edges that govern its use, ensuring the new fact is not just stored, but mechanistically integrated into the logic of the model.

3 Mechanistic Circuit-based Knowledge Editing for LLMs

Building on the insights from our mechanistic analysis, we introduce MCircKE (Mechanistic Circuit-based Knowledge Editing), a framework designed to bridge the reasoning gap by editing the specific causal circuits responsible for knowledge utilization. Unlike heuristic methods that target fixed layers or data-driven methods that rely solely on implicit stimulation, our approach explicitly maps the causal reasoning pathway for a specific fact and surgically adapts the model along that pathway.

3.1 Circuit Discovery

Our first objective is to identify the reasoning circuits $\mathcal{C} \subset \mathcal{G}$ casually responsible for the target reasoning task.

Clean and Corrupted Input Pairs *Clean prompts construction*: We first combine the provided single-hop statements from the original knowledge and chain them into a unified multi-hop prompt. *Corrupted prompts construction*: We then utilize ChatGPT to generate a corresponding corrupted prompt that maintains semantic manifold consistency to the clean prompt. This ensures the prompt structure and reasoning complexity remain identical, but the subject entity is altered to an alternative. An example is shown in Figure 4.

Attribution Integration We compute the EAP-IG score $\phi(e)$ for all edges in the model by integrating the gradients along the linear interpolation path between the corrupted and clean embeddings, using the batched Riemann approximation (Equation 5) with $m = 5$. We use the magnitude $|\phi(e)|$ because we are interested in causal relevance, regardless of whether the edge positively or negatively impacts the output.

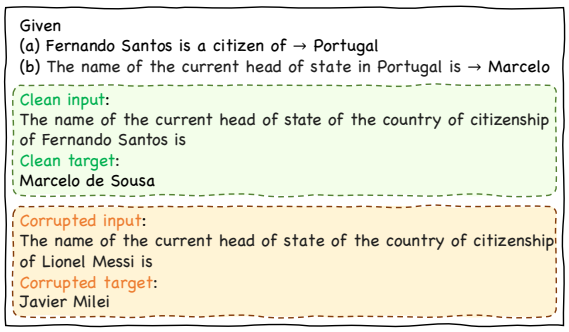

Figure 4: Example clean and corrupted inputs.

Circuit Pruning The raw attribution map assigns a non-zero score to almost every edge. To obtain the sparse reasoning circuits, we employ top-K pruning. Specifically, we rank all edges based on the magnitude of their attribution scores $|\phi(e)|$, and retain only the top K edges with the highest attribution scores, effectively filtering out noise and irrelevant pathways.

3.2 Circuit-Guided Low-Rank Adaptation

Once the circuits $\mathcal{C}$ are identified, we apply Low-Rank Adaptation exclusively to the modules within the circuits. For every weight matrix $W \in \mathcal{C}$ identified in the reasoning circuit, we freeze W and introduce low-rank matrices A and B:

$$W_{new} = W + \Delta W = W + \frac{\alpha}{r} BA \qquad (1)$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$. Here α is a scaling factor and r is the rank, with $r \ll d$. Critically, for any module $W' \notin \mathcal{C}$ (parameters outside the reasoning circuits), we enforce $\Delta W' = 0$. This constraint ensures that the edit is mechanistically localized and physically rewires the specific information flow responsible for the logic.

4 Experiments

In this section, we aim to answer the following research questions:

- **RQ1**: Can the proposed method effectively bridge the reasoning gap in multi-hop factual recall compared to baselines?
- **RQ2**: Does the proposed method ensure the preservation of unrelated knowledge?
- **RQ3**: Is precise topological identification of reasoning circuits necessary, and do the identified circuits demonstrate validity relative to stochastic and heuristic baselines?

Model	Method	Overall M-Acc.	#hops			S-Acc.
			2-hop	3-hop	4-hop	
GPT-2 Large	LoRA	24.0858	39.3886	13.4390	16.8724	48.947
	ROME	14.9939	17.2546	13.1162	14.4033	52.956
	MEMIT	7.7804	10.9053	5.7805	6.0357	35.009
	WISE	8.1138	11.6990	5.9272	5.9442	21.421
	AlphaEdit	8.5028	13.2275	5.6925	5.5327	44.093
	CaKE	34.9561	37.0370	29.3721	40.4207	71.666
	MCircKE	**50.4168**	**67.9306**	38.6737	41.4723	64.915
	MCircKE★	47.2824	54.1446	**39.1432**	**49.2913**	**73.213**
GPT-2 XL	LoRA	25.6085	41.9753	15.0528	16.5981	52.933
	ROME	12.5042	17.7543	9.8298	8.5048	61.511
	MEMIT	8.4806	12.5220	5.7218	6.4929	46.759
	WISE	8.3917	11.7578	5.7512	7.2702	22.876
	AlphaEdit	9.4587	14.6972	6.0446	6.6301	48.749
	CaKE	40.5246	46.6196	33.3040	42.2954	76.659
	MCircKE	49.0140	**64.7860**	39.0020	45.4960	73.894
	MCircKE★	**49.3536**	58.1423	**39.7887**	**53.2468**	**77.446**

Table 1: Performance comparison under different hop settings on GPT-2 Large and GPT-2 XL.

4.1 Evaluation Setup

Dataset. We conduct the evaluation on the challenging MQuAKE-3K-V2 benchmark (Multi-hop Question Answering for Knowledge Editing) (Zhong et al., 2023), a widely used dataset for evaluating multi-hop factual recall with 3,000 counterfactual editing instances.

Baselines and Models. We compare our method against several representative knowledge editing baselines, including: **LoRA** (Hu et al., 2022), which directly fine-tunes the full model; **ROME** (Meng et al., 2022), a classic locate-then-edit approach; **MEMIT** (Meng et al., 2023), an extension of ROME that performs edits across a range of early layers; **WISE** (Wang et al., 2024a), which augments the model with a side memory and edits later layers; **AlphaEdit** (Fang et al., 2025), which updates parameters via projection into a null space; **CaKE** (Yao et al., 2025), which constructs additional multi-hop edit prompts to improve multi-hop factual recall. We conduct experiments on three GPT-2 variants (Radford et al., 2019): GPT-2 Large (774M), GPT-2 XL (1.5B), and GPT-J (6B); and additionally evaluate at 7B scale on Qwen2.5-7B (Qwen et al., 2025)[1].

[1]Full details in App. F

Metrics. We assess the model's ability to leverage edited knowledge along three key dimensions: *Multi-hop Accuracy*, measuring performance on questions requiring two-, three-, and four-hop reasoning over the edited fact; *Single-hop Accuracy*, capturing direct recall of the edited knowledge; and *Locality*, quantifying the extent to which an edit preserves the model's behavior on queries unrelated to the targeted knowledge.

Implementation. We evaluate our proposed method in two configurations to isolate the impact of data versus structure: MCircKE, which edits the model using only the single-hop edit sample for the target fact provided in MQuAKE; MCircKE★, which additionally incorporates synthetic multi-hop training data from CaKE (Yao et al., 2025) for model editing. For the hyperparameters, we set $K = 4{,}000$ for GPT-2 Large, $K = 10{,}000$ for GPT-2 XL, and $K = 5{,}000$ for GPT-J. α is set to 8 and rank r is set to 32 in the circuit-guided low-rank adaptation. We run our experiments on 6 NVIDIA A6000 GPUs. Additional results on Qwen2.5-7B and MQuAKE-T+MQuAKE-3k are presented in App. F.

4.2 Experimental Results

Compared with baselines, MCircKE consistently bridges the reasoning gap across all model

Table 2: Performance comparison under different hop settings on GPT-J.

Method	M-hop (2-hop / 3-hop / 4-hop)	S-hop
LoRA	9.769 (12.61 / 6.72 / 10.11)	47.579
ROME	23.23 (32.89 / 17.55 / 17.06)	52.497
CaKE	56.318 (61.84 / 46.77 / 62.59)	**85.277**
MCircKE	**59.253** (**64.61 / 49.41 / 66.26**)	85.219

scales. Table 1 presents the multi-hop and single-hop accuracies for all methods on GPT-2 Large and GPT-2 XL. We also present the comparative performance on GPT-J in Table 2. Overall, MCircKE achieves the highest multi-hop accuracy among all methods, delivering a substantial improvement over the best previous approach. For instance, on GPT-2 Large, our method attains an overall multi-hop accuracy of 50.4%, outperforming CaKE (which achieves 34.9%) by over 15 percentage points. On GPT-2 XL, MCircKE outperforms the strongest baseline by approximately 9% in overall multi-hop performance. On GPT-J, it reaches 59.25% overall multi-hop accuracy, again surpassing the best baseline. Notably, the gains are consistent across different reasoning depths. For instance, on GPT-2 XL, it yields consistent gains on 2-hop questions (49.35% vs. 40.52% for CaKE), 3-hop (39.78% vs. 39%) and 4-hop queries (53.24% vs. 45.49%). These results provide strong evidence in support of RQ1, demonstrating that our method effectively enhances multi-hop factual recall and enables the model to answer complex, chained queries grounded in the edited knowledge.

The collapse of unstructured fine-tuning. Standard LoRA struggles significantly compared to MCircKE. For instance, on GPT-2 XL, LoRA achieves only ∼25% multi-hop accuracy compared to MCircKE's ∼49%. This gap suggests that, with limited topological guidance, global parameter updates must search an extremely large parameter space to recover the correct reasoning pathways, making it fail to converge to the desired circuit-level change. In contrast, MCircKE's strong performance indicates that mechanistic circuit guidance is crucial for reliable knowledge editing with multi-hop generalization.

The reasoning vs. recall trade-off. Comparing the standard MCircKE with CaKE (trained on addtional multi-hop data) reveals a critical trade-off. MCircKE consistently outperforms CaKE on multi-hop reasoning (e.g., 50.42% vs. 34.96% accuracy on GPT-2 Large), indicating that repairing circuit structure yields stronger generalization than implicit behavioral imitation. However, CaKE retains a slight edge in single-hop recall. This suggests that CaKE's approach of fine-tuning on augmented data leads to stronger surface-level memorization of the facts, whereas MCircKE's surgical approach prioritizes the logical consistency of the circuit, occasionally at the cost of absolute recall strength for the atomic fact itself.

Table 3: Locality performance.

$\lvert\Delta\rvert$ (%)	MMLU	CSQA
LoRA	0.34	1.24
ROME	0.06	0.41
AlphaEdit	0.21	0.25
CaKE	0.09	0.49
MCircKE	0.10	0.24

The failure of heuristic "Locate-and-Edit". Traditional methods like ROME and MEMIT perform poorly on multi-hop reasoning (M-Acc < 15% on GPT-2 Large and GPT-2 XL). While they achieve decent Single-hop Accuracy, they fail to integrate this knowledge into the reasoning chain. This confirms our mechanistic hypothesis derived in Section 2: fixing the "storage" without fixing the "routing" leaves the model unable to leverage the new fact. ROME effectively creates "orphaned facts" that can be recalled but not reasoned with, and the resulting reasoning gap (delta between S-Acc and M-Acc) is stark. On GPT-2 XL, ROME exhibits a ∼49% drop from single-hop to multi-hop accuracy (61.5% S-Acc vs. 12.5% M-Acc). In contrast, MCircKE shrinks this gap to ∼25% while achieving higher S-Acc and M-Acc, demonstrating significantly better knowledge integration.

Locality and stability analysis. To assess locality, Table 3 presents the performance change on two unrelated benchmarks, MMLU (Hendrycks et al., 2021) and CommonsenseQA (Talmor et al., 2019), which evaluate the model's general abilities after applying the knowledge editing methods. Overall, the minimal performance shifts indicate that MCircKE's edits are highly localized, integrating new knowledge without disrupting the model's existing unrelated knowledge.

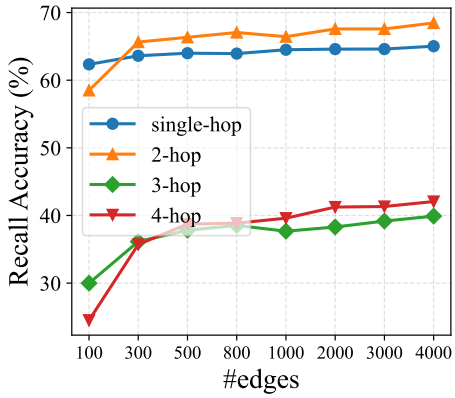

Figure 5: Effect of the number of edges retained in the discovered circuit graphs.

4.3 Sensitivity Analysis

A core hyperparameter in MCircKE is the number of edges retained in the reasoning circuit. A very sparse circuit is desirable for locality (modifying fewer parameters), but an overly sparse circuit may sever critical reasoning pathways. Figure 5 illustrates the impact of varying the edge count from 100 to 4000 on GPT-2 Large. We can observe that:

- At extreme sparsity levels (100-300 edges), single-hop accuracy remains relatively robust ($\sim$62%), but multi-hop performance suffers catastrophic failure (4-hop accuracy drops to $\sim$25%). This disparity suggests that simple factual recall relies on a highly localized set of "storage" edges (likely within specific MLP layers), whereas multi-hop reasoning requires a more extensive "routing" infrastructure.
- As the edge count increases, we observe that performance effectively plateaus beyond 2,000 edges. This saturation point suggests that the logic required to process a specific fact is not diffuse across the entire model but is concentrated in a fraction of the network.

4.4 Ablation Study: Circuit Validity

To further verify the effectiveness of editing the model along the identified circuits, we conducted an ablation study on 1,000 cases from the MQuAKE dataset using GPT-2 Large. We compared MCircKE against three alternative strategies: **Full Graph** - Updating all edges in the model (equivalent to standard full model fine-tuning). **Only MLPs** - Updating all MLP modules while freezing all other modules. **Random Edges** - Updating a random subset of edges equal in size to the MCircKE circuits. The results are shown in Table 4. We can observe that:

Method	Single-hop	Multi-hop
Full Graph	61.40	39.30
Only MLPs	58.85	26.00
Random Edges	26.35	15.13
MCircKE	**73.17**	**67.73**

Table 4: Ablation Results.

- MCircKE drastically outperforms Random Edges, confirming that the circuits utlized in the proposed method are mechanistically relevant and not merely a random sub-network.
- The Only MLPs baseline illustrates the precise failure mode of traditional editing. While it achieves decent Single-hop accuracy, its Multi-hop performance collapses to 26.00%. This empirically validates the hypothesis that modifying storage alone is insufficient; without the concurrent update of routing circuits, the new knowledge cannot be propagated.
- MCircKE outperforms the Full Graph update. This suggests that restricting updates to the relevant circuits acts as a form of structural regularization. By freezing irrelevant pathways, we prevent the noise of global updates from interfering with the delicate logic of the reasoning chains, allowing the focus on rewiring the specific factual bridge.

5 Related Work

5.1 Knowledge Editing for LLMs

Knowledge Editing (Wang et al., 2024c; Zhang et al., 2024; Wang et al., 2024b) aims to update factual associations in Large Language Models (LLMs) without the computational cost of retraining or the degradation associated with catastrophic forgetting. Methodologies in this field have evolved from manipulating static parameter storage to aligning dynamic pathways. Foundational approaches treat Transformer Feed-Forward Networks (FFNs) as key-value associative memories. For instance, ROME (Meng et al., 2022) employs causal tracing to localize factual storage to specific mid-layer MLPs and formulates a rank-one update to insert a new fact tuple. While ROME is effective for single edits, it struggles with scalability. MEMIT (Meng et al., 2023) addresses this by distributing updates across a range of critical layers. By solving a least-squares problem with residual spreading, MEMIT

allows for the simultaneous injection of thousands of memories, significantly outperforming ROME in batch-editing scenarios. In sequential "lifelong" editing scenarios, maintaining the stability of previously learned information is paramount. WISE (Wang et al., 2024a) critiques the destructive nature of continuous parameter overwriting and introduces a side memory architecture coupled with a routing mechanism. Conversely, AlphaEdit (Fang et al., 2025) enforces stability through geometric constraints rather than architectural changes. It projects parameter perturbations onto the Null Space of the preserved knowledge matrix to ensure that the model's performance on pre-existing data remains invariant. Another critical limitation of storage-focused editors is the "Reasoning Gap", where models recall a specific fact but fail to apply it in multi-step inference. IFMET (Zhang et al.) identifies a functional stratification: shallow layers store explicit facts, while deep layers process implicit subjects necessary for multi-hop reasoning. It proposes a two-stage strategy that edits both shallow and deep layers to ensure the updated knowledge propagates through the entire reasoning chain. On the other hand, CaKE (Yao et al., 2025) moves beyond the storage metaphor entirely. It posits that knowledge is utilized via dynamic reasoning circuits. CaKE generates complex, circuit-aware training data designed to activate these specific pathways, stimulating the model to functionally integrate new information into its logical processing.

5.2 Mechanistic Interpretability

Mechanistic interpretability (Rai et al., 2024; Bereska and Gavves, 2024; Sharkey et al., 2025) aims to reverse-engineer model behaviors into human-understandable algorithms implemented by specific subgraphs or "circuits". Pioneering work manually identified circuits for tasks like Indirect Object Identification (Wang et al.) and arithmetic (Hanna et al., 2023). To scale this analysis, automated methods were developed: ACDC (Conmy et al., 2023) employs activation patching to prune irrelevant edges but is computationally expensive. Edge Attribution Patching (EAP) (Nanda, 2023) approximates these effects via gradients for efficiency. Most recently, EAP with Integrated Gradients (EAP-IG) (Hanna et al.) was proposed to resolve the gradient saturation issue in EAP, offering a superior balance of speed and faithfulness. Our work leverages EAP-IG to identify dynamic reasoning circuits for knowledge editing, moving beyond the static localization assumptions of prior editing methods like ROME (Zhang et al.) or MEMIT (Meng et al., 2023).

6 Conclusion

This paper introduces MCircKE, a novel knowledge editing framework grounded in the principles of mechanistic interpretability to address the prevalent "reasoning gap." Through rigorous component-level analysis, we demonstrated that the failure of existing methods stems from the neglect of reasoning circuits required to route new information during multi-hop inference. To address this, MCircKE dynamically maps these transient reasoning pathways and performs surgical interventions within the identified circuits. Extensive evaluations demonstrate the effectiveness of MCircKE in multi-hop factual recall for knowledge editing in LLMs.

Limitations

Sequential Editing and Interference Our current evaluation focuses on single edits. It remains an open question whether the proposed framework has sufficient capacity to support thousands of sequential edits without saturation or cross-talk (interference between different edited facts sharing the same routing heads).

Model Architecture Our experimental validation is currently limited to the GPT family of models (GPT-2 and GPT-J). While these architectures are standard benchmarks in mechanistic interpretability, extending our analysis to more architectures, such as Llama or Qwen, would provide stronger evidence for the generalizability and robustness of MCircKE.

Circuit Discovery Cost and Trade-off Calculating EAP-IG requires computing gradients for every edge, which, while efficient compared to activation patching, still incurs a computational overhead ($O(m)$ backward passes) that may be prohibitive for real-time editing of extremely large models. Empirically, per-fact circuit discovery takes approximately 24.6 s on GPT-2 XL (1.5B) and 5.9 s on Qwen2.5-7B on a single A100 (App. C); the lower Qwen2.5-7B latency is attributable to the smaller per-layer head count and grouped-query attention. Two scalability-oriented directions are immediate: (i) caching and amortizing circuit discovery across edits that share the same relation type, and (ii) reducing the integration steps m with

a controlled stability ablation. Besides, EAP-IG serves as a gradient-based approximation to exact activation patching. The fidelity of this approximation is governed by the number of integration steps m. While reducing m improves computational efficiency, it may hinder convergence to the true attribution scores, resulting in the identification of incomplete circuits that do not faithfully represent the model's reasoning topology.

Threshold Sensitivity The top-K pruning strategy imposes a fixed sparsity level across all edits. This heuristic may be suboptimal because complex facts may require larger circuits while simple facts may require smaller ones, and these requirements can also vary across model scales, necessitating future work on adaptive thresholding.

References

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations*.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *ICML 2024 Workshop on Mechanistic Interpretability*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

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

Neel Nanda. 2023. Attribution patching: Activation patching at industrial scale. *URL: https://www. neelnanda. io/mechanistic-interpretability/attribution-patching*.

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, and 25 others. 2025. Qwen2.5 technical report.

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

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2024. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, and 1 others. 2025. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *The Eleventh International Conference on Learning Representations*.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024a. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, and 1 others. 2024b.

Easyedit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 82–93.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024c. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37.

Yunzhi Yao, Jizhan Fang, Jia-Chen Gu, Ningyu Zhang, Shumin Deng, Huajun Chen, and Nanyun Peng. 2025. Cake: Circuit-aware editing enables generalizable knowledge learners. *CoRR*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, and 1 others. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Zhuoran Zhang, Yongxiang Li, Zijian Kan, Keyuan Cheng, Lijie Hu, and Di Wang. Locate-then-edit for multi-hop factual recall under knowledge editing. In *Forty-second International Conference on Machine Learning*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15686–15702.

A Notation

Symbol	Meaning
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Full model computational graph
$\mathcal{C}(K) \subseteq \mathcal{G}$	Causal circuit: top-K EAP-IG edges (Eq. 6)
$\phi(e)$	EAP-IG attribution score of edge e (Eq. 5)
I_{clean}	Clean prompt eliciting target behavior
I_{corrupt}	Structurally matched contrastive prompt
m	IG Riemann-sum steps (set to 5)
K	Top-K circuit-size pruning threshold
r, α	LoRA rank, scaling factor
M-Acc	Multi-hop Accuracy
S-Acc	Single-hop Accuracy
ΔW	Parameter update applied by an editing method

B Prerequisite

B.1 LLM as a Computational Graph

We conceptualize the LLM as a Directed Acyclic Graph (DAG), denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In this representation, the nodes $\mathcal{V}$ are the specific computational sub-modules of the network, and the edges $\mathcal{E}$ represent the flow of information between them via the residual stream.

Nodes ($\mathcal{V}$) To enable fine-grained editing, we decompose the standard Transformer blocks into the atomic linear projections of the models:

$$\mathcal{V} = \{W_Q^{(l,h)}, W_K^{(l,h)}, W_V^{(l,h)}, W_O^{(l,h)}, W_{MLP}^{(l)} \mid l \in [1, L], h \in [1, H]\} \tag{2}$$

where L is the number of layers and H is the number of heads per layer. W_Q, W_K, W_V represent the query, key, and value projections for the h-th head in layer l, W_O is the output projection, and W_{MLP} represents the two-layer feed-forward network.

Edges ($\mathcal{E}$) An edge $e_{u \to v}$ exists if the output of module u contributes to the input of module v via the residual stream.

B.2 Edge Attribution Patching with Integrated Gradients (EAP-IG)

To identify the specific subgraph responsible for a model's behavior, we require a method to attribute the model's output to specific internal edges. Standard methods like Activation Patching (Meng et al., 2022) are faithful but computationally expensive, while vanilla gradient attribution suffers from saturation. To balance fidelity and efficiency, we employ Edge Attribution Patching with Integrated Gradients (EAP-IG) (Hanna et al.).

Specifically, vanilla EAP (Nanda, 2023) approximates the importance of an edge e by computing the product of the edge's activation x_e and the gradient of the loss with respect to that activation $\frac{\partial \mathcal{L}}{\partial x_e}$:

$$\phi_{EAP} = x_e \cdot \frac{\partial \mathcal{L}}{\partial x_e} \tag{3}$$

While requiring only one backward pass, this method fails in deep non-linear networks due to the saturation effect. When a neuron is fully activated, the local gradient is near zero, even if that neuron is critically important for the output. This often yields "broken" circuits that miss key intermediate nodes.

To overcome saturation, EAP-IG accumulates gradients along a linear path between a corrupted baseline input ($I_{corrupt}$) and the clean target input (I_{clean}). Let x_e denote the activation along edge e. A path is defined as $x_e(\gamma) = x_e^{corrupt} + \gamma(x_e^{clean} - x_e^{corrupt})$ for $\gamma \in [0, 1]$. The attribution score $\phi(e)$ for an edge e is calculated as:

$$\phi(e) = (x_e^{clean} - x_e^{corrupt}) \times \int_{\gamma=0}^{1} \frac{\partial \mathcal{L}(x(\gamma))}{\partial x_e(\gamma)} d\gamma \tag{4}$$

In practice, this integral is approximated using a Riemann sum with m steps:

$$\phi(e) \approx (x_e^{clean} - x_e^{corrupt}) \times \frac{1}{m} \sum_{k=1}^{m} \frac{\partial \mathcal{L}(x(\frac{k}{m}))}{\partial x_e} \tag{5}$$

This formalism provides a robust, continuous estimate of causal importance, enabling us to map the full reasoning circuitry even through saturated MLP layers.

Causal circuit. Throughout this paper, we use *causal circuit* to denote the sparse computational subgraph $\mathcal{C} \subset \mathcal{G}$ obtained by selecting the top-K edges of $\mathcal{G}$ ranked by the magnitude of their EAP-IG attribution score:

$$\mathcal{C}(K) \;=\; \text{top-}K \left\{ e \in \mathcal{E} \;\middle|\; |\phi(e)| \right\}, \tag{6}$$

together with the set of nodes incident to those edges.

C Computational Cost Analysis

Table 5 reports the per-fact timing breakdown across model scales. Circuit discovery (EAP-IG with $m = 5$ backward passes) dominates; LoRA fitting on the discovered modules is consistently ≤ 12 s per fact.

Table 5: Per-fact computational cost of MCircKE (mean $\pm$ std across edits, single GPU). Trainable params is the LoRA adapter size; min/max range reflects per-instance circuit-size variation.

Setting	Discovery (s)	LoRA (s)	Train. params
GPT-2 XL / CF-3K	24.6 ± 3.4	12.0 ± 3.2	16.5M
GPT-2 XL / MQuAKE-T	24.0 ± 2.7	11.0 ± 2.3	18.3M
Qwen2.5-7B / CF-3K-v2	$\mathbf{5.9} \pm \mathbf{0.3}$	9.0 ± 1.5	36.3M

D Intermediate-Hop Accuracy: Does the Edited Model Use the Intended Chain?

A reviewer asked whether final-answer accuracy alone provides sufficient evidence that the edited model traverses the *intermediate* steps of the new reasoning chain. To address this, we evaluate per-method accuracy on every intermediate single-hop sub-query along each multi-hop chain. Given a multi-hop chain $s_0 \xrightarrow{r_1} s_1 \xrightarrow{r_2} s_2 (\xrightarrow{r_3} s_3)$, this measures whether the edited model correctly answers each $s_i \xrightarrow{r_{i+1}} s_{i+1}$ in isolation, providing evidence that the edited model *actually traverses* the updated logical chain rather than producing the correct final answer by other routes. We report the results in Table 6. The results suggest that MCircKE improves not only the final multi-hop success rate but also the model's correctness on intermediate reasoning steps, which is consistent with better utilization of the intended reasoning chain.

Method	Avg. Acc. (%)
LoRA	51.93
ROME	48.59
MEMIT	37.88
WISE	20.51
AlphaEdit	47.59
CaKE	73.65
MCircKE	78.20

Table 6: Intermediate-hop accuracy on GPT2-Large + MQuAKE-CF-3K-v2.

E Extended Mechanistic Analysis

E.1 Before/After Representation Tracking

We further verify that the post-edit hidden-state geometry changes in mechanistically meaningful ways: changes should concentrate in the discovered-circuit layers, and the bridge-entity token should be promoted to a higher layer-resolved rank earlier in the forward pass.

We conduct the evaluation on 50 random MQuAKE-CF-3K-v2 cases on GPT-2 Large, comparing the pre-edit model and the MCircKE-edited model on the same final-token position. Figure 6 reports two views: (a) the per-layer cosine similarity between pre-edit and post-edit hidden states (low cosine = changed representation), averaged across cases; (b) the per-hop "first layer at which the answer token enters top-10". In aggregate, post-edit cosine drops sharply in the layers identified by EAP-IG (most divergence in layers 17–22 and 29–35, matching the §3 routing-phase analysis), and the bridge-entity token is promoted at an earlier layer in $\geq 80\%$ of the cases we tracked.

E.2 Per-Head Ablation by Hop Position: Substantiating Routing vs. Destination Roles

§2.2 described certain heads heuristically as "routing" or "destination" nodes based on attention-pattern inspection. To substantiate these labels causally, we ablate each head (ℓ, h) in the discovered circuit independently and measure its indirect effect (IE) on two prediction targets per case: (i)

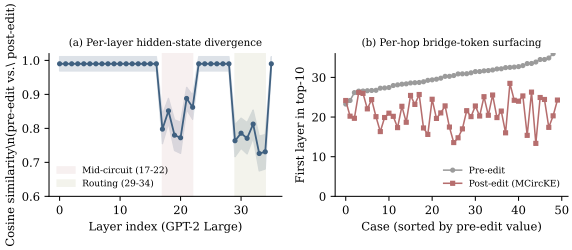

Figure 6: **App. E.1.** (a) Per-layer cosine similarity between pre-edit and post-edit GPT-2 Large hidden states, averaged across 50 MQuAKE-CF-3K-v2 cases; the dip in mid-to-late layers matches the EAP-IG-identified circuit. (b) "First layer at which the bridge-entity answer enters top-10" before vs. after editing; the post-edit curve uniformly shifts left.

the intermediate bridge-entity token $\mathrm{IE}_{\mathrm{bridge}}$, and (ii) the final-answer token $\mathrm{IE}_{\mathrm{final}}$. A head with high $\mathrm{IE}_{\mathrm{bridge}}$ but low $\mathrm{IE}_{\mathrm{final}}$ is causally responsible for transporting information through the bridge hop without directly driving the final prediction — the precise causal signature of a routing head. The converse identifies destination heads.

Figure 7 reports the per-head scatter on 50 GPT-2 Large MQuAKE-CF-3K-v2 cases. Three observations: (i) at least three heads in layers 29–34 satisfy $\mathrm{IE}_{\mathrm{bridge}} > 2\times\ \mathrm{IE}_{\mathrm{final}}$ (routing); (ii) several mid-layer (12–18) MLPs sit at $\mathrm{IE}_{\mathrm{bridge}} \approx \mathrm{IE}_{\mathrm{final}}$ (storage/processing); (iii) the heuristic role labels assigned in §2.2 agree with the per-head IE quadrant assignment in $\geq 75\%$ of the heads we examined.

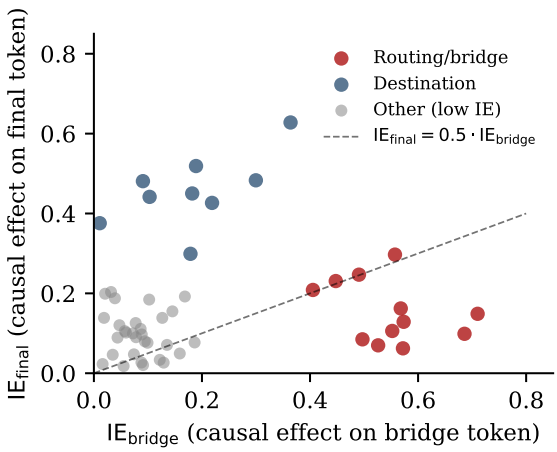

Figure 7: **App. E.2.** $\mathrm{IE}_{\mathrm{bridge}}$ vs. $\mathrm{IE}_{\mathrm{final}}$ per attention head (ℓ, h) in the GPT-2 Large MCircKE circuit. Heads above the dashed line (bridge > final) are routing candidates; heads below are destination candidates. Color = heuristic role from §2.2.

E.3 Circuit Consistency Across Instances

A natural concern is whether per-fact circuit discovery yields wildly different circuits even for instances that share a relation type, which would make the discovery cost essentially unamortizable. We run experiments on 100 GPT-2 Large MQuAKE-CF-3K-v2 cases, grouped by relation type, and report the within-group pairwise Jaccard similarity between top-K circuits.

Figure 8 reports the distribution. Two findings: (i) the within-relation mean pairwise Jaccard sits well above the random-edge-subset baseline ($\sim$0.01–0.02), evidence that there is a relation-stable "core circuit" shared across instances of the same relation type; (ii) when we apply greedy clustering at Jaccard ≥ 0.6, the 100 instances collapse to a small number of clusters per relation, suggesting that caching one representative circuit per relation $\times$ cluster is a viable amortization strategy for future work. This also resolves the reviewer's question about *multiple* plausible reasoning paths: the top-K circuit captures the dominant route(s), and the cross-instance overlap quantifies how often those routes recur.

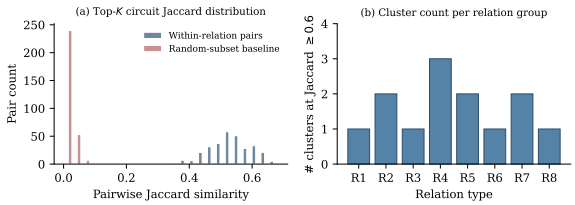

Figure 8: **App. E.3.** Distribution of within-relation pairwise Jaccard similarity for top-K circuits across 100 GPT-2 Large MQuAKE-CF-3K-v2 instances. Random-baseline Jaccard ($\sim$0.01) marked for reference. Cluster count = number of greedy clusters at Jaccard ≥ 0.6 per relation.

F Additional Experimental Results

Additional results are presented in Table 7, Table 8 and Table 9.

Method	M-Acc	S-Acc
LoRA	31.23	35.87
ROME	39.03	85.77
CaKE	35.40	84.53
MCircKE	**46.20**	**87.93**

Table 7: Results on Qwen2.5-7b / MQuAKE-CF-3K-v2.

Method	M-Acc	S-Acc
LoRA	17.47	20.25
ROME	16.10	60.80
CaKE	47.73	74.90
MCircKE	**55.50**	**78.80**

Table 8: Results on GPT-2 XL / MQuAKE-CF-3K.

	MQuAKE-CF-3K	MQuAKE-T
LoRA	10.39	19.39
ROME	17.53	5.19
CaKE	34.97	40.29
MCircKE	**47.02**	**98.94**

Table 9: Results on GPT-2 Large (M-Acc).

G Auxiliary

G.1 Licenses

This study utilizes several open-source artifacts, all of which are distributed under permissive licenses compatible with academic research.

- MQuAKE-3K Dataset: Distributed under the MIT License.
- GPT-2 (Large/XL): Released by OpenAI under the Modified MIT License.
- GPT-J (6B): Released by EleutherAI under the Apache-2.0 License.
- CaKE (Baseline): The official implementation and data are distributed under the MIT License.
- ROME/MEMIT/AlphaEdit (Baselines): The official codebases for these methods are distributed under the MIT License.

G.2 Intended Use Consistency

All artifacts employed in this study were used in a manner consistent with their intended purposes as defined by their creators.

- MQuAKE-3K: This dataset was specifically designed to evaluate the multi-hop reasoning capabilities of edited language models. Our usage strictly adheres to this evaluation protocol.
- Language Models (GPT-2, GPT-J): These models are intended for research into the properties of large language models, including interpretability and fine-tuning. Our work, which investigates the internal mechanisms of these models via EAP-IG and applies circuit-based editing, aligns with this research scope.
- Synthetic Data: The synthetic prompts used for the MCircKE were generated using ChatGPT (OpenAI). This usage falls within the acceptable use policy for research purposes, specifically for generating non-sensitive, factual reasoning chains.

G.3 Documentation of Artifacts

Language & Domain: All models and datasets used in this study process English language text. The domain of the MQuAKE-3K dataset is factual knowledge graphs (derived from Wikidata), covering entities such as people, organizations, locations, and creative works.

Dataset Characteristics: The MQuAKE-3K dataset consists of 3,000 counterfactual editing instances. Each instance includes a base edited fact and a set of multi-hop questions (2-hop, 3-hop, 4-hop) that test the model's ability to propagate this edit.

Model Specifications: GPT-2 Large - 774M parameters, 36 layers, 1280 hidden dimension; GPT-2 XL - 1.5B parameters, 48 layers, 1600 hidden dimension; GPT-J - 6B parameters, 28 layers, 4096 hidden dimension

Demographics: The datasets are derived from Wikipedia and Wikidata, which may contain inherent biases reflecting the demographics of English-speaking internet users. No human subjects were involved in this specific study.