MULTI-HOP KNOWLEDGE EDITING VIA INTERMEDIATE-ENTITY REPRESENTATION ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Even when a single-hop edit succeeds, related multi-hop predictions may remain unchanged. We investigate whether access to the updated intermediate-entity representation can help the edited knowledge participate in downstream computation. Through layer-wise hidden-state interventions, we find that introducing the desired-object representation at the subject position can increase the target probability and redirect the final predictive state toward direct next-hop recall. This intervention identifies intermediate-entity reuse as an intervention-sensitive representational factor associated with multi-hop propagation. Based on this observation, we propose Intermediate-Entity Representation Editing (IERE), which converts the temporary intervention into persistent parameter updates. IERE schedules interacting edits from downstream to upstream, recursively re-extracts the latest desired-object representations, aligns subject representations accordingly, and finally applies batched factual editing. It uses only single-hop edit requests and requires no edit-specific multi-hop questions, next-hop relations, reasoning paths, or target answers. Experiments on Llama-3-8B-Instruct and GPT-J-6B show that IERE preserves strong direct editing, achieves the highest forward multi-hop accuracy among single-hop-only methods in all six settings, and remains competitive with methods that use auxiliary multi-hop information. Relation-specific evaluation further shows subject-level locality comparable to MEMIT.

1 INTRODUCTION

Knowledge editing aims to update specific factual associations in large language models (LLMs) without retraining the entire model. Methods such as ROME and MEMIT can reliably rewrite direct facts through localized parameter updates (Meng et al., 2022; 2023). As illustrated in Figure 1(A)–(B), an edit may change the answer to "the president of the United States is" from "Donald Trump" to "Joe Biden." However, successful direct editing does not guarantee that the updated fact will be reused in related multi-hop questions (Zhong et al., 2023; Cohen et al., 2024). The model may still answer "Melania Trump" rather than "Jill Biden" when asked for the wife of the president of the United States. This multi-hop question composes two facts: the president of the United States is Donald Trump, and his wife is Melania Trump. Although the surface input includes only "the United States," "president of," and "wife of," the model may rely on these components as a shortcut rather than explicitly recovering Donald Trump as the intermediate entity (Ju et al., 2024). After the first fact is edited to Joe Biden, the multi-hop input remains unchanged. If the original shortcut does not incorporate the updated intermediate-entity information, the model may continue producing the old answer, as illustrated in Figure 1(B).

Existing multi-hop editing methods include parameter-preserving and parameter-modifying approaches. Parameter-preserving methods use prompting, retrieval, or stepwise decomposition at inference time, introducing additional task-specific reasoning procedures (Lu et al., 2025; Lee et al., 2025; Shi et al., 2026; Guo et al., 2026). Parameter-modifying methods directly update model weights, but often rely on auxiliary multi-hop information, such as complete questions, next-hop relations, reasoning paths, or target answers, and rarely consider dependencies among interacting edits in a batch (Zhang et al., 2024b; Yang et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

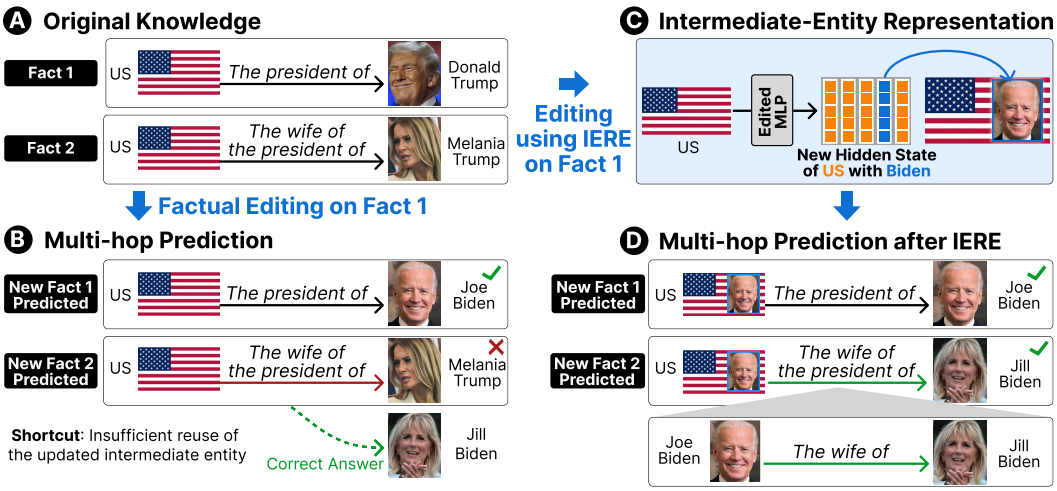

Figure 1: Overview of multi-hop propagation failure and IERE. **(A)** Original factual associations. **(B)** Factual editing updates direct recall but fails to propagate to the multi-hop prediction. **(C)** IERE incorporates edited-object information into the subject representation. **(D)** The aligned representation supports the corrected multi-hop prediction.

This raises the central question of this work: can edited knowledge be made reusable in multi-hop prediction using only the original single-hop edit requests, without providing edit-specific multi-hop questions, next-hop relations, reasoning paths, or target answers?

In a multi-hop question, the edited object can serve as an intermediate entity connecting the edited fact to an existing downstream association. We therefore test whether making its representation available at the subject position can steer the multi-hop computation toward direct next-hop recall. Specifically, we patch hidden-state information from "Joe Biden" into the subject position of "the United States," without modifying model parameters or the textual input. The intervention increases the probability of "Jill Biden" and moves the final predictive state toward that induced by direct next-hop recall.

These observations show that multi-hop prediction is sensitive to desired-object-aligned information at the subject position, motivating intermediate-entity representation as a target for persistent editing. Importantly, this intervention does not assume that the model explicitly recovers the intermediate entity as a symbolic reasoning step; alternative shortcut computations may coexist.

Based on this observation, we propose **Intermediate-Entity Representation Editing (IERE)**, converting the temporary intervention into a persistent parameter update. Figure 1(C) shows IERE edits MLP parameters so the subject-position hidden state incorporates edited-object information. This representation is designed to facilitate reuse of the next-hop association, leading to the corrected prediction in the example shown in Figure 1(D). IERE aligns representations according to a dependency-aware schedule, then applies batched factual editing to establish direct associations. It uses only single-hop edit requests, requiring no multi-hop questions, next-hop relations, reasoning paths, or target answers.

Extending this mechanism to batched editing introduces an additional challenge. When the desired object of one request is also edited as the subject of another, an upstream request may align with an outdated entity representation. IERE therefore constructs a dependency graph from subject–object matches within the edit batch and schedules representation updates from downstream to upstream, ensuring that each request uses the latest representation of its desired object. This ordering also recursively carries downstream-conditioned representations toward upstream edits, making dependency-aware scheduling essential for extending representation alignment to longer chains.

We evaluate IERE on Llama-3-8B-Instruct and GPT-J-6B using two- to four-hop questions and batches of up to 100 edits. IERE preserves direct editing performance comparable to established parameter-editing methods and achieves the highest multi-hop accuracy among single-hop-only

methods in all settings. Without edit-specific multi-hop questions, next-hop relations, reasoning chains, or target answers, it ranks first in four settings and second in the other two. We further evaluate its generality, stability, and the contributions of representation alignment, dependency-aware scheduling, and factual editing. Our contributions are summarized as follows:

- We show via layer-wise hidden-state interventions that introducing edited-object information at the subject position can redirect multi-hop prediction toward direct next-hop recall, identifying intermediate-entity representation as an intervention-sensitive factor in edit propagation.
- We propose a multi-hop knowledge editing method named IERE, which combines intermediate-entity representation alignment, dependency-aware scheduling, and factual editing using only single-hop edit requests.
- Experiments across two models and multiple hop lengths show that IERE substantially improves multi-hop propagation using only single-hop edit requests while maintaining reliable direct editing and subject-level locality comparable to MEMIT.

2 Related Work

Knowledge Editing. Knowledge editing methods can be broadly divided into parameter-preserving approaches, which keep the backbone model fixed and incorporate edited knowledge through external or auxiliary memories (Mitchell et al., 2022b; Wang et al., 2024), and parameter-modifying approaches, which update model weights through learned editors or direct parameter editing (De Cao et al., 2021; Mitchell et al., 2022a; Meng et al., 2022; 2023; Li et al., 2024; Fang et al., 2025; Zhang et al., 2025b). Recent work further extends parameter editing to sequential, lifelong, and long-form settings (Li et al., 2025; Jiang et al., 2025a;b; Li & Chu, 2025). IERE belongs to direct parameter editing. **Multi-Hop Editing and Representation Intervention.** MQuAKE (Zhong et al., 2023) and RippleEdits (Cohen et al., 2024) evaluate whether edited knowledge propagates to related predictions. Existing multi-hop methods either introduce external knowledge, retrieval, or inference-time reasoning (Zhang et al., 2024a; Lu et al., 2025; Lee et al., 2025; Shi et al., 2026; Guo et al., 2026; Liu et al., 2025), or directly modify model parameters through multi-hop localization, attribution, or residual-knowledge erasure (Zhang et al., 2024b; Yang et al., 2026; Zhang et al., 2025a), often relying on auxiliary multi-hop information. In parallel, prior work views Transformer MLPs as key–value memories and associates factual recall with layers, neurons, and computational pathways (Geva et al., 2021; 2022; Dai et al., 2022; Hernandez et al., 2024; Meng et al., 2022; Yu & Ananiadou, 2024). Following causal tracing and activation patching studies (Meng et al., 2022; Zhang & Nanda, 2024; Wang et al., 2023), IERE uses hidden-state intervention to motivate a parameter-editing method that requires only single-hop edit requests and explicitly handles dependencies among interacting edits.

3 Problem Formulation

Let $p_\theta(y \mid x)$ denote the conditional probability assigned by a language model with parameters θ. A single-hop factual edit is represented as $\epsilon = (e_1, r_1, e_2, e_2')$, which rewrites $e_1 \xrightarrow{r_1} e_2$ as $e_1 \xrightarrow{r_1} e_2'$. Let $x_{\text{edit}} = \tau_{r_1}(e_1)$ denote the corresponding rewrite prompt. Following existing parameter-editing methods (Meng et al., 2022; 2023; Fang et al., 2025), the editing objective is

$$p_{\theta'}\left(e_2' \mid x_{\text{edit}}\right) > p_{\theta'}\left(e_2 \mid x_{\text{edit}}\right). \tag{1}$$

For a batch of N edits $\mathcal{B} = \{\epsilon_i\}_{i=1}^N$, we denote the editing process by $\theta' = \mathcal{E}(\theta, \mathcal{B})$. For an input sequence x, let $\mathbf{h}_\ell(x, t) \in \mathbb{R}^d$ denote the hidden state at Transformer layer ℓ and token position t. We focus on forward multi-hop propagation, where the desired object of the edited fact serves as an intermediate entity:

$$e_1 \xrightarrow{r_1} e_2' \xrightarrow{r_2} e_3. \tag{2}$$

Here, e_2' is both the desired object of the direct edit and the intermediate entity, while e_3 is the downstream target. Let $x_{\text{mh}} = \tau_{r_1:r_2}(e_1)$ denote the corresponding multi-hop input and $x_{\text{next}} = \tau_{r_2}(e_2')$ the direct next-hop prompt. Although successful editing under x_{edit} establishes the desired direct association, it does not ensure that the updated intermediate-entity information influences

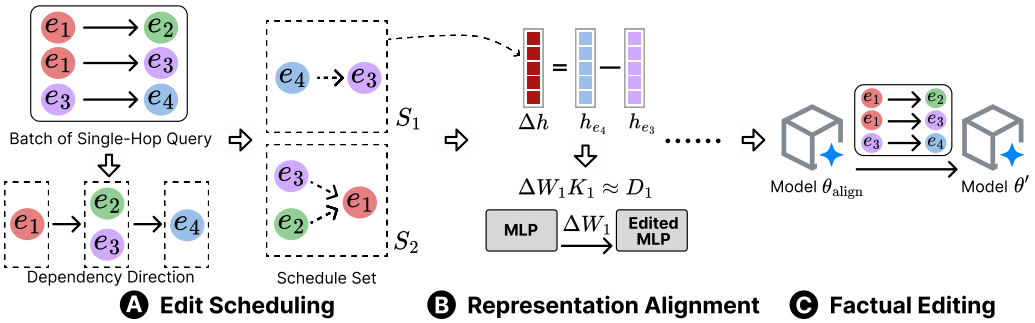

Figure 3: Overview of Intermediate-Entity Representation Editing (IERE). IERE first schedules interacting edits from downstream to upstream, then aligns subject representations with updated desired-object representations, and finally applies batched factual editing.

prediction under x_{mh}. We therefore use x_{next} as a reference for examining whether multi-hop computation moves toward the direct downstream association.

4 Intermediate-Entity Intervention

For the forward chain

$$e_1 \xrightarrow{r_1} e_2' \xrightarrow{r_2} e_3,$$

we test whether introducing the desired-object representation e_2' at the subject positions of the multi-hop input can steer prediction toward direct next-hop recall. We construct a context-averaged representation of e_2' using neutral prompts that contain neither the next-hop relation nor the multi-hop target, and independently patch it into the subject positions at each layer.

We then measure both the target NLL and the movement of the final predictive state toward that induced by direct next-hop recall. Specifically, the alignment score measures whether the intervention-induced state shift follows the direction toward the direct next-hop state, while the progress score measures how far the state advances along that direction. Each layer is intervened independently without modifying model parameters. Full definitions are provided in Appendix A.1.

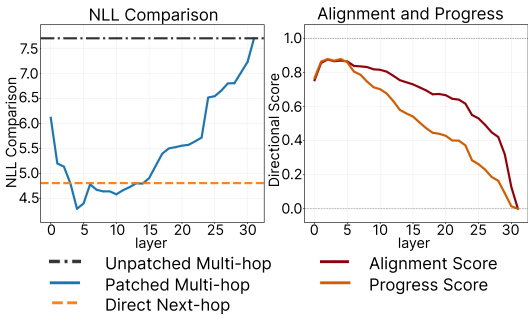

Figure 2: Effect of intermediate-entity intervention on target NLL and final-state movement.

Figure 2 reports mean results over 50 two-hop cases from MQuAKE (Zhong et al., 2023). Earlier-layer interventions substantially reduce the target NLL toward the direct next-hop reference. Consistently, the final predictive state moves toward that induced by direct next-hop recall, while both effects weaken at later layers.

These results show that multi-hop prediction is sensitive to desired-object-aligned information introduced at the subject position, motivating the persistent representation alignment used by IERE.

5 Intermediate-Entity Representation Editing

Based on the intervention results in the previous section, we propose **Intermediate-Entity Representation Editing (IERE)**. As illustrated in Figure 3, IERE first determines the representation-alignment order according to dependencies among single-hop edits, then aligns subject representations with their desired-object representations, and finally applies batched factual editing. These stages play complementary roles: dependency-aware scheduling ensures that each alignment uses

an up-to-date desired-object representation, representation alignment establishes subject states compatible with downstream reuse, and factual editing finally guarantees reliable direct recall. IERE requires only single-hop edit requests and uses no multi-hop question, next-hop relation, reasoning chain, or multi-hop target answer. The complete code is provided in Code and Data Supplement.

5.1 Dependency-Aware Edit Scheduling

Related alignment updates cannot always be processed simultaneously. Consider the following edit batch:

$$e_1 \to e_2, \qquad e_1 \to e_3, \qquad e_3 \to e_4. \tag{3}$$

For clarity, each arrow denotes the subject and desired object of an edit, while relations are omitted. The alignment for $e_3 \to e_4$ changes the representation of e_3, which serves as the alignment target of $e_1 \to e_3$. If the two edits are processed simultaneously, e_1 is aligned with the original representation of e_3, which becomes outdated after $e_3 \to e_4$. Therefore, $e_3 \to e_4$ must be processed first. This downstream-to-upstream ordering also enables longer-hop propagation. After a downstream edit is processed, its updated representation is re-extracted before being used as the alignment target of the next upstream edit. For example, for a longer dependency chain $e_a \to e_b \to e_c \to e_d$, IERE processes $e_c \to e_d$ first, then $e_b \to e_c$, and finally $e_a \to e_b$, so that every upstream alignment uses a representation obtained after its downstream dependency has been updated. Repeating this procedure recursively carries downstream-conditioned representations toward upstream entities, allowing longer dependency chains to be handled using only single-hop edit requests. Detailed execution is provided in Appendix A.2.

We construct a dependency graph whose nodes are the subjects in the edit batch. All edits sharing the same subject are associated with the same node and assigned to the same scheduling round, avoiding repeated updates to that subject across different rounds.

A directed edge $b \to a$ is introduced whenever an edit with subject a has b as its desired object and b is also a subject in the edit batch. The edge means the representation of b must be updated before it is used as the alignment target of a.

At each scheduling round, all edits associated with zero-indegree nodes are collected into $\mathcal{S}_q$, after which these nodes are removed from the graph. Independent edits can therefore be aligned in parallel, while dependent edits are processed from downstream to upstream. Cyclically dependent nodes are merged and assigned to the same round. The resulting schedule is

$$\mathrm{Schedule}(\mathcal{B}) = (\mathcal{S}_1, \ldots, \mathcal{S}_Q)\,, \tag{4}$$

where $\mathcal{S}_q$ contains all edits processed in the q-th scheduling round. Further details are provided in the Appendix A.2.

5.2 Intermediate-Entity Representation Alignment

Given the schedule in Equation 4, we initialize $\theta^{(0)} = \theta$ and process $\mathcal{S}_1, \ldots, \mathcal{S}_Q$ sequentially. At round q, all representations and keys are computed from the current model $\theta^{(q-1)}$; the round index is omitted from token-level representations for simplicity.

For each edit $\epsilon_i = (e_{1,i}, r_{1,i}, e_{2,i}, e'_{2,i}) \in \mathcal{S}_q$, the desired object $e'_{2,i}$ provides the alignment target. Its context-averaged representation $\bar{\mathbf{h}}^{(\ell)}_{e'_{2,i}}$ is computed over C neutral contextual prompts as detailed in Appendix A.1. Importantly, this target is constructed solely from the desired object in the single-hop edit request; no downstream relation, multi-hop question, reasoning path, or multi-hop answer is required for representation alignment. Let $x_i = \tau_{r_{1,i}}(e_{1,i})$ be the rewrite prompt and T_i the subject-token positions. For each $t \in T_i$, the required hidden-state displacement is

$$\Delta\mathbf{h}^{(\ell)}_{i,t} = \bar{\mathbf{h}}^{(\ell)}_{e'_{2,i}} - \mathbf{h}_\ell\left(x_i, t; \theta^{(q-1)}\right), \tag{5}$$

using the same intermediate-entity representation as the target for all subject tokens. Thus, $\Delta\mathbf{h}^{(\ell)}_{i,t}$ directly specifies the representation change required to move each subject-token state toward the desired-object representation observed in the intervention study.

Let $\mathbf{k}_{i,t}^{(\ell)}$ denote the editable-layer MLP key for subject token t. We concatenate the hidden-state displacements and their corresponding keys into matrices $\mathbf{D}_q$ and $\mathbf{K}_q$, respectively. When multiple requests share the same subject, their relation-specific prompts produce separate displacement–key constraints, which are solved jointly within the same scheduling round rather than through repeated sequential updates to the same subject. We then seek an MLP update satisfying $\Delta\mathbf{W}_q\mathbf{K}_q \approx \mathbf{D}_q$. The regularized closed-form solution is

$$\Delta\mathbf{W}_q = \mathbf{D}_q\mathbf{K}_q^T \left(\mathbf{K}_q\mathbf{K}_q^T + \mathbf{R}_q\right)^{-1}, \tag{6}$$

where $\mathbf{R}_q$ follows the MEMIT protection term (Meng et al., 2023) to preserve background knowledge and representations written in earlier rounds. Intuitively, this update converts the temporary intervention in Section 4 into a persistent MLP mapping: when the subject keys are activated, the edited MLP induces a hidden-state displacement toward the desired-object representation. We then update $\mathbf{W}_\ell^{(q)} = \mathbf{W}_\ell^{(q-1)} + \Delta\mathbf{W}_q$, yielding model $\theta^{(q)}$. The next round recomputes its intermediate-entity representations and keys from $\theta^{(q)}$. After all Q rounds, the representation-aligned model is $\theta_{\text{align}} = \theta^{(Q)}$.

5.3 Factual Editing

Intermediate-entity representation alignment makes the desired object available to next-hop relational computation, but does not by itself guarantee reliable direct factual recall. After completing all Q alignment rounds, we therefore apply factual editing once to the complete batch:

$$\theta' = \mathcal{E}_{\text{fact}}\left(\theta_{\text{align}}, \mathcal{B}\right), \tag{7}$$

where $\mathcal{E}_{\text{fact}}$ denotes the MEMIT factual editing operation. All keys and target values required by factual editing are recomputed from θ_{align}, rather than inherited from the original model. Consequently, the factual update is solved under the representation geometry already established by intermediate-entity alignment.

Representation alignment is performed before factual editing so that the subsequent factual update is computed from subject representations already adjusted for downstream reuse. In this sense, alignment first establishes a propagation-compatible representation condition under which the direct factual association is subsequently written.

The execution order is also important because representation alignment and factual editing act on overlapping editable MLP parameters. Representation alignment introduces representation changes that facilitate downstream reuse of the desired object, whereas factual editing writes the direct subject–relation–object association required for reliable single-hop recall. Since the two updates modify overlapping parameter locations, a later update may partially perturb the effect introduced by the earlier one. We therefore apply factual editing last, ensuring that the requested direct association remains reliably established in the final model. Reversing the order would apply representation alignment after the factual association has already been written, potentially perturbing the direct edit. We empirically examine this ordering effect in Section 6.3.

6 Experiments

We organize our experiments around four questions. **RQ1:** Can IERE achieve effective multi-hop propagation using only single-hop edit requests, without edit-specific multi-hop information? **RQ2:** Do its multi-hop gains reflect improved propagation rather than merely stronger direct editing? **RQ3:** Are the dependency-aware schedule, representation alignment, factual editing, and their execution order necessary for IERE? **RQ4:** Does representation alignment preserve other relations associated with the edited subject? We evaluate these questions on Llama-3-8B-Instruct and GPT-J-6B. Additional backward-propagation, global-capability, and failure analyses are reported in the Appendix. All experiments are conducted on a single NVIDIA RTX PRO 6000 GPU with 110 GB of system memory.

6.1 Experimental Setup

Models and Baselines. We evaluate IERE on Llama-3-8B-Instruct and GPT-J-6B. Baselines include the direct parameter-editing methods ROME (Meng et al., 2022), MEMIT (Meng et al.,

2023), and AlphaEdit (Fang et al., 2025), together with the multi-hop parameter-editing methods IFMET (Zhang et al., 2024b) and ACE (Yang et al., 2026). IERE, ROME, MEMIT, and AlphaEdit receive only single-hop edit requests. Following their original protocols, IFMET additionally constructs multi-hop editing prompts, while ACE uses multi-hop factual-recall queries and attribution to identify its editing locations. Thus, IFMET and ACE use edit-specific multi-hop information, whereas IERE does not. Since this work focuses on parameter editing, parameter-preserving methods are not included.

Dataset and Protocol. We use MQuAKE-3K (Zhong et al., 2023) as the primary benchmark and evaluate forward propagation with a batch size of 100. The Random split of RippleEdits (Cohen et al., 2024) is additionally used for subject-level locality, where Relation Specificity (RS) queries test facts that share the edited subject but involve relations different from the edited relation. We retain only RS facts for which the original model prefers the target answer over the alternative answers according to NLL. Additional backward-propagation experiments are reported in Appendix A.3.

Metrics. Single-hop edit requests are factual continuations in which the desired object is expected to immediately follow the prompt. Following ROME, MEMIT, and AlphaEdit (Meng et al., 2022; 2023; Fang et al., 2025), we therefore report the single-hop NLL success rate (**SNll.**), where an edit is successful if the desired object has lower NLL than the original object. Multi-hop evaluation, in contrast, is formulated as question answering, where the target entity need not immediately follow the input and may appear after intermediate or formatting tokens. Following ACE (Yang et al., 2026), we therefore use generation accuracy (**MAcc.**), which considers a generated response correct when it contains the target answer. Thus, the different metrics reflect the corresponding query formulations rather than method-specific evaluation choices. For subject-level locality, we additionally report an NLL-based Relation Specificity preservation rate (**RSNll.**), detailed in Section 6.4. Broader capability stability is evaluated on CSQA and MMLU and reported in Appendix A.5. For representation alignment, we select one layer per model based on the layer-wise intervention analysis in Section 4, using layer 5 for Llama-3-8B-Instruct and layer 8 for GPT-J-6B. These layers are then fixed across all hop lengths and experimental settings.

6.2 Effectiveness and Propagation

Table 1 jointly evaluates **RQ1** and **RQ2** by reporting direct editing success and forward multi-hop accuracy.

RQ1: Multi-Hop Effectiveness without Multi-Hop Information. IERE maintains reliable direct editing, with SNll. above 96% on Llama-3-8B-Instruct and 90% on GPT-J-6B across all settings. More importantly, among methods receiving only single-hop edit requests, IERE achieves the highest MAcc. in all six model–hop settings. On Llama-3-8B-Instruct, it reaches 39.76%, 34.37%, and 21.41% from two to four hops, exceeding the strongest conventional single-hop editor by 31.12, 28.16, and 13.74 percentage points, respectively. IFMET and ACE additionally use edit-specific multi-hop information and are included as multi-hop-informed references. Despite this additional information, IERE ranks first in four forward settings and second in the remaining two. Notably, this advantage extends beyond the two-hop setting used in the intervention study: IERE remains the strongest single-hop-only method on both 3-hop and 4-hop evaluations. This indicates that the representation-alignment mechanism generalizes from the local intervention observation to longer dependency chains.

RQ2: Propagation beyond Direct Editing. The MAcc. gains cannot be explained solely by stronger direct editing. Several baselines already achieve high SNll. but remain substantially lower in MAcc.; conversely, IERE does not consistently obtain the highest SNll. on GPT-J-6B while still achieving the highest MAcc. among single-hop-only methods in every hop setting. Thus, reliable direct editing and downstream reuse are distinct: IERE improves the ability of edited knowledge to participate in subsequent multi-hop prediction rather than merely strengthening the direct edit.

6.3 Method Design Validation

We use Table 2 to answer **RQ3** and validate the design choices underlying IERE. **FULL** denotes the complete method; **NoSched** removes dependency-aware scheduling and processes all edits in a single round; **NoFact** retains scheduling and representation alignment but removes the final factual-

Table 1: Forward propagation results on MQuAKE (%). † denotes methods whose original configurations rely on auxiliary multi-hop information. Bold: The best results. Underlined: The second-best results.

Method	2-hop		3-hop		4-hop	
	SNll.	MAcc.	SNll.	MAcc.	SNll.	MAcc.
Forward: Llama-3-8B-Instruct						
ROME	22.34	2.13	17.89	2.22	28.35	0.00
MEMIT	94.08	8.51	94.09	5.10	<u>96.41</u>	7.35
AlphaEdit	91.42	8.64	93.20	6.21	94.41	7.67
IFMET†	**98.40**	<u>19.68</u>	**98.67**	7.54	95.21	<u>18.53</u>
ACE†	79.52	7.85	78.49	<u>11.31</u>	77.16	14.06
IERE	<u>96.68</u>	**39.76**	<u>96.01</u>	**34.37**	**97.28**	**21.41**
Forward: GPT-J-6B						
ROME	4.06	4.79	2.88	3.10	4.47	6.07
MEMIT	<u>97.54</u>	17.69	<u>92.02</u>	5.54	94.17	8.95
AlphaEdit	97.01	25.00	**92.46**	5.10	93.53	8.31
IFMET†	**98.01**	**30.45**	90.91	<u>7.10</u>	**96.17**	**13.42**
ACE†	86.04	9.04	85.14	3.55	87.86	8.31
IERE	96.54	<u>25.40</u>	90.24	**8.20**	<u>95.21</u>	<u>9.90</u>

editing stage; and **F→A** reverses the default order by applying factual editing before representation alignment.

Dependency-Aware Scheduling. Removing dependency-aware scheduling leaves direct editing largely unchanged but consistently reduces MAcc. On Llama-3-8B-Instruct, MAcc. drops from 39.76%, 34.37%, and 21.41% to 23.01%, 7.54%, and 4.79% from two to four hops, with the same trend on GPT-J-6B. This separation between relatively stable SNll. and degraded MAcc. shows that scheduling is not primarily responsible for establishing the direct facts; rather, it is critical for ensuring that upstream edits use updated intermediate-entity representations and for recursively carrying downstream-conditioned states toward upstream edits.

Roles of Representation Alignment and Factual Editing. NoFact reveals distinct roles for the two stages. On Llama-3-8B-Instruct, representation alignment alone retains comparable multi-hop accuracy but reduces SNll. from above 96% to approximately 55%, showing that alignment can establish representations favorable to downstream reuse without reliably writing the requested direct association. On GPT-J-6B, removing factual editing reduces both SNll. and MAcc., showing that representation alignment alone is insufficient to maintain either reliable direct recall or propagation on this model. Overall, the two stages are complementary: representation alignment supports downstream reuse, while factual editing is required for reliable direct recall.

Execution Order. The ordering of the two stages is deliberate. Alignment is performed first so that the factual update is optimized from subject representations already adjusted for downstream reuse. At the same time, representation alignment and factual editing modify overlapping editable MLP parameters, so a later update can perturb information introduced by the earlier one. Under F→A, applying alignment last yields higher MAcc. on Llama-3-8B-Instruct, reaching 44.02%, 51.22%, and 24.28%, but reduces SNll. to 62.94%–66.08%, indicating that the later alignment can disturb the direct association written by the factual editor. In contrast, under the default A→F order, factual editing is applied last, maintaining SNll. above 96% while retaining propagation gains. On GPT-J-6B, reversing the order degrades both direct editing and propagation. These results support first establishing a propagation-compatible representation condition through alignment and then finalizing the requested factual association through factual editing.

6.4 Subject-Level Locality

RQ4 examines whether IERE preserves other factual relations associated with the edited subject while performing representation alignment. We evaluate this using the full Random split of RippleEdits (Cohen et al., 2024). Its Relation Specificity (RS) queries share the edited subject but involve relations different from the edited relation, providing a targeted measure of subject-level

Table 2: Method-design validation on Llama-3-8B-Instruct and GPT-J-6B (%).

Method	2-hop		3-hop		4-hop	
	SNll.	MAcc.	SNll.	MAcc.	SNll.	MAcc.
Llama-3-8B-Instruct						
FULL	96.68	39.76	96.01	34.37	97.28	21.41
NoSched	98.34	23.01	94.83	7.54	96.88	4.79
NoFact	55.52	38.03	55.88	35.48	54.23	23.96
F→A	65.29	44.02	66.08	51.22	62.94	24.28
GPT-J-6B						
FULL	96.54	25.40	90.24	8.20	95.21	9.90
NoSched	99.00	20.08	94.38	5.10	96.57	3.19
NoFact	33.38	4.12	31.93	6.87	25.56	2.56
F→A	36.17	4.39	19.73	2.88	20.45	8.31

locality after editing. We retain only RS facts for which the original model prefers the target answer over the alternative answers according to NLL. Table 3 reports SNll. on the corresponding direct edits and RSNll., the proportion of retained RS facts whose absolute target-NLL change is at most 1 after editing. We report SNll. together with RSNll. because locality should be interpreted jointly with direct-edit success: preserving existing relations is less informative if the requested edit itself is not reliably established. The two metrics therefore characterize direct-edit reliability and subject-level preservation together.

Table 3: Single-hop editing and subject-level locality on the Random split of RippleEdits (%). SNll. measures direct editing success, while RSNll. measures preservation of initially recognized facts that share the edited subject but involve different relations.

Model	Method	SNll.↑	RSNll.↑
Llama-3-8B-Instruct	MEMIT	91.18	83.72
	ACE	80.03	88.37
	IERE	90.54	82.92
GPT-J-6B	MEMIT	98.80	47.07
	ACE	92.46	60.47
	IERE	97.43	53.01

On Llama-3-8B-Instruct, IERE remains close to MEMIT in both direct editing (90.54% vs. 91.18%) and RSNll. (82.92% vs. 83.72%). On GPT-J-6B, it similarly maintains comparable direct editing (97.43% vs. 98.80%) while achieving a higher RSNll. than MEMIT (53.01% vs. 47.07%). Overall, IERE maintains subject-level locality comparable to MEMIT while preserving similar direct editing performance. ACE achieves higher RSNll. on both models but lower direct editing success, particularly on Llama-3-8B-Instruct. Broader capability preservation on CSQA and MMLU is reported in Appendix A.5.

7 Conclusion

We propose Intermediate-Entity Representation Editing (IERE) to improve the reuse of edited knowledge in multi-hop prediction using only single-hop edit requests. Layer-wise interventions show that multi-hop prediction is sensitive to desired-object-aligned information at the subject position, motivating persistent representation alignment. IERE combines dependency-aware downstream-to-upstream alignment with batched factual editing, substantially improving forward multi-hop propagation while maintaining reliable direct editing and subject-level locality comparable to MEMIT. These results show that edited knowledge can be made more reusable in downstream multi-hop prediction without requiring edit-specific multi-hop questions, next-hop relations, reasoning paths, or target answers.

AI USE STATEMENT

We used generative AI tools solely to improve the language, clarity, and readability of the manuscript. These tools were not used to formulate research hypotheses, design the methodology or experiments, implement the proposed method, analyze data, or interpret experimental results. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide implementation details, intervention definitions, dependency-aware scheduling procedures, and additional experimental analyses in the Appendix. The complete implementation and experimental configurations are included in the anonymous Code and Data Supplement. We use publicly available models and benchmarks, and report the model, dataset, evaluation protocol, and editing settings used in our experiments.

REFERENCES

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Trans. Assoc. Comput. Linguistics*, 12:283–298, 2024. doi: 10.1162/TACL_A_00644.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022. doi: 10.18653/V1/2022.ACL-LONG.581.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506. Association for Computational Linguistics, November 2021. doi: 10.18653/v1/2021.emnlp-main.522. URL https://aclanthology.org/2021.emnlp-main.522/.

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.446.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.3.

Jiaxin Guo, Hao Sun, Wenhao Zhang, Xuanbo Fan, and Yan Zhang. Mitigating error accumulation in knowledge editing for multi-hop question answering. In *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 30862–30870, 2026. doi: 10.1609/AAAI.V40I37.40344.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=ADtL6fgNRv.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Mingyang Wan, Guojun Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Anyedit: Edit any knowledge encoded in language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267. PMLR / OpenReview.net, 2025a. URL https://proceedings.mlr.press/v267/jiang25b.html.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, Baolong Bi, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. Neuron-level sequential editing for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 16678–16702, 2025b. doi: 10.18653/v1/2025.acl-long.815.

Tianjie Ju, Yijin Chen, Xinwei Yuan, Zhuosheng Zhang, Wei Du, Yubin Zheng, and Gongshen Liu. Investigating multi-hop factual shortcuts in knowledge editing of large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8987–9001, 2024. doi: 10.18653/V1/2024.ACL-LONG.486.

Jaewook Lee, Dahyun Jung, and Heuiseok Lim. Stepke: Stepwise knowledge editing for multi-hop question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7752–7765, 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 (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 4127–4149, 2025. doi: 10.18653/v1/2025.acl-long.208.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18564–18572, 2024. doi: 10.1609/aaai.v38i17.29818.

Zherui Li, Houcheng Jiang, Hao Chen, Baolong Bi, Zhenhong Zhou, Fei Sun, Junfeng Fang, and Xiang Wang. Reinforced lifelong editing for language models. In *Proceedings of International Conference on Machine Learning*, volume 267, 2025.

Yi Liu, Xiangrong Zhu, Xiangyu Liu, Wei Wei, and Wei Hu. Avoiding knowledge edit skipping in multi-hop question answering with guided decomposition. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics*, pp. 16256–16272, 2025. doi: 10.18653/V1/2025.FINDINGS-EMNLP.883.

Yifan Lu, Yigeng Zhou, Jing Li, Yequan Wang, Xuebo Liu, Daojing He, Fangming Liu, and Min Zhang. Knowledge editing with dynamic knowledge graphs for multi-hop question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24741–24749, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Proceedings of Annual Conference on Neural Information Processing Systems*, 2022.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In *ICML*, pp. 15817–15831, 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Jinwei Shi, Wenxuan Huang, Yu Xing, Yunhui Liu, Tao Zheng, Bin Chong, and Tieke He. EchoEdit: Consistent multi-hop question answering via ripple control in knowledge editing. In *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 19362–19370, 2026. doi: 10.1609/AAAI.V40I23.39013.

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 *Proceedings of International Conference on Learning Representations*, 2023.

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. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/60960ad78868fce5c165295fbd895060-Abstract-Conference.html.

Jiayu Yang, Yuxuan Fan, Songning Lai, Shengen Wu, Jiaqi Tang, Chun Kang, Zhijiang Guo, and Yutao Yue. ACE: Attribution-controlled knowledge editing for multi-hop factual recall. In *Proceedings of International Conference on Learning Representations*, 2026.

Zeping Yu and Sophia Ananiadou. Neuron-level knowledge attribution in large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 3267–3280, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.191.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *Proceedings of International Conference on Learning Representations*, 2024.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. Knowledge graph enhanced large language model editing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 22647–22662, 2024a.

Mengqi Zhang, Bowen Fang, Qiang Liu, Xiaotian Ye, Shu Wu, Pengjie Ren, Zhumin Chen, and Liang Wang. KELE: residual knowledge erasure for enhanced multi-hop reasoning in knowledge editing. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24537–24552, 2025a. doi: 10.18653/V1/2025.FINDINGS-EMNLP.1334.

Tianyu Zhang, Junfeng Fang, Houcheng Jiang, Baolong Bi, Xiang Wang, and Xiangnan He. Explainable and efficient editing for large language models. In *THE WEB CONFERENCE 2025*, 2025b. URL https://openreview.net/forum?id=iAn7rlIfgc.

Zhuoran Zhang, Yongxiang Li, Zijian Kan, Keyuan Cheng, Lijie Hu, and Di Wang. Locate-then-edit for multi-hop factual recall under knowledge editing. *arXiv preprint arXiv:2410.06331*, 2024b.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.971.

A Appendix

A.1 Intermediate-Entity Intervention

This section provides additional details of the intermediate-entity intervention introduced in the main text. We consider the two-hop chain

$$e_1 \xrightarrow{r_1} e_2' \xrightarrow{r_2} e_3, \tag{8}$$

where e_2' is the desired object of the single-hop edit and the intermediate entity required by the multi-hop question. The multi-hop input x_{mh} asks for e_3 without explicitly containing e_2', whereas x_{next} directly queries the existing association $e_2' \xrightarrow{r_2} e_3$. All interventions are performed on the original model without modifying its parameters.

Intermediate-Entity Representation. To reduce the influence of any particular context, we place the intermediate entity e_2' in C contextual prompts. Let $x_{e_2'}^{(c)}$ denote the c-th prompt and $T_{e_2'}^{(c)}$ the

token positions occupied by e_2'. At layer ℓ, we average the hidden states over all entity tokens and contextual prompts:

$$\bar{\mathbf{h}}_{e_2'}^{(\ell)} = \frac{\sum_{c=1}^{C} \sum_{t \in T_{e_2'}^{(c)}} \mathbf{h}_\ell \left(x_{e_2'}^{(c)}, t \right)}{\sum_{c=1}^{C} \left| T_{e_2'}^{(c)} \right|}. \tag{9}$$

The contextual prompts contain neither the next-hop relation r_2, the multi-hop target e_3, nor the complete multi-hop question. Thus, the representation is constructed using only the desired object provided by the single-hop edit request.

Layer-Wise Subject Intervention. Let T_s denote the token positions occupied by subject e_1 in x_{mh}. At each layer ℓ, we replace the hidden state of every subject token with the mean intermediate-entity representation:

$$\mathbf{h}_\ell \left(x_{\mathrm{mh}}, t \right) \leftarrow \bar{\mathbf{h}}_{e_2'}^{(\ell)}, \qquad \forall t \in T_s. \tag{10}$$

All non-subject positions remain unchanged, and the modified states are passed through the remaining Transformer layers normally. Each layer is evaluated independently using the original model. We use $\widetilde{x}_{\mathrm{mh}}^{(\ell)}$ as shorthand for the computation under intervention at layer ℓ; the textual input itself remains unchanged.

Target NLL. For an input x and target answer y, we define the negative log-likelihood as

$$\mathcal{L}(x, y) = -\log p_\theta(y \mid x), \tag{11}$$

where lower NLL indicates higher probability assigned to the target answer.

We compare three conditions:

$$\begin{aligned} \mathcal{L}_{\mathrm{int}}^{(\ell)} &= \mathcal{L} \left(\widetilde{x}_{\mathrm{mh}}^{(\ell)}, e_3 \right), \\ \mathcal{L}_{\mathrm{mh}} &= \mathcal{L} \left(x_{\mathrm{mh}}, e_3 \right), \\ \mathcal{L}_{\mathrm{next}} &= \mathcal{L} \left(x_{\mathrm{next}}, e_3 \right). \end{aligned} \tag{12}$$

Here, $\mathcal{L}_{\mathrm{int}}^{(\ell)}$ is the target NLL after intervention at layer ℓ, $\mathcal{L}_{\mathrm{mh}}$ is the original multi-hop NLL, and $\mathcal{L}_{\mathrm{next}}$ is the direct next-hop NLL. Since $\mathcal{L}_{\mathrm{mh}}$ and $\mathcal{L}_{\mathrm{next}}$ do not depend on the intervention layer, they serve as reference values in Figure 2.

When $\mathcal{L}_{\mathrm{int}}^{(\ell)} < \mathcal{L}_{\mathrm{mh}}$, the intervention increases the probability of the multi-hop target. As $\mathcal{L}_{\mathrm{int}}^{(\ell)}$ approaches $\mathcal{L}_{\mathrm{next}}$, the intervened prediction approaches the direct factual recall available from the intermediate entity.

Final Predictive States. We further examine whether the NLL improvement is accompanied by movement of the final predictive representation toward that of the direct next-hop input. Let L denote the final Transformer layer and $t_{\mathrm{out}}(x)$ the final input position used to predict the first answer token. We define

$$\mathbf{z}(x) = \mathbf{h}_L \left(x, t_{\mathrm{out}}(x) \right). \tag{13}$$

The final predictive states under the three conditions are

$$\begin{aligned} \mathbf{z}_{\mathrm{mh}} &= \mathbf{z} \left(x_{\mathrm{mh}} \right), \\ \mathbf{z}_{\mathrm{int}}^{(\ell)} &= \mathbf{z} \left(\widetilde{x}_{\mathrm{mh}}^{(\ell)} \right), \\ \mathbf{z}_{\mathrm{next}} &= \mathbf{z} \left(x_{\mathrm{next}} \right). \end{aligned} \tag{14}$$

Taking the original multi-hop state as a common origin, we define

$$\begin{aligned} \mathbf{d}_{\mathrm{int}}^{(\ell)} &= \mathbf{z}_{\mathrm{int}}^{(\ell)} - \mathbf{z}_{\mathrm{mh}}, \\ \mathbf{d}_{\mathrm{next}} &= \mathbf{z}_{\mathrm{next}} - \mathbf{z}_{\mathrm{mh}}. \end{aligned} \tag{15}$$

Here, $\mathbf{d}_{\mathrm{int}}^{(\ell)}$ is the intervention-induced displacement, while $\mathbf{d}_{\mathrm{next}}$ is the direction from the original multi-hop state toward the direct next-hop state.

Alignment. The alignment score measures whether the intervention moves the final predictive state toward the direct next-hop direction:

$$\mathrm{Align}^{(\ell)} = \frac{\mathbf{d}_{\mathrm{int}}^{(\ell)T}\mathbf{d}_{\mathrm{next}}}{\left\|\mathbf{d}_{\mathrm{int}}^{(\ell)}\right\|_2 \left\|\mathbf{d}_{\mathrm{next}}\right\|_2}. \tag{16}$$

This is the cosine similarity between the intervention-induced displacement and the target direction. A value close to 1 indicates movement in the same direction as the direct next-hop state, a value near 0 indicates approximately orthogonal movement, and a negative value indicates movement in the opposite direction. Alignment therefore measures direction rather than distance.

Progress. The progress score measures how far the intervention advances along the target direction:

$$\mathrm{Prog}^{(\ell)} = \frac{\mathbf{d}_{\mathrm{int}}^{(\ell)T}\mathbf{d}_{\mathrm{next}}}{\left\|\mathbf{d}_{\mathrm{next}}\right\|_2^2}. \tag{17}$$

A value of 0 indicates no projected movement toward the direct next-hop state, values between 0 and 1 indicate partial progress, and 1 means that the projection reaches the direct next-hop reference. Values above 1 indicate movement beyond the reference, while negative values indicate movement away from it.

Alignment and progress provide complementary information. High alignment with low progress indicates movement in the correct direction but by a small distance, whereas high values for both indicate substantial movement toward the direct next-hop computation.

Results. We apply the intervention independently at every layer and report mean results over 50 two-hop cases from MQuAKE (Zhong et al., 2023). As shown in Figure 2, early- and middle-layer interventions substantially reduce target NLL relative to the original multi-hop input and move it toward the direct next-hop reference. The effect weakens at later layers.

The representation-space results are consistent: early-layer interventions achieve high alignment and progress scores, showing that the final predictive state moves both directionally and substantially toward the direct next-hop state. Both scores decrease at later layers, suggesting that interventions applied after much of the relational computation has been completed have less opportunity to redirect the prediction.

Together, these results show that multi-hop prediction is sensitive to desired-object-aligned information introduced at the subject position: the intervention increases the target probability and moves the final predictive state toward direct next-hop recall. This intervention does not establish that the model explicitly recovers the desired object as a symbolic intermediate reasoning step, nor does it rule out the coexistence of factual shortcuts. We therefore interpret intermediate-entity representation as an intervention-sensitive factor associated with edit propagation.

A.2 Dependency-Aware Edit Scheduling

This section describes how IERE identifies dependencies among single-hop edit requests and converts them into an execution schedule. Given an edit batch

$$\mathcal{B} = \{\epsilon_i\}_{i=1}^{N}, \qquad \epsilon_i = \left(e_{1,i}, r_{1,i}, e_{2,i}, e'_{2,i}\right), \tag{18}$$

$e_{1,i}$ is the subject of the i-th edit, $r_{1,i}$ its relation, and $e_{2,i}$ and $e'_{2,i}$ the original and desired objects, respectively.

Why Scheduling Is Required. Dependencies arise when the desired object of one edit is also modified as the subject of another. Consider

$$e_1 \to e_2, \qquad e_1 \to e_3, \qquad e_3 \to e_4. \tag{19}$$

The edit $e_3 \to e_4$ changes the representation of e_3. If all alignment targets are extracted before any update, $e_1 \to e_3$ uses the original representation of e_3, which becomes outdated after $e_3 \to e_4$. IERE therefore processes $e_3 \to e_4$ first and re-extracts the representation of e_3 before aligning $e_1 \to e_3$.

Subject Groups. We group all edits sharing the same subject:

$$\mathcal{G}_a = \{\epsilon_i \in \mathcal{B} \mid e_{1,i} = a\}, \tag{20}$$

where $\mathcal{G}_a$ contains all edit requests whose subject is entity a. Requests in the same group may express different relations and desired objects but are always processed in the same round, avoiding repeated updates to the same subject across stages.

For the example in Equation equation 19,

$$\begin{aligned} \mathcal{G}_{e_1} &= \{e_1 \to e_2,\, e_1 \to e_3\}, \\ \mathcal{G}_{e_3} &= \{e_3 \to e_4\}. \end{aligned} \tag{21}$$

Dependency Graph. We construct a directed graph

$$\mathcal{D} = (\mathcal{V}, \mathcal{A}), \tag{22}$$

where each node in $\mathcal{V}$ represents a nonempty subject group. For an edit $\epsilon_i \in \mathcal{G}_a$, if its desired object is the subject of another group $\mathcal{G}_b$, we add

$$e'_{2,i} = b \quad \Longrightarrow \quad \mathcal{G}_b \to \mathcal{G}_a. \tag{23}$$

The direction follows the required execution order: $\mathcal{G}_b$ is processed first because its update may change the desired-object representation required by $\mathcal{G}_a$.

In the running example, e_3 is both the desired object of $e_1 \to e_3$ and the subject of $e_3 \to e_4$, yielding

$$\mathcal{G}_{e_3} \longrightarrow \mathcal{G}_{e_1}. \tag{24}$$

No dependency is introduced for $e_1 \to e_2$ because e_2 is not edited as a subject elsewhere in the batch.

Scheduling Rounds. The graph is processed from downstream to upstream. At each step, all zero-indegree nodes are assigned to the same round and removed from the graph. Since they have no unresolved next-hop dependencies, they can be processed together. Repeating this procedure yields

$$(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_Q), \tag{25}$$

where $\mathcal{S}_q$ contains all edits assigned to round q.

For the running example,

$$\begin{aligned} \mathcal{S}_1 &= \{e_3 \to e_4\}, \\ \mathcal{S}_2 &= \{e_1 \to e_2,\, e_1 \to e_3\}. \end{aligned} \tag{26}$$

Thus, e_3 is first aligned with e_4. Before the second round, IERE re-extracts the representation of e_3 from the updated model and uses it as the target for $e_1 \to e_3$.

Cyclic Dependencies. A cycle may occur when edit groups mutually depend on one another, for example,

$$e_a \to e_b, \qquad e_b \to e_a. \tag{27}$$

Such a cycle does not admit a linear order in which both desired-object representations are updated beforehand. We therefore merge each strongly connected component into a coupled group and process all of its edits in the same round. This removes cycles from the condensed graph and allows the remaining groups to be topologically scheduled. Within a cyclic group, all target representations are extracted from the same current model state.

Execution with Updated Representations. The schedule is used only for intermediate-entity representation alignment. At round q, all subject and desired-object representations are extracted from the model $\theta^{(q-1)}$ produced by the previous round. After applying the alignment updates associated with $\mathcal{S}_q$, the model is updated to $\theta^{(q)}$, from which the representations required by $\mathcal{S}_{q+1}$ are recomputed.

This iterative re-extraction serves two purposes. First, it prevents an upstream edit from using an outdated desired-object representation. Second, and more importantly for longer-hop propagation,

Algorithm 1 Dependency-Aware Edit Scheduling

Require: Edit batch $\mathcal{B}$
Ensure: Ordered rounds $(\mathcal{S}_1, \ldots, \mathcal{S}_Q)$
Group edits by subject to obtain $\{\mathcal{G}_a\}$
Initialize $\mathcal{D} = (\mathcal{V}, \mathcal{A})$, with one node for each subject group
for each edit $\epsilon_i \in \mathcal{G}_a$ **do**
  Set $b \leftarrow e'_{2,i}$
  if a subject group $\mathcal{G}_b$ exists **then**
    Add edge $\mathcal{G}_b \rightarrow \mathcal{G}_a$
  end if
end for
Merge each strongly connected component into one coupled node
Initialize $q \leftarrow 1$
while the condensed graph is not empty **do**
  Collect all zero-indegree nodes into $\mathcal{Z}_q$
  Set $\mathcal{S}_q$ to all edits contained in $\mathcal{Z}_q$
  Remove $\mathcal{Z}_q$ and its outgoing edges
  Set $q \leftarrow q + 1$
end while
return $(\mathcal{S}_1, \ldots, \mathcal{S}_Q)$

it allows the effect of a downstream alignment update to be recursively reflected in the alignment targets used by upstream edits.

Consider a dependency chain

$$e_1 \rightarrow e_2, \qquad e_2 \rightarrow e_3, \qquad e_3 \rightarrow e_4.$$

IERE first processes $e_3 \rightarrow e_4$, producing an updated model $\theta^{(1)}$. The representation of e_3 used in the next round is then re-extracted from $\theta^{(1)}$, rather than from the original model $\theta^{(0)}$. Therefore, when $e_2 \rightarrow e_3$ is processed, e_2 is aligned with an updated representation of e_3 that already reflects the preceding downstream alignment. This produces $\theta^{(2)}$. The representation of e_2 is subsequently re-extracted from $\theta^{(2)}$ and used as the alignment target of $e_1 \rightarrow e_2$.

Thus, the effect of downstream alignment is progressively carried toward upstream entities through refreshed intermediate-entity representations. The same procedure naturally extends to longer dependency chains: each round uses the representation produced after all of its downstream dependencies have been processed. Consequently, IERE can support longer-hop propagation using only the dependency structure among single-hop edit requests, without explicitly constructing the corresponding multi-hop questions or target answers.

After all Q alignment rounds, IERE applies one batched factual edit to the complete original edit batch. The dependency graph therefore serves both to maintain representation consistency and to recursively propagate the effect of dependent alignment updates from downstream to upstream.

Complexity. Using a mapping from entity names to subject groups, subject grouping and dependency-edge construction are linear in the number of edits and detected dependencies. Strongly connected component detection and topological scheduling are linear in the size of the resulting graph. Their overhead is therefore small relative to the model forward passes and parameter updates required for representation alignment.

A.3 Backward Propagation

Compared with MEMIT, IERE improves MAcc. by 10.99, 9.53, and 5.99 percentage points from two to four hops while maintaining high SNll. It also outperforms IFMET in the 2-hop and 3-hop settings, while remaining lower in the 4-hop setting. These results indicate that the learned updates can generalize beyond the forward direction explicitly targeted by IERE.

Table 4: Backward propagation results on MQuAKE (%).

Method	2-hop		3-hop		4-hop	
	SNll.	MAcc.	SNll.	MAcc.	SNll.	MAcc.
MEMIT	94.69	31.41	97.62	5.44	99.15	5.98
IFMET	97.95	37.00	95.75	6.12	99.43	15.38
IERE	100.00	42.40	99.15	14.97	99.72	11.97

A.4 Failure Analysis

To characterize the remaining errors, we divide failed forward multi-hop cases into three categories. *Negative/ineffective update* means that the direct edit fails to increase preference for the target answer; *Partial propagation* means that the edited information reaches only an intermediate step or shorter subchain; and *No propagation* means that its effect does not extend beyond the direct edit. Table 5 reports their proportions.

Table 5: Distribution of failure types among failed forward multi-hop cases (%).

Hop	Llama-3-8B-Instruct			GPT-J-6B		
	No Prop.	Partial	Neg./Inef.	No Prop.	Partial	Neg./Inef.
2	83.87	6.45	9.68	64.64	12.71	22.65
3	93.08	2.31	4.62	73.37	14.13	12.50
4	89.70	5.45	4.85	73.13	10.13	16.74

No propagation dominates on Llama-3-8B-Instruct, accounting for 83.87%–93.08% of failures, while the other categories remain rare. GPT-J-6B shows a more diverse pattern: no propagation remains the largest category (64.64%–73.37%), but partial propagation and negative or ineffective updates are more frequent. Overall, most failures reflect insufficient propagation after successful direct editing, with GPT-J-6B additionally showing less reliable update establishment and propagation.

A.5 Global Capability Stability

Complementary to the subject-level locality evaluation in Section 6.4, we evaluate whether IERE preserves broader model capabilities unrelated to the edited facts. Table 6 reports CSQA and MMLU accuracy after applying the edit batches associated with the 2-hop, 3-hop, and 4-hop evaluation subsets. Base denotes the corresponding unedited model, while MEMIT and ACE provide stability references for other parameter-editing methods.

On Llama-3-8B-Instruct, IERE largely preserves CSQA performance, with decreases of only 0.22, 1.00, and 1.44 percentage points across the three hop settings. The degradation is more noticeable on MMLU, ranging from 3.16 to 4.45 points and exceeding that of MEMIT and ACE. Thus, IERE retains most unrelated capabilities on Llama-3-8B-Instruct, although its stronger multi-hop propagation introduces a moderate stability cost. On GPT-J-6B, the stability cost is more pronounced. The CSQA decrease ranges from 1.89 to 4.11 points, while the MMLU decrease ranges from 4.57 to 5.62 points, generally larger than those of the baselines. This suggests that the current representation-alignment procedure is less stable on GPT-J-6B. Nevertheless, the degradation shows no monotonic trend with hop length, indicating that it is not determined solely by dependency length. These global capability results are complementary to the subject-level locality evaluation in Section 6.4. RSNll. specifically measures whether other relations associated with an edited subject remain stable, whereas CSQA and MMLU evaluate broader capabilities that are not tied to the edited entities. The comparable RSNll. of IERE and MEMIT therefore suggests that the global degradation observed here should not be interpreted as systematic corruption of other relations associated with the edited subject.

At the same time, the larger CSQA and MMLU changes, particularly on GPT-J-6B, indicate that representation alignment may still introduce broader parameter interference that is not captured by

relation-specific locality. Improving such global stability remains an important direction for future work. A detailed analysis of the remaining forward multi-hop failures is provided in Appendix A.4.

Table 6: Stability results on Llama-3-8B-Instruct and GPT-J-6B (%).

Method	2-hop		3-hop		4-hop	
	CSQA	MMLU	CSQA	MMLU	CSQA	MMLU
Llama-3-8B-Instruct						
Base	80.33	70.18	80.33	70.18	80.33	70.18
MEMIT	80.33	69.12	79.89	69.71	80.00	70.64
ACE	80.00	68.65	78.56	68.30	77.33	69.47
IERE	80.11	67.02	79.33	65.73	78.89	66.67
GPT-J-6B						
Base	22.00	30.53	22.00	30.53	22.00	30.53
MEMIT	21.00	31.70	20.56	30.41	21.33	32.05
ACE	21.22	32.51	21.22	29.94	19.89	29.01
IERE	17.89	25.38	19.00	25.96	20.11	24.91