RESOLVING HARD PARAPHRASE-UNRELATED QUERY REGIMES IN LIFELONG MODEL EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Lifelong model editing requires Reliability on edited knowledge, Generality over semantically equivalent paraphrases, and Locality on unrelated queries. In memory-based editors, these objectives depend on routing each query to either the edit-specific path or the base-model path correctly. We identify **hard paraphrase-unrelated query regimes**, where paraphrases are lexically distant from the edit query, unrelated queries are lexically near it, or both. Although defined by surface distance, these regimes exhibit the same distance trend in hidden-state representation space across evaluated backbones, increasing routing-score overlap between paraphrase and unrelated queries and thereby making simultaneous Generality and Locality difficult for memory-based routers. We propose **SEAL** (**S**yn**E**rgistic intr**A**- and extra-mode**L** framework), which combines an intra-model Hopfield-based router with a lightweight extra-model Sentinel Prompt used at test-time inference. The Sentinel Prompt induces an Asymmetric Pulling Effect, suppressing unrelated-query routing scores while maintaining or increasing paraphrase scores, thereby separating the overlapping routing-score distributions. Decoupled Two-Stage Inference removes the Sentinel Prompt on the base-model path to preserve the original output distribution. Experiments on four backbones across four benchmarks show that SEAL improves Generality and Locality in hard regimes.

1 INTRODUCTION

Despite the impressive capabilities of Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2023; Wei et al., 2022), the dynamic evolution of real-world knowledge renders their internalized knowledge susceptible to obsolescence (Zhang et al., 2023; Jang et al., 2022; Kim et al., 2024). Given the prohibitive costs of full model retraining (Strubell et al., 2019; Bender et al., 2021), Lifelong Model Editing has emerged as a crucial paradigm for efficient, continual knowledge updates (Hartvigsen et al., 2023; Wang et al., 2024a). An ideal system must balance three core objectives: accurately incorporating updated knowledge (Reliability), correctly answering diverse rephrasings of the edited fact (Generality), and preserving pre-existing knowledge on unrelated queries (Locality) (Yao et al., 2023; Mitchell et al., 2022a; De Cao et al., 2021). These objectives are evaluated per editing instance, which consists of an edit query accessible to the editor at edit time, together with paraphrase and unrelated queries that are held out and revealed only at test time.

Memory-based editors address this by storing each edit in external memory and routing each incoming test query either to an edit-specific path that applies the selected edit memory, or to a base-model path that bypasses edit memories and preserves the frozen model's pre-edit behavior (Hartvigsen et al., 2023; Yu et al., 2024; Wang et al., 2024a; Mitchell et al., 2022b). Routing decisions are made in hidden-state representation space, where a routing score is obtained by comparing the query representation with stored edit representations, so that queries close to a stored edit in hidden-state representation space are routed to the edit memory, whereas queries far from all stored edits fall through to the base-model path. This design is scalable and non-destructive, but its correctness critically depends on the router's ability to correctly route edit/paraphrase queries to the corresponding edit-specific path and unrelated queries to the base-model path (Hsueh et al., 2024; Wang et al., 2024c). More details about lifelong model editing and memory-based editors are in **Related Work** (Appendix A).

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.

However, not all editing instances are equally easy to route. We identify two data properties that stress routing reliability: (1) paraphrase queries that are lexically distant from the edit query, and (2) unrelated queries that are lexically near it. Taking these two surface distances as axes, editing instances can be grouped into four quadrants, as illustrated in Figure 1. Instances where the edit–paraphrase surface distance is small and the edit–unrelated surface distance is large (the second quadrant Q2) are relatively straightforward to route; the remaining three quadrants (Q1, Q3 and Q4) featuring larger edit–paraphrase surface distance, smaller edit–unrelated surface distance, or both, pose substantially greater routing difficulty, and we collectively term them the **hard paraphrase-unrelated query regimes**.

Critically, this is a property of the data rather than of any particular model, and it indicates where routing is more likely to fail in hidden-state representation space. Our analysis (Section 4.3) shows that the hidden-state representation distances used by the router largely follow the same trend as the corresponding surface distances. In hard regimes, paraphrase queries tend to lie farther from the edit, while unrelated queries tend to lie closer, in hidden-state representation space. This lowers paraphrase routing scores and raises unrelated-query scores, causing overlap in routing-score distributions. Consequently, memory-based routers have greater difficulty distinguishing paraphrases from unrelated queries and routing both correctly, making simultaneous Generality and Locality harder to achieve in hard regimes (Section 4.2).

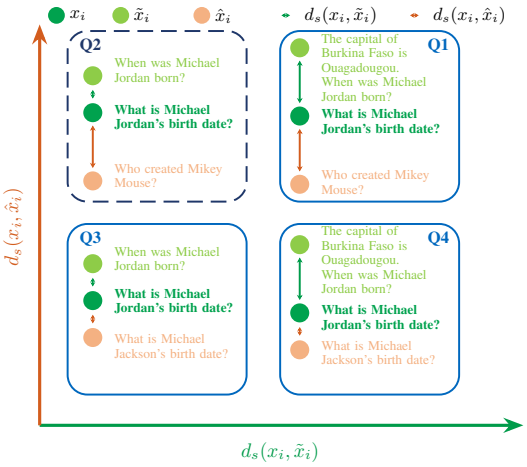


Figure 1: Four regimes are defined on the two axes $d_s(x_i, \tilde{x}_i)$ and $d_s(x_i, \hat{x}_i)$. Here x_i, $\tilde{x}_i$, and $\hat{x}_i$ denote the edit, paraphrase, and unrelated queries, and d_s denotes surface distance. Q2 (low $d_s(x_i, \tilde{x}_i)$, high $d_s(x_i, \hat{x}_i)$) is the easy regime. The remaining quadrants Q1, Q3, and Q4 form hard paraphrase-unrelated query regimes, where $d_s(x_i, \tilde{x}_i)$ is relatively high, $d_s(x_i, \hat{x}_i)$ is relatively low, or both.

To address routing failures in hard regimes, we propose **SEAL** (**S**yn**E**rgistic intr**A**- and extra-mode**L** framework), which enhances routing from two complementary directions. Internally, we adopt a Hopfield-based router as a highly discriminative routing mechanism following HoReN (Fang et al., 2026). More importantly, externally, we introduce a lightweight, continuously optimized Sentinel Prompt at the input level that reshapes query hidden-state representations used by the router. It is prepended before test queries only during test-time inference, not during editing or memory construction. The Sentinel Prompt exhibits a striking Asymmetric Pulling Effect (shown in Section 4.4): routing scores of lexically near unrelated queries are drastically suppressed, while those of lexically distant paraphrase queries remain unchanged or increase, making the two categories substantially easier to distinguish. To prevent the Sentinel Prompt from perturbing frozen-base-model generation on unrelated queries, SEAL employs a Decoupled Two-Stage Inference: the forward pass is split into a routing stage, where the Sentinel Prompt is prepended to select the correct key–value pair or the base-model path, and a generation stage that strips it only for queries routed to the base-model path while queries routed to an edit-specific path can decode with it.

We evaluate SEAL on four backbone models across four benchmarks: WikiBigEdit, CounterFact, ZsRE-hard (a Q1 dataset constructed from ZsRE), and ZsRE. SEAL simultaneously improves Generality and Locality under hard regimes while maintaining performance on the easy regime, and remains effective at large editing scales (10**k** edits). Our contributions are as follows:

- We identify hard paraphrase-unrelated query regimes as a data-level property that indicates where memory-based routing is more likely to fail in hidden-state representation space.
- We propose SEAL, a synergistic intra- and extra-model framework for improving hard-regime Generality and Locality. It couples an intra-model Hopfield router with a lightweight extra-model Sentinel Prompt used only during test-time inference.
- We demonstrate that SEAL achieves simultaneous improvements in Generality and Locality across four backbone models, four benchmarks covering all hard regimes and the easy regime, and lifelong editing scales of up to 10**k** edits, with little sacrifice in edit accuracy.

2 PROBLEM FORMULATION

Following prior work on lifelong model editing (Hartvigsen et al., 2023; Wang et al., 2024a), let $f_{\theta_0} : \mathcal{X} \to \mathcal{Y}$ denote a pretrained language model. A stream of T edits provides pairs (x_i, y_i^*); after step i,

$$f_{\theta_i} = \mathrm{Edit}(f_{\theta_{i-1}}, x_i, y_i^*). \tag{1}$$

We focus on memory-based editors (Hartvigsen et al., 2023; Yu et al., 2024; Wang et al., 2024a; Fang et al., 2026): the backbone remains frozen, the first i edits are stored in external memory $\mathcal{M}_i$ attached to layer l, and a router uses layer $l-1$ representations to send each query either to the edit-specific path or to the base-model path.

After all edits, f_{θ_T} is evaluated on aligned triplets $(x_i, \tilde{x}_i, \hat{x}_i)$: raw edit queries $\{x_i\}_{i=1}^T$ for ***Reliability***, paraphrases $\{\tilde{x}_i\}_{i=1}^T$ sharing target y_i^* for ***Generality***, and unrelated queries $\{\hat{x}_i\}_{i=1}^T$ whose outputs should remain unchanged for ***Locality***. The latter two query types are held out at edit time.

2.1 HARD PARAPHRASE-UNRELATED QUERY REGIMES

We identify hard paraphrase-unrelated query regimes directly from text using *surface distance*:

$$d_s(a, b) \triangleq \frac{\mathrm{Lev}(a, b)}{\mathrm{len}(a) + \mathrm{len}(b)} \in [0, 1], \tag{2}$$

where Lev is the token-level Levenshtein distance (Levenshtein, 1966; Wagner & Fischer, 1974) between lowercased whitespace-tokenized strings a and b. For instance i, the two axes are $d_s(x_i, \tilde{x}_i)$ and $d_s(x_i, \hat{x}_i)$: the former increases when the paraphrase is lexically distant from the edit query, making recall harder, while the latter decreases when an unrelated query is lexically near the edit query, making false activation more likely.

For empirical grouping, we split these two axes by their pooled medians over all evaluated instances, yielding four quadrants Q1–Q4 (Figure 1). Q2, with low $d_s(x_i, \tilde{x}_i)$ and high $d_s(x_i, \hat{x}_i)$, is the easy regime. Q1, Q3, and Q4, where the $d_s(x_i, \tilde{x}_i)$ is larger, the $d_s(x_i, \hat{x}_i)$ is smaller, or both, constitute the hard paraphrase-unrelated query regimes.

2.2 ROUTING FAILURE IN HIDDEN-STATE REPRESENTATION SPACE

Routing failure occurs in the hidden-state representation space used by the router. Let $z^j(x)$ denote the output hidden states of query x after layer j. For an edit at layer l, the sentence-level hidden-state representation used by router is extracted from layer $l-1$'s outputs $z^{l-1}(x)$ following HoReN (Fang et al., 2026):

$$h_r(x) \triangleq \mathrm{Pool}_\rho\left(z^{l-1}(x)\right), \tag{3}$$

where Pool_ρ mean-pools the last $\rho\%$ token positions. We measure hidden-state representation space distance by the cosine distance (Salton et al., 1975):

$$d_r(a, b) \triangleq 1 - \cos(h_r(a), h_r(b)). \tag{4}$$

The relevant distances are $d_r(x_i, \tilde{x}_i)$ and $d_r(x_i, \hat{x}_i)$. As Section 4.3 shows, d_r follows the same trend as d_s: instances in hard regimes have larger $d_r(x_i, \tilde{x}_i)$, smaller $d_r(x_i, \hat{x}_i)$, or both. Consequently, paraphrase scores decrease while unrelated scores increase, increasing overlap of the two routing-score distributions. As a result, the router has greater difficulty routing paraphrases and unrelated queries correctly, making simultaneous Generality and Locality harder to achieve.

3 SEAL

We propose **SEAL** (**S**yn**E**rgistic intr**A**- and extra-mode**L** framework), which addresses routing failure in hard regimes from two complementary directions. Sections 3.1 and 3.2 present its intra-/extra-model components: a Hopfield router and a Sentinel Prompt for reshaping hidden-state representations used by the router. Section 3.3 describes how the Sentinel Prompt is trained to induce the Asymmetric Pulling Effect, and Section 3.4 details its use in Decoupled Two-Stage Inference. SEAL keeps edit-time memory construction consistent with HoReN (Fang et al., 2026), and focuses its main methodological changes on Sentinel Prompt training and test-time inference.

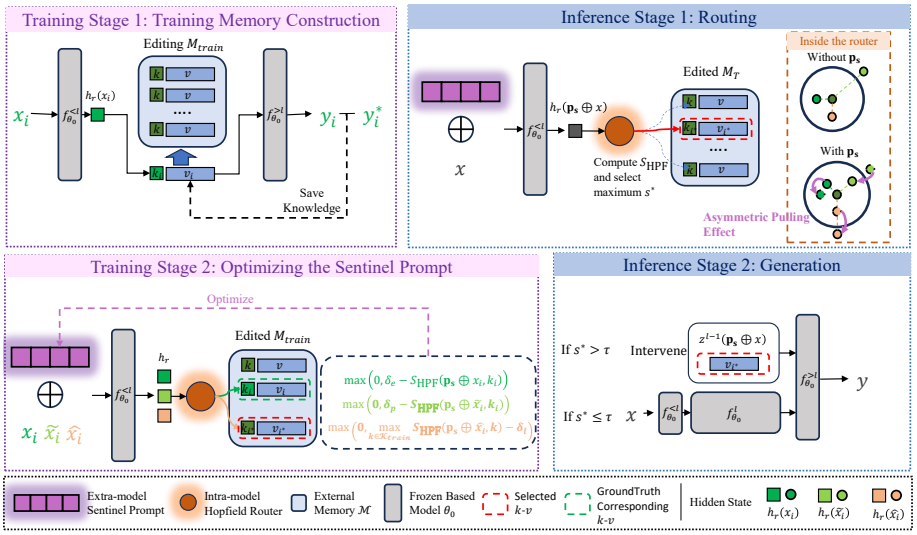


Figure 2: Overview of SEAL training and inference. **Training Stage 1:** Training edits are saved without $\mathbf{p}_s$, producing $\mathcal{M}_{\text{train}}$. **Training Stage 2:** With the backbone and $\mathcal{M}_{\text{train}}$ frozen, $\mathbf{p}_s$ is optimized to keep edit/paraphrase queries close to their corresponding keys and suppress unrelated queries against all keys. **Inference Stage 1:** $\mathbf{p}_s$ is prepended for routing; the Hopfield router scores $h_r(\mathbf{p}_s \oplus x)$ and selects the maximum score s^*. The inset illustrates the resulting Asymmetric Pulling Effect. **Inference Stage 2:** If $s^* > \tau$, SEAL injects the selected value and continues decoding with the prompted state; otherwise, it reruns the frozen model on the original input x without $\mathbf{p}_s$.

3.1 Intra-model Component: Hopfield Router

Following HoReN, we use a Hopfield router as the intra-model routing mechanism. **Crucially, SEAL is not tied to the Hopfield router. It's also effective with other routers** (Appendix F.5). After T edits, the memory $\mathcal{M}_T = \{(k_i, v_i)\}_{i=1}^T$ at edit layer l is constructed. $k_i = h_r(x_i)$ is the routing key of the i-th edit and v_i is the corresponding edit value storing the edited knowledge. Let $\mathcal{K}_T = \{k_i\}_{i=1}^T$. For an input query x, the router extracts $h_r(x)$ from $z^{l-1}(x)$ (Eq. 3) and refines it by

$$\bar{h}_r(x) = \phi_{\text{HPF}}(h_r(x),\, \mathcal{K}_T)\,, \tag{5}$$

where ϕ_{HPF} is the modern Hopfield update (Ramsauer et al., 2021). For each memory key k_i, the routing score is $S_{\text{HPF}}(x, k_i)$. The selected score s^* is the maximum routing score:

$$\begin{aligned} S_{\text{HPF}}(x, k_i) &= \cos\big(\bar{h}_r(x), k_i\big)\,, \\ i^* = \arg\max_i S_{\text{HPF}}(x, k_i), &\qquad s^* = S_{\text{HPF}}(x, k_{i^*}). \end{aligned} \tag{6}$$

The routing decision is made based on the selected score s^* and a hyper-parameter threshold τ. If $s^* > \tau$, the router selects the edit value v_{i^*} for injection; otherwise, the query follows the base-model path. The full SEAL inference procedure is defined in Section 3.4.

3.2 Extra-model Component: Sentinel Prompt

To reshape routing hidden-state representations, we introduce a lightweight extra-model continuous prompt $\mathbf{p}_s \in \mathbb{R}^{m \times d}$ of m trainable soft tokens at input level, termed the **Sentinel Prompt**. $\mathbf{p}_s$ is not used at editing time. Therefore, the edit memory remains the unprompted $\mathcal{M}_T$. Instead, after the memory is constructed, $\mathbf{p}_s$ is prepended only during test-time inference, as illustrated in Inference Stage 1 of Figure 2:

$$x^+ = \mathbf{p}_s \oplus x. \tag{7}$$

The query hidden-state representation becomes $h_r(x^+)$, producing prompted routing scores $S_{\text{HPF}}(x^+, k_i)$. The Sentinel Prompt is trained to induce an **Asymmetric Pulling Effect**, as shown in the two router insets of Figure 2:

- For edit and paraphrase queries $(x_i, \tilde{x}_i)$: routing scores against their associated memory key k_i are maintained above τ (drawn closer to their associated keys), preserving Reliability and Generality even when $d_s(x_i, \tilde{x}_i)$ is large.

- For unrelated queries $\hat{x}_i$: routing scores against all memory keys k_i are suppressed below τ (pushed away), preserving Locality even when $d_s(x_i, \hat{x}_i)$ is small.

3.3 Sentinel Prompt Training

We train $\mathbf{p}_s$ to induce the Asymmetric Pulling Effect through the following two-stage procedure.

Stage 1: Training Memory Construction. We reserve a subset of each benchmark as the Sentinel Prompt training set containing N editing instances, disjoint from the editing instances used to construct $\mathcal{M}_T$. As Training Stage 1 of Figure 2 shows, we first edit the model on these training instances using the Hopfield router *without* $\mathbf{p}_s$, constructing Training Memory $\mathcal{M}_{\text{train}} = \{(k_i, v_i)\}_{i=1}^{N}$.

Stage 2: Optimizing the Sentinel Prompt. As illustrated in Training Stage 2 of Figure 2, we randomly initialize $\mathbf{p}_s$ and optimize it based on $\mathcal{M}_{\text{train}}$. For each training triplet $(x_i,\, \tilde{x}_i,\, \hat{x}_i)$, we prepend $\mathbf{p}_s$ to each query, run a forward pass to obtain Hopfield routing scores against the key set $\mathcal{K}_{\text{train}}$ with θ_0 and $\mathcal{M}_{\text{train}}$ frozen, and update only $\mathbf{p}_s$ via $\mathcal{L}_{\text{sentinel}}$:

$$\mathcal{L}_{\text{edit}} = \max(0, \delta_e - S_{\text{HPF}}(\mathbf{p}_s \oplus x_i, k_i))\,, \quad \mathcal{L}_{\text{para}} = \max(0, \delta_p - S_{\text{HPF}}(\mathbf{p}_s \oplus \tilde{x}_i, k_i))\,,$$
$$\mathcal{L}_{\text{unrel}} = \max\left(0, \max_{k \in \mathcal{K}_{\text{train}}} S_{\text{HPF}}(\mathbf{p}_s \oplus \hat{x}_i, k) - \delta_l\right) \tag{8}$$

$$\mathcal{L}_{\text{sentinel}} = \mathcal{L}_{\text{edit}} + \mathcal{L}_{\text{para}} + \mathcal{L}_{\text{unrel}} \tag{9}$$

where k_i denotes the stored memory key corresponding to the current training x_i or $\tilde{x}_i$ in Stage 1. $\delta_e, \delta_p, \delta_l$ are margin hyper-parameters. The first two terms keep edit and paraphrase queries' routing scores against their corresponding keys above the routing threshold, while the third suppresses false activation on unrelated queries. Together, these objectives induce the Asymmetric Pulling Effect.

In deployment, the editor may only observe edit-side data (x_i, y_i^*), without future paraphrase or unrelated queries for training $\mathbf{p}_s$. Appendix H introduces an Automated Training Data Synthesis pipeline that takes only edit data as input; we denote synthetic-supervision variant as SEAL_{SYN}.

Once trained, $\mathbf{p}_s$ generalizes to new edits within the same benchmark distribution, including edits entirely excluded from training, removing the need for per-edit retraining. Algorithmic details for Sentinel Prompt training are provided in Appendix G.

3.4 Decoupled Two-Stage Inference

After $\mathbf{p}_s$ is trained, SEAL performs test-time inference over the constructed memory $\mathcal{M}_T = \{(k_i, v_i)\}_{i=1}^{T}$, with $\mathbf{p}_s$ prepended before the query x. Since a soft prompt can perturb frozen-base-model generation, SEAL separates routing from generation into two stages to preserve unrelated-query behavior while retaining the routing benefit of $\mathbf{p}_s$. See the Inference Stage in Figure 2.

Stage 1 (Routing). Prepend $\mathbf{p}_s$ and perform a forward pass up to the edit layer. With $x^+ = \mathbf{p}_s \oplus x$, the selected routing score s^* is

$$i^* = \arg\max_i S_{\text{HPF}}(x^+,\, k_i), \qquad s^* = S_{\text{HPF}}(x^+,\, k_{i^*}). \tag{10}$$

The score is computed from $h_r(x^+)$, which is extracted from the prompted $z^{l-1}(x^+)$.

Stage 2 (Generation). Use the selected routing score s^* from Stage 1 to choose the generation branch. Let $f_{\theta_0}^{>l}$ denote the frozen suffix from layer $l+1$ onward. $\text{Intervene}(\cdot)$ denotes the edit value injection. Then

$$f_{\theta_T}(x) = \begin{cases} f_{\theta_0}^{>l}\big(\text{Intervene}(z^{l-1}(x^+),\, v_{i^*})\big) & \text{if } s^* > \tau \\ f_{\theta_0}(x) & \text{otherwise} \end{cases} \tag{11}$$

If $s^* > \tau$, SEAL injects the selected edit value v_{i^*} into the prompted $z^{l-1}(x^+)$ cached in Stage 1 and directly continues inference with $\mathbf{p}_s$ from layer $l+1$ onward. If $s^* \leq \tau$, SEAL treats the query as unrelated: it discards the prompted state, removes $\mathbf{p}_s$, and restarts from the original input x for clean $\mathbf{p}_s$-free inference, exactly recovering $f_{\theta_0}(x)$. This branch-specific design follows our empirical observation in Appendix F.4 that $\mathbf{p}_s$ does not substantially perturb generation after the edit value is injected, but can significantly change the frozen base model's output on the base-model path. Therefore, only queries judged unrelated are recomputed without $\mathbf{p}_s$. Algorithmic details for the two-stage inference procedure are provided in Appendix G.

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

Backbone Models. We conduct experiments on four backbone language models: Llama-3.1-8B-Instruct (Team, 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), DeepSeek-R1-Distill-Llama-8B, and DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI, 2025), spanning three model families (Llama, Mistral, and Qwen) and multiple parameter scales (7B, 8B, and 32B). For brevity, we refer to them as Llama-3.1-8B, Mistral-7B, DS-Llama-8B, and DS-Qwen-32B, respectively.

Datasets. We evaluate on four structured lifelong model editing benchmarks with distinct surface-distance characteristics, collectively covering all cases analyzed in Section 2. To better analyze and visualize the relative distribution and routing difficulty of instances across benchmarks, we split the $d_s(x_i, \tilde{x}_i)$ and $d_s(x_i, \hat{x}_i)$ axes by their pooled medians over all evaluated instances, yielding four quadrants. Figure 3a visualizes $1\mathbf{k}$ instances from each benchmark under this median-based grouping. **ZsRE** (Levy et al., 2017) has relatively small $d_s(x_i, \tilde{x}_i)$ and large $d_s(x_i, \hat{x}_i)$, so its instances mainly fall in Q2, the easy regime. **ZsRE-hard** is constructed from ZsRE (Appendix I.1) by increasing $d_s(x_i, \tilde{x}_i)$ while preserving relatively large $d_s(x_i, \hat{x}_i)$, placing it mainly in Q1. **WikiBigEdit** (Thede et al., 2025) has relatively small $d_s(x_i, \tilde{x}_i)$ and small $d_s(x_i, \hat{x}_i)$, so it mainly occupies Q3. **CounterFact** (Meng et al., 2022) combines relatively large $d_s(x_i, \tilde{x}_i)$ with relatively small $d_s(x_i, \hat{x}_i)$, placing most instances in Q4, with a notable portion in Q1. More detailed benchmark characteristics are provided in Appendix D.1.

Baselines. We compare SEAL against eight baselines. Non-memory-based methods: FT–L (Zhu et al., 2020; Meng et al., 2022), MEMIT-M (Meng et al., 2023), AlphaEdit (Fang et al., 2025); Memory-based methods: DEFER (Mitchell et al., 2022b; Hartvigsen et al., 2023), WISE (Wang et al., 2024a), GRACE (Hartvigsen et al., 2023), MELO (Yu et al., 2024), HoReN (Fang et al., 2026). Among these, HoReN is a strong baseline and also serves as the main ablation counterpart to SEAL: it uses the same Hopfield router without $\mathbf{p}_s$.

Metrics. Following prior work (Hartvigsen et al., 2023; Wang et al., 2024a; Fang et al., 2025), we evaluate three core properties. **Reliability** (Rel.) measures the proportion of edits for which the model correctly outputs the updated answer on the original edit query. **Generality** (Gen.) measures the success rate on semantically equivalent paraphrases of the edit query. **Locality** (Loc.) measures the proportion of unrelated queries whose model output remains unchanged after editing. We report the harmonic mean of these three metrics as **Overall Performance** (OP). Appendix D.2 formalizes these metrics in detail.

Implementation Details. For SEAL, the $\mathbf{p}_s$ length is set to $m = 4$ soft tokens. We edit zero-indexed layer 29 (56 for DS-Qwen-32B). $h_r(x)$ is obtained with $\rho = 60$, mean-pooling the hidden states of the last 60% token positions before the edit layer. $\mathbf{p}_s$ is trained for 1–3 epochs with a learning rate of 0.001. More hyper-parameter details are in Appendix D.3.

For each benchmark, we reserve a subset for Sentinel Prompt training, disjoint from the editing instances for evaluation. Specifically, for ZsRE and ZsRE-hard we use the first $1\mathbf{k}$ instances of the `mend_eval` split for training and the `mend_train` split for evaluation; for CounterFact we use the first $1\mathbf{k}$ instances of the `test` split for training and the `train` split for evaluation; for WikiBigEdit we use the first $1\mathbf{k}$ instances for training, augmented by LLM-based synthesis as described in Appendix I.2, and use instances starting from index $4\mathbf{k}$ for evaluation.

4.2 MAIN RESULTS

Table 1 compares SEAL with baselines at $T = 1\mathbf{k}$ on Llama-3.1-8B, Mistral-7B, and DS-Qwen-32B; DS-Llama-8B results are deferred to Appendix F.1. Non-memory-based FT and MEMIT-M have unstable Reliability and low OP, while AlphaEdit often improves Generality at the cost of Locality. Among memory-based editors, the main trend is that performance degrades in hard regimes. DEFER often fails to produce meaningful results. For WISE, GRACE, and MELO, Table 1 shows the same regime pattern: their best OP is achieved on the easy ZsRE benchmark, while their OPs degrade on the three hard-regime benchmarks. This degradation is most visible for HoReN, the strongest baseline: although it is nearly perfect on the easy ZsRE regime, on Llama-3.1-8B its OP drops from 0.979 on ZsRE to 0.776/0.364/0.751 on WikiBigEdit/CounterFact/ZsRE-hard, and

Table 1: Main results at editing scale $T = 1\text{k}$. We report Reliability (Rel.), Generality (Gen.), Locality (Loc.), and Overall Performance (OP). Bold marks the best OP per backbone and benchmark; blue shading identifies SEAL.

Backbone	Method	WikiBigEdit (Thede et al., 2025)				CounterFact (Meng et al., 2022)				ZsRE-hard (Levy et al., 2017)				ZsRE (Levy et al., 2017)			
		Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP
Llama-3.1-8B	FT-L	0.084	0.065	0.009	0.022	0.020	0.005	0.000	0.000	0.127	0.083	0.002	0.006	0.127	0.119	0.002	0.006
	MEMIT-M	0.565	0.501	0.229	0.369	0.218	0.149	0.331	0.210	0.826	0.713	0.400	0.587	0.826	0.737	0.400	0.592
	AlphaEdit	0.855	0.712	0.281	0.489	0.980	0.699	0.601	0.729	0.971	0.829	0.437	0.663	0.971	0.877	0.437	0.673
	DEFER	0.173	0.105	0.971	0.184	0.052	0.046	0.469	0.070	0.074	0.013	0.859	0.033	0.074	0.055	0.859	0.091
	WISE	0.140	0.119	0.713	0.177	0.348	0.190	0.354	0.274	0.192	0.061	0.998	0.133	0.277	0.229	0.998	0.334
	GRACE	0.994	0.201	0.951	0.427	0.997	0.071	0.984	0.186	0.997	0.015	1.000	0.044	0.997	0.225	1.000	0.465
	MELO	0.903	0.249	0.914	0.483	0.919	0.107	0.951	0.261	0.848	0.200	0.994	0.418	0.848	0.377	1.000	0.621
	HoReN	0.987	0.753	0.655	0.776	0.999	0.478	0.194	0.364	0.998	0.525	0.920	0.751	0.998	0.942	0.999	0.979
	SEAL	0.965	0.805	0.852	**0.869**	0.969	0.852	0.919	**0.911**	0.974	0.933	1.000	**0.968**	0.998	0.974	1.000	**0.991**
Mistral-7B	FT-L	0.464	0.395	0.358	0.401	0.241	0.142	0.098	0.140	0.465	0.417	0.529	0.466	0.465	0.451	0.529	0.479
	MEMIT-M	0.186	0.162	0.134	0.158	0.114	0.105	0.068	0.091	0.542	0.407	0.277	0.379	0.542	0.451	0.277	0.391
	AlphaEdit	0.688	0.564	0.208	0.373	0.891	0.608	0.529	0.644	0.835	0.683	0.417	0.593	0.835	0.738	0.417	0.606
	DEFER	0.003	0.003	0.922	0.004	0.070	0.048	0.122	0.069	0.001	0.011	0.919	0.003	0.001	0.000	0.919	0.000
	WISE	0.422	0.300	0.244	0.306	0.671	0.402	0.052	0.129	0.385	0.333	0.999	0.454	0.385	0.335	0.999	0.456
	GRACE	0.981	0.112	0.945	0.273	0.994	0.045	0.966	0.124	0.988	0.017	1.000	0.049	0.988	0.197	1.000	0.423
	MELO	0.836	0.169	0.889	0.364	0.923	0.081	0.953	0.207	0.866	0.243	0.997	0.478	0.866	0.382	1.000	0.629
	HoReN	0.980	0.767	0.558	0.729	0.997	0.368	0.335	0.447	0.991	0.684	0.876	0.830	0.991	0.946	0.993	0.976
	SEAL	0.956	0.798	0.816	**0.851**	0.911	0.755	0.819	**0.823**	0.986	0.921	0.974	**0.959**	0.990	0.956	1.000	**0.982**
DS-Qwen-32B	HoReN	1.000	0.784	0.728	0.822	0.999	0.295	0.747	0.524	1.000	0.748	0.817	0.842	1.000	0.975	0.994	0.990
	SEAL	1.000	0.842	0.820	**0.880**	0.988	0.750	0.814	**0.839**	0.999	0.960	0.964	**0.974**	1.000	0.984	0.998	**0.994**

Table 2: Results at editing scale $T = 10\text{k}$ on Llama-3.1-8B.

Method	WikiBigEdit				CounterFact				ZsRE-hard				ZsRE			
	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP
WISE	0.214	0.208	0.329	0.240	0.173	0.093	0.306	0.152	0.183	0.108	0.993	0.191	0.191	0.166	0.990	0.245
GRACE	0.991	0.124	0.972	0.297	0.986	0.041	0.970	0.113	0.996	0.021	1.000	0.060	0.996	0.187	1.000	0.408
HoReN	0.987	0.754	0.556	0.725	0.991	0.360	0.080	0.184	0.995	0.436	0.790	0.657	0.995	0.915	0.978	0.961
SEAL	0.965	0.801	0.790	**0.845**	0.943	0.754	0.833	**0.836**	0.974	0.873	1.000	**0.946**	0.994	0.948	1.000	**0.980**

the same pattern appears on Mistral-7B and DS-Qwen-32B. Thus, hard-regime degradation occurs across different memory-based editors and backbones.

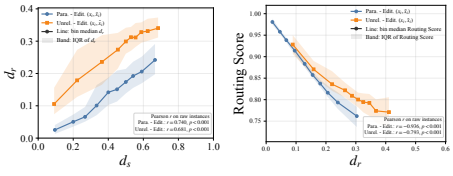
SEAL consistently improves over HoReN across the evaluated hard regimes. On Llama-3.1-8B, it improves HoReN's Gen./Loc. from 0.753/0.655 to 0.805/0.852 on WikiBigEdit, from 0.478/0.194 to 0.852/0.919 on CounterFact, and from 0.525/0.920 to 0.933/1.000 on ZsRE-hard; it also improves OP on the easy ZsRE regime from 0.979 to 0.991. On a larger and stronger DS-Qwen-32B, where we compare only HoReN and SEAL due to computational cost, SEAL continues to achieve improvements. The routing-accuracy analysis in Appendix F.2 further indicates that hard regimes are more prone to routing failures, and that SEAL improves performance by achieving higher routing accuracy for paraphrase and unrelated queries.

Table 2 evaluates scalability at $T = 10\text{k}$ on Llama-3.1-8B. The same hard-regime pattern remains at this larger scale: even HoReN drops to 0.725/0.184/0.657 OP on WikiBigEdit/CounterFact/ZsRE-hard while remaining high on ZsRE (0.961), with GRACE and WISE showing a similar pattern. SEAL again performs best, showing its effectiveness at large editing scales.

Finally, Appendix H.3 provides the detailed results and analysis of SEAL_{SYN}. Trained only on synthetic triplets generated from edit-side queries, it still improves over HoReN on all three hard-regime benchmarks, suggesting the pipeline is effective.

4.3 Analysis of Hard Paraphrase-Unrelated Query Regimes

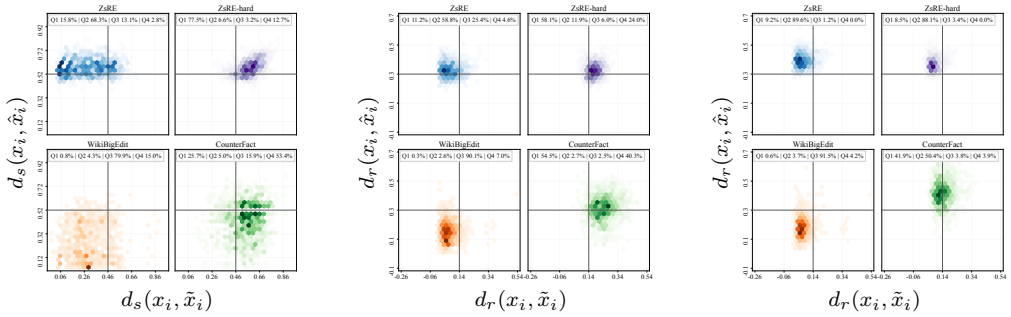
We analyze how hard paraphrase-unrelated query regimes affect routing through d_s, d_r, and routing scores. Unless otherwise stated, all experiments are conducted with the Hopfield router on Llama-3.1-8B and 1k instances per benchmark from the evaluation splits defined in Section 4.1.



(a) d_s–d_r (b) d_r–Routing Score

Figure 4: Pooled binned analyses over 4k evaluation instances on Llama-3.1-8B. (a) Equal-frequency d_s bins with d_r; (b) equal-frequency d_r bins with routing score.

Surface distance distributions. ZsRE mainly falls in the easy regime, whereas the other three benchmarks mainly fall in hard regimes. Figure 3a plots each instance in the $(d_s(x_i, \tilde{x}_i),\ d_s(x_i, \hat{x}_i))$ plane using the pooled medians of the two d_s variables over all instances to split quadrants: ZsRE mainly lies in Q2 (68.3%), ZsRE-hard in Q1 (77.5%), WikiBigEdit in Q3 (79.9%), and CounterFact in Q4 (53.4%) with a notable Q1 portion (25.7%). Thus the bench-



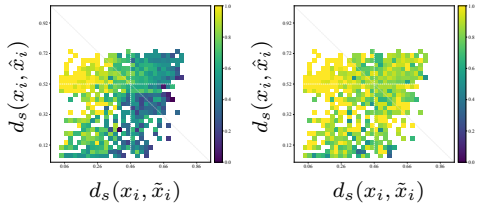
(a) Surface-level distances. (b) Representation space (without $\mathbf{p}_s$). (c) Representation space (with $\mathbf{p}_s$).

Figure 3: Quadrant distributions for 1**k** evaluation instances per benchmark on Llama-3.1-8B. (a) on $(d_s(x_i, \tilde{x}_i),\ d_s(x_i, \hat{x}_i))$ plane. (b) and (c) on $(d_r(x_i, \tilde{x}_i),\ d_r(x_i, \hat{x}_i))$ plane without and with $\mathbf{p}_s$

marks cover the easy regime and all hard regimes. Appendix E.1 analyzes the surface-distance distributions without the quadrant discretization.

Representation distance distributions. The d_r in the hidden-state representation space follows the same trend as d_s for both edit–paraphrase and edit–unrelated pairs. Figure 3b plots each instance in the $(d_r(x_i, \tilde{x}_i),\ d_r(x_i, \hat{x}_i))$ plane without $\mathbf{p}_s$ using the pooled medians of the two d_r variables over all instances to split quadrants, where a similar quadrant structure largely appears in representation space. Figure 4(a) removes the quadrant discretization by pooling all benchmarks and equal-frequency binning query pairs by d_s; the median d_r increases with d_s for both edit–paraphrase and edit–unrelated pairs. Thus, hard regimes tend to induce larger $d_r(x_i, \tilde{x}_i)$, smaller $d_r(x_i, \hat{x}_i)$, or both. Appendix E.2 further verifies the same continuous trend across backbones and layers.

Routing-score overlap. Representation-distance characteristics in hard regimes finally result in routing-score overlap. Figure 4(b) shows that larger d_r is associated with lower routing scores under HoReN. Therefore, in hard regimes, larger $d_r(x_i, \tilde{x}_i)$ tends to lower routing scores for paraphrase queries, while smaller $d_r(x_i, \hat{x}_i)$ tends to raise routing scores for unrelated queries, making the two routing-score distributions harder to separate. Consistently, Figure 6 (top row) shows well-separated paraphrase and unrelated routing-score distributions on ZsRE but substantial overlap on the three hard-regime benchmarks, making it harder for the router to activate the correct edit-specific path for paraphrase queries while rejecting unrelated queries. Appendix E.3 shows this phenomenon across backbones and editors. Figure 5(a) further shows this failure pattern directly: moving toward the lower-right region, where $d_s(x_i, \tilde{x}_i)$ is larger and $d_s(x_i, \hat{x}_i)$ is smaller, the Hopfield router's Para./Loc. routing accuracy without $\mathbf{p}_s$ declines, indicating higher routing-failure risk in hard regimes.



(a) Without $\mathbf{p}_s$ (b) With $\mathbf{p}_s$

Figure 5: Pooled d_s-plane heatmaps over 4**k** evaluation instances on Llama-3.1-8B; color denotes per-bin mean Para./Loc. Hopfield routing accuracy. (a) without $\mathbf{p}_s$; (b) with $\mathbf{p}_s$.

4.4 Analysis of the Asymmetric Pulling Effect

We analyze how $\mathbf{p}_s$ induces the Asymmetric Pulling Effect: it reshapes hidden-state representations used by the router and separates the overlapping paraphrase/unrelated routing-score distributions.

Representation-space reshaping. The Sentinel Prompt reshapes hard-regime query representations toward an easier routing geometry. Figure 3c shows the d_r distribution with $\mathbf{p}_s$, compared with the distribution without $\mathbf{p}_s$. ZsRE-hard and CounterFact move markedly toward Q2 in the d_r plane. WikiBigEdit shifts less visibly, but its Q2 proportion still increases from 2.6% to 3.7%. These shifts make paraphrases closer to their edits and unrelated queries farther away in the representation space, making the two query types easier to distinguish and route correctly.

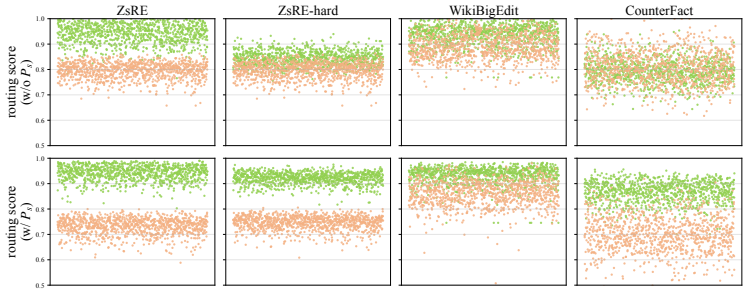


Figure 6: Max Hopfield routing-score distributions for 1**k** evaluation instances per benchmark on Llama-3.1-8B. Green denotes paraphrase queries and Orange denotes unrelated queries. Top row: without $\mathbf{p}_s$. Bottom row: with $\mathbf{p}_s$.

Routing-score separation. $\mathbf{p}_s$ reduces the hard-regime routing-score overlap. Compared with the top row of Figure 6 without $\mathbf{p}_s$, the bottom row shows clearer separation between paraphrase and unrelated routing-score distributions with $\mathbf{p}_s$ on the hard-regime benchmarks: paraphrase scores are maintained or increased, while unrelated scores are suppressed, showing the Asymmetric Pulling Effect. This separation makes it easier for the router to distinguish the two query types and correctly route both paraphrase and unrelated queries. Appendix F.3 provides the detailed mean-score statistics. As a result, Figure 5(b) shows that $\mathbf{p}_s$ improves routing accuracy in hard regimes.

4.5 Ablation and Efficiency Analysis

Decoupled Two-Stage Inference Ablation. We compare two Stage-2 generation policies: continuing from the prompted state for all queries versus discarding the prompted state and rerunning from the original input for all queries. Continuing from the prompted state keeps Reliability and Generality nearly unchanged (0.965/0.805 vs. 0.965/0.806), but substantially hurts Locality (0.552 vs. 0.852) on WikiBigEdit. This supports SEAL's branch-specific design: keep the prompted state after edit value injection, but discard it on the base-model path. Experiment details in Appendix F.4.

Router ablation. With a GRACE-style router on WikiBigEdit, adding a router-specific $\mathbf{p}_s$ raises OP from 0.621 to 0.655. With a BGE-based router on CounterFact, it raises OP from 0.763 to 0.962. Both variants improve Generality and Locality. These results show that the SEAL framework is not tied to the Hopfield router and is effective with other routers. Experiment details in Appendix F.5.

More Analysis. Efficiency analysis in Appendix F.6 shows that SEAL introduces only moderate inference overhead and negligible additional parameters over HoReN, with the extra cost mainly on queries routed to the base-model path. **Threshold sensitivity and edit-only calibration analysis** in Appendix F.7 shows that $\mathbf{p}_s$ improves the Generality–Locality trade-off across thresholds, indicating that the gain comes from better routing-score separation rather than threshold tuning. We also introduce edit-only calibration of SEAL's τ to better reflect deployment; SEAL still substantially outperforms HoReN in hard regimes. **Prompt length and training-size ablations** in Appendix F.8 and F.9 show that a four-token p_s trained on 1**k** instances is sufficient.

5 Conclusion

We identify hard paraphrase-unrelated query regimes as a data-level property that increases routing difficulty in memory-based lifelong model editing. In these regimes, paraphrases that are lexically distant from the edit query and unrelated queries that are lexically near it induce the same trend in hidden-state representation space, causing routing-score overlap between paraphrase and unrelated queries and making simultaneous Generality and Locality difficult for standard routers. SEAL addresses this difficulty by combining an intra-model Hopfield-based router with a lightweight extra-model Sentinel Prompt that separates the two routing-score distributions, and Decoupled Two-Stage Inference that removes the Sentinel Prompt on the base-model path to preserve the frozen model's generation. Across backbones, benchmarks, and editing scales, SEAL improves Generality and Locality in hard regimes while preserving easy-regime performance. Remaining limitations, including extra prefix computation for queries routed to the base-model path and the remaining gap when synthesizing training triplets from edit-side data alone, are detailed in Appendix B.

AI USE STATEMENT

We used generative AI tools for two purposes: to aid and polish writing, and to generate synthetic datasets. For writing, these tools helped improve the clarity and language of author-written text. For data generation, we used Qwen3-MAX to synthesize auxiliary queries for Sentinel Prompt training (Appendices H.2 and I.2) and paraphrase queries for constructing the ZsRE-hard evaluation benchmark (Appendix I.1). All AI-assisted outputs were reviewed and verified by the authors. We take responsibility for the final content of this work, including text and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The main paper specifies the routing formulation, Sentinel Prompt training objective, and two-stage inference procedure in Sections 3.1–3.4, and reports the backbones, benchmarks, baselines, metrics, data splits, and training settings in Section 4.1. The appendices provide metric details and hyperparameters (Appendices D.2 and D.3), algorithmic procedures, the synthetic-data generation instructions (Appendix H.2), and dataset construction details (Appendix I.1 and I.2).

REFERENCES

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In Madeleine Clare Elish, William Isaac, and Richard S. Zemel (eds.), *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pp. 610–623. ACM, 2021. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, et al. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023. URL https://jmlr.org/papers/v24/22-1144.html.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.522. URL https://aclanthology.org/2021.emnlp-main.522/.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=HvSytvg3Jh.

Yuan Fang, Yi Xie, and Xuming Ran. Horen: Normalized hopfield retrieval for large-scale sequential model editing. *CoRR*, abs/2605.08143, 2026. doi: 10.48550/ARXIV.2605.08143. URL https://doi.org/10.48550/arXiv.2605.08143.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with GRACE: lifelong model editing with discrete key-value adaptors. In Alice Oh,

Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/95b6e2ff961580e03c0a662a63a71812-Abstract-Conference.html.

Cheng-Hsun Hsueh, Paul Kuo-Ming Huang, Tzu-Han Lin, Che Wei Liao, Hung-Chieh Fang, Chao-Wei Huang, and Yun-Nung Chen. Editing the mind of giants: An in-depth exploration of pitfalls of knowledge editing in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9417–9429, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.550. URL https://aclanthology.org/2024.findings-emnlp.550/.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun KIM, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vfsRB5MImo9.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

Yujin Kim, Jaehong Yoon, Seonghyeon Ye, Sangmin Bae, Namgyu Ho, Sung Ju Hwang, and Se-Young Yun. Carpe diem: On the evaluation of world knowledge in lifelong language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5401–5415, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.302. URL https://aclanthology.org/2024.naacl-long.302/.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. URL https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In Roger Levy and Lucia Specia (eds.), *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pp. 333–342. Association for Computational Linguistics, 2017. doi: 10.18653/V1/K17-1034. URL https://doi.org/10.18653/v1/K17-1034.

Jiaang Li, Quan Wang, Zhongnan Wang, Yongdong Zhang, and Zhendong Mao. ELDER: enhancing lifelong model editing with mixture-of-lora. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 24440–24448. AAAI Press, 2025. doi: 10.1609/AAAI.V39I23.34622. URL https://doi.org/10.1609/aaai.v39i23.34622.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Zihao Lin, Mohammad Beigi, Hongxuan Li, Yufan Zhou, Yuxiang Zhang, Qifan Wang, Wenpeng Yin, and Lifu Huang. Navigating the dual facets: A comprehensive evaluation of sequential memory editing in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 13755–13772. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.742. URL https://doi.org/10.18653/v1/2024.acl-long.742.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

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 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pp. 15817–15831. PMLR, 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David P. Kreil, Michael K. Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=tL89RnzIiCd.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975. ISSN 0001-0782. doi: 10.1145/361219.361220. URL https://doi.org/10.1145/361219.361220.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355/.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. Wikibigedit: Understanding the limits of lifelong knowledge editing in llms. 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*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/thede25a.html.

Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *J. ACM*, 21 (1):168–173, 1974. doi: 10.1145/321796.321811. URL https://doi.org/10.1145/321796.321811.

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 Globerson, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/60960ad78868fce5c165295fbd895060-Abstract-Conference.html.

Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. EasyEdit: An easy-to-use knowledge editing framework for large language models. In Yixin Cao, Yang Feng, and Deyi Xiong (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 82–93, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.9. URL https://aclanthology.org/2024.acl-demos.9/.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Comput. Surv.*, 57(3), November 2024c. ISSN 0360-0300. doi: 10.1145/3698590. URL https://doi.org/10.1145/3698590.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, et al. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022. URL https://openreview.net/forum?id=yzkSU5zdwD.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-Pack: Packaged resources to advance general Chinese embedding. *CoRR*, abs/2309.07597, 2023. doi: 10.48550/ARXIV.2309.07597. URL https://doi.org/10.48550/arXiv.2309.07597.

Wanli Yang, Fei Sun, Jiajun Tan, Xinyu Ma, Qi Cao, Dawei Yin, Huawei Shen, and Xueqi Cheng. The mirage of model editing: Revisiting evaluation in the wild. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15336–15354, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.745. URL https://aclanthology.org/2025.acl-long.745/.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.632. URL https://aclanthology.org/2023.emnlp-main.632/.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. MELO: enhancing model editing with neuron-indexed dynamic lora. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 19449–19457. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29916. URL https://doi.org/10.1609/aaai.v38i17.29916.

Zihan Zhang, Meng Fang, Ling Chen, Mohammad-Reza Namazi-Rad, and Jun Wang. How do large language models capture the ever-changing world knowledge? a review of recent advances. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8289–8311, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.516. URL https://aclanthology.org/2023.emnlp-main.516/.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. Modifying memories in transformer models. *CoRR*, abs/2012.00363, 2020. URL https://arxiv.org/abs/2012.00363.

A Related Work

A.1 Lifelong Model Editing and Memory-Based Methods

Lifelong Model Editing aims to update factual knowledge in pretrained language models without full retraining, including in settings with long editing sequences. A first line of work directly modifies model parameters. FT-L (Zhu et al., 2020) fine-tunes a selected layer under a norm constraint to localize the update. While effective in single-edit or few-edit settings, FT-L degrades substantially in lifelong model editing, where many sequential updates accumulate interference in the base model. MEMIT (Meng et al., 2023) treats MLP layers as associative memories and performs closed-form updates over selected transformer layers to insert many facts at once. AlphaEdit (Fang et al., 2025) further improves sequential parameter editing by projecting edit perturbations into a null space that preserves previously encoded knowledge. Nevertheless, because these methods still edit the base model parameters, maintaining strong Locality under many sequential edits remains challenging.

Following prior work (Yao et al., 2023), *Memory-based methods* refer to editors that store edited knowledge in external memories or lightweight edit-specific modules and use a routing mechanism to decide whether an incoming query should activate an edit-specific path or fall back to the frozen base-model path. Because edited knowledge is stored outside the base model, these methods are naturally scalable (Li et al., 2025), while the frozen base model makes it possible to achieve high Locality together with high Reliability and Generality (Lin et al., 2024). DEFER (Mitchell et al., 2022b) stores edit examples in an external memory and routes by a learned scope classifier: if the input is judged in-scope for a cached edit, a counterfactual model handles the query; otherwise the query follows the base-model path. WISE (Wang et al., 2024a) stores knowledge in side memory and keeps its base model frozen in main memory, and routes by comparing the activation representations induced by the side memory and the main memory: only when their difference exceeds a threshold is the side memory activated; otherwise the query remains on the main-memory path. GRACE (Hartvigsen et al., 2023) stores edits as discrete key–value adaptors; it retrieves the nearest hidden-state representation key and injects the associated value only when the query falls within that key's learned radius. MELO (Yu et al., 2024) stores edits as neuron-indexed LoRA blocks organized by an inner vector database; its router activates the LoRA block associated with the nearest in-radius cluster, and otherwise uses a zero block to preserve the base model. HoReN (Fang et al., 2026) keeps the same memory-routing perspective with GRACE but replaces one-shot nearest-neighbor retrieval with normalized Hopfield retrieval: normalized keys and queries are refined by a Hopfield update before selecting the best-matching edit for value or LoRA injection.

Across these memory-based editors, success depends on whether the router can route edit and paraphrase queries to the correct edit-specific path while rejecting unrelated queries to the base-model path. However, even a strong router such as HoReN, which performs almost perfectly in the easy regime, still struggles in the hard paraphrase-unrelated query regimes we identify, where paraphrases are lexically distant from the edit query, unrelated queries are lexically near it, or both. SEAL addresses this failure mode by reshaping the routing representations with a Sentinel Prompt at the input level, rather than only changing the router's retrieval architecture or the stored edit module.

A.2 Prompt Tuning and Representation Steering

Prompt-based adaptation provides a lightweight way to steer frozen language models from the input side (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022). Instead of modifying the backbone

parameters, prompt tuning optimizes a small set of continuous input or prefix embeddings that influence the model's internal representations and output behavior. Prior work shows that such continuous prompts can provide effective task adaptation while leaving the pretrained model largely unchanged.

SEAL is inspired by this idea, and its Sentinel Prompt $\mathbf{p}_s$ is essentially a form of soft prompt tuning: it optimizes a small set of trainable soft-token embeddings while keeping the backbone frozen. However, SEAL uses prompting for a different purpose. $\mathbf{p}_s$ is not trained to directly encode edited knowledge or improve generation quality; instead, it is used to improve routing by reshaping the hidden-state representation used by the router, improving the separation between paraphrase and unrelated-query routing scores. Combined with Decoupled Two-Stage Inference, this allows prompt-based control to improve routing without perturbing the frozen base-model generation on queries judged to be unrelated.

B LIMITATIONS

Inference overhead of Decoupled Two-Stage Inference. The extra cost arises only for queries routed to the base-model path. All queries first run a routing forward with $\mathbf{p}_s$ up to the edit layer. Queries routed to the edit-specific path reuse the cached prompted state, inject the edit value, and continue decoding, whereas queries routed to the base-model path must remove $\mathbf{p}_s$ and recompute the prompt-free prefix from x. The overhead therefore comes from recomputing the prompt-free prefix up to layer l, not from a full duplicate generation pass, but it can still be non-negligible in real-world deployments, where most user queries are unrelated to the stored edits and are therefore routed to the base-model path.

Dependence on benchmark annotations for Sentinel Prompt training. The main SEAL setting trains $\mathbf{p}_s$ using benchmark-provided edit–paraphrase–unrelated triplets. In realistic deployments, however, the editor may only observe edit-side data, because future user paraphrases and unrelated queries cannot be known in advance. To reduce this dependence, Appendix H introduces an edit-only synthesis pipeline that constructs training triplets from edit queries alone, yielding SEAL_{SYN}. Although SEAL_{SYN} still substantially improves over HoReN, it does not consistently match the benchmark-supervised SEAL trained with annotated paraphrase and unrelated queries.

C BROADER IMPACTS

This work studies lifelong model editing, which can help keep deployed language models up to date while reducing the cost and environmental footprint of full model retraining. Improved Generality and Locality may also make knowledge updates more reliable by allowing edited facts to generalize to paraphrases while preserving unrelated model behavior.

The same capability can be misused to inject false, biased, or otherwise harmful knowledge into language models, or to make targeted changes that are difficult to notice through limited testing. These risks are especially relevant in high-stakes or user-facing deployments. We therefore recommend using model editing with provenance tracking, validation on held-out paraphrase and unrelated queries, human review for sensitive edits, and monitoring mechanisms that can detect unintended behavioral changes after editing.

D EXPERIMENTAL SETUP DETAILS

D.1 BENCHMARK CHARACTERISTICS

This section expands the benchmark analysis in Section 4.3. All statements use the same surface-distance axes as the main text: $d_s(x_i, \tilde{x}_i)$ for the edit–paraphrase distance and $d_s(x_i, \hat{x}_i)$ for the edit–unrelated distance. Table 3 gives one representative instance from each benchmark.

ZsRE. ZsRE paraphrases are usually direct rewrites of the edit query and retain the main entity tokens, whereas unrelated queries often come from different domains and share little surface form with the edit query. Thus ZsRE has relatively low $d_s(x_i, \tilde{x}_i)$ and high $d_s(x_i, \hat{x}_i)$, and its overall distribution is concentrated in Q2, the easy regime. The ZsRE instance in Table 3 illustrates this

Table 3: Representative instances from each benchmark arranged in the 2×2 quadrant grid. Each cell shows the edit query (x_i), paraphrase query ($\tilde{x}_i$), unrelated query ($\hat{x}_i$), and surface distances. Red text marks distractor prefixes that increase $d_s(x_i, \tilde{x}_i)$; underlined tokens in unrelated queries highlight surface overlap with the edit query that decreases $d_s(x_i, \hat{x}_i)$. $d_s(x_i, \tilde{x}_i)$ and $d_s(x_i, \hat{x}_i)$ are computed by Eq. 2 in the main text.

	Lower $d_s(x_i, \tilde{x}_i)$	**Higher** $d_s(x_i, \tilde{x}_i)$
Higher $d_s(x_i, \hat{x}_i)$	**Q2 — ZsRE (Easy Regime)** **Edit:** The manufacturer of Colt King Cobra was who? **Para:** Which company did Colt King Cobra produce? **Unrel:** ek veer ki ardaas veera meaning in english $d_s(x_i, \tilde{x}_i) = 0.333 \quad d_s(x_i, \hat{x}_i) = 0.500$	**Q1 — ZsRE-hard (Hard Regime)** **Edit:** The manufacturer of Colt King Cobra was who? **Para:** Global supply chains faced disruptions in 2021. By whom was the Colt King Cobra produced? **Unrel:** ek veer ki ardaas veera meaning in english $d_s(x_i, \tilde{x}_i) = 0.565 \quad d_s(x_i, \hat{x}_i) = 0.500$
Lower $d_s(x_i, \hat{x}_i)$	**Q3 — WikiBigEdit (Hard Regime)** **Edit:** In which country is Bernadette Cozart recognized as a citizen? **Para:** In what country is Bernadette Cozart acknowledged as a citizen? **Unrel:** In which country is Bernadette Robert recognized as a citizen? $d_s(x_i, \tilde{x}_i) = 0.100 \quad d_s(x_i, \hat{x}_i) = 0.050$	**Q4 — CounterFact (Hard Regime)** **Edit:** The mother tongue of Danielle Darrieux is **Para:** Shayna does this and Yossel goes still and dies. Danielle Darrieux, a native **Unrel:** The mother tongue of Léon Blum is $d_s(x_i, \tilde{x}_i) = 0.600 \quad d_s(x_i, \hat{x}_i) = 0.143$

pattern. The edit and paraphrase queries both ask about the manufacturer of Colt King Cobra and retain the main entity tokens, giving a relatively low edit–paraphrase distance. The unrelated query asks about a different topic (Indian TV Drama: Ek Veer Ki Ardaas Veera) and shares little surface form with the edit query, giving a relatively high edit–unrelated distance.

ZsRE-hard. ZsRE-hard keeps the edit and unrelated queries from ZsRE but replaces the paraphrase with a surface-distant paraphrase prepended by an unrelated distractor sentence, as shown in Appendix I.1. This increases $d_s(x_i, \tilde{x}_i)$ while leaving $d_s(x_i, \hat{x}_i)$ unchanged, moving the overall distribution toward Q1, a hard paraphrase-unrelated query regime. In Table 3, the ZsRE-hard instance keeps the same edit query and unrelated query as the ZsRE example, but the paraphrase is prepended with an unrelated distractor sentence (International Trade topic) and is rewritten with a different surface form. This raises the edit–paraphrase distance while preserving the large edit–unrelated distance, placing the instance in Q1.

WikiBigEdit. WikiBigEdit contains paraphrases that are often close to the edit query, but its unrelated queries are selected to be surface-similar to the edit query, typically by preserving the relation template and changing only the subject entity. It therefore has relatively low $d_s(x_i, \tilde{x}_i)$ and low $d_s(x_i, \hat{x}_i)$, with most instances in the hard regime Q3. The WikiBigEdit instance in Table 3 follows this structure. The paraphrase keeps the same entity and relation while only slightly changing

the wording of the edit query, leading to a low edit–paraphrase distance. The unrelated query preserves most of the edit query's template and only changes the subject entity (Bernadette Cozart $\rightarrow$ Bernadette Robert) to a confusable one, producing strong surface overlap and therefore a low edit–unrelated distance.

CounterFact. CounterFact combines surface-distant paraphrases with same-domain unrelated queries. Its paraphrases contain unrelated distractor sentences, while unrelated queries are kept in the same domain as the edit query. As a result, many unrelated queries have small $d_s(x_i, \hat{x}_i)$, although some can still have a larger surface distance when the same-domain query changes both the sentence template and the entity. Its overall distribution therefore spans Q1 and Q4, with a substantial mass in Q4, where both axes make routing difficult. The CounterFact instance in Table 3 shows both sources of difficulty. The paraphrase prepends an unrelated distractor sentence (from Jewish folktales), which makes it surface-distant from the edit query. The unrelated query keeps the same relation template, "The mother tongue of ... is," and changes only the subject entity, making it surface-near to the edit query. This combination places the instance in Q4.

Together, these benchmarks cover the easy regime and all hard paraphrase-unrelated query regimes, ensuring that both SEAL and the baselines are evaluated across the full range of surface-distance d_s considered in this work.

D.2 Metric Details

We evaluate all methods with *vanilla generation*, which provides a faithful measure of edited models' real-world behavior (Yang et al., 2025). The model is given only the query and must generate the answer autoregressively. Thus, x_i, $\tilde{x}_i$, and $\hat{x}_i$ are evaluated independently, and neither the target answer y_i^* nor any generated answer from another query is provided as context.

Let $\mathsf{Ans}(f, x)$ denote the answer string generated by model f from query x under this vanilla-generation protocol. We evaluate generated answers using a token-level exact-match score (Wang et al., 2024b). Let $\mathrm{tok}(a)$ denote the tokenizer output for the answer string a. For a generated answer a and a reference answer b, let $\mathrm{tok}(b) = (b_1, \ldots, b_L)$ and let $\mathrm{tok}_L(a) = (a_1, \ldots, a_L)$ denote the generated token sequence truncated or padded to the reference length L. We define

$$\mathrm{EM}_{\mathrm{tok}}(a, b) = \frac{1}{L} \sum_{j=1}^{L} \mathbb{I}[a_j = b_j] \,. \tag{12}$$

Thus, $\mathrm{EM}_{\mathrm{tok}}$ is a token-level score in $[0, 1]$. After T edits, the edited model f_{θ_T} is evaluated on aligned triplets $(x_i, \tilde{x}_i, \hat{x}_i)$ with target answer y_i^*, as defined in Section 2. The three editing metrics are

$$\begin{aligned} \mathrm{Rel} &= \frac{1}{T} \sum_{i=1}^{T} \mathrm{EM}_{\mathrm{tok}}(\mathsf{Ans}(f_{\theta_T}, x_i),\ y_i^*) \,, \\ \mathrm{Gen} &= \frac{1}{T} \sum_{i=1}^{T} \mathrm{EM}_{\mathrm{tok}}(\mathsf{Ans}(f_{\theta_T}, \tilde{x}_i),\ y_i^*) \,, \\ \mathrm{Loc} &= \frac{1}{T} \sum_{i=1}^{T} \mathrm{EM}_{\mathrm{tok}}(\mathsf{Ans}(f_{\theta_T}, \hat{x}_i),\ \mathsf{Ans}(f_{\theta_0}, \hat{x}_i)) \,. \end{aligned} \tag{13}$$

Reliability and Generality therefore require the edited model to generate the updated answer y_i^* from the original edit query and its paraphrase, respectively. Locality requires the edited model's generated answer on the unrelated query $\hat{x}_i$ to remain unchanged from the frozen base model f_{θ_0}.

Overall Performance (OP) is the harmonic mean of the three metrics:

$$\mathrm{OP} = \frac{3\,\mathrm{Rel}\,\mathrm{Gen}\,\mathrm{Loc}}{\mathrm{Rel}\,\mathrm{Gen} + \mathrm{Rel}\,\mathrm{Loc} + \mathrm{Gen}\,\mathrm{Loc}}. \tag{14}$$

D.3 SEAL Hyper-parameter Details

This section provides the SEAL hyper-parameters omitted from the main text. Unless otherwise specified, the following settings are shared across all backbone models and benchmarks. The Sentinel Prompt length is fixed to $m = 4$ soft tokens, and the Sentinel Prompt training set contains

$N = 1\mathbf{k}$ editing instances. For each training triplet, we use $R = 10$ inner optimization steps in Algorithm 1, and train $\mathbf{p}_s$ with a learning rate of 0.001. Routing representations are extracted following Eq. 3 with $\rho = 60$, i.e., Pool_ρ mean-pools the last 60% token positions before the edit layer. Following HoReN, the edit-layer index is 29 for Llama-3.1-8B, Mistral-7B, and DS-Llama-8B, and 56 for DS-Qwen-32B. Indices start at 0; each backbone uses the same edit layer across datasets and editing instances.

For each setting, the routing threshold τ is tuned once on the Sentinel Prompt training set, which is disjoint from evaluation, and then fixed for all evaluation edits. Table 4 reports the Llama-3.1-8B thresholds and Sentinel Prompt training epochs E.

All experiments are conducted on a single NVIDIA A800 80GB GPU.

Table 4: Llama-3.1-8B SEAL hyper-parameters across benchmarks. The shared settings are $m = 4$, $N = 1\mathbf{k}$, $R = 10$, learning rate 0.001, and $\rho = 60$.

Benchmark	Routing threshold τ	Epochs E
WikiBigEdit	0.905	3
CounterFact	0.795	3
ZsRE-hard	0.835	2
ZsRE	0.86	2

D.4 Compute Resources

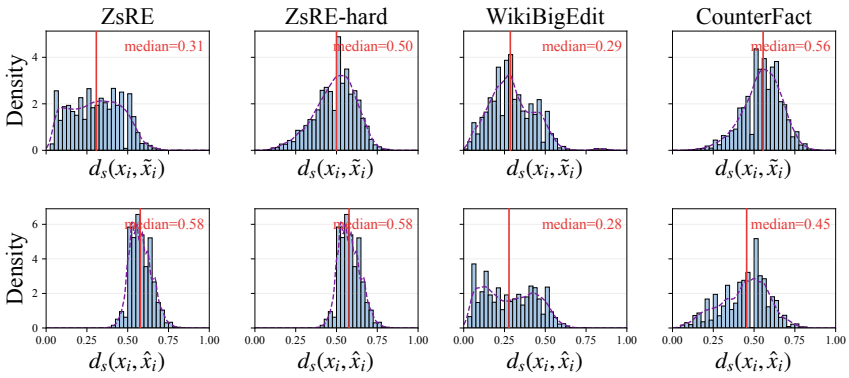
All reported experiments are run on compute workers equipped with a single NVIDIA A800 80GB GPU. Unless otherwise stated, each backbone–benchmark run uses one GPU. Sentinel Prompt training, including $\mathcal{M}_{\mathrm{train}}$ construction over $N = 1\mathbf{k}$ training instances and three-epoch Sentinel Prompt optimization, takes approximately 2.54 GPU-hours per backbone–benchmark pair. This cost is paid once for each backbone–benchmark distribution: the trained Sentinel Prompt can be reused for new edit memories and evaluation edits from the same distribution, without per-edit retraining. Evaluation, including edit-memory construction and test-time inference, at editing scale $T = 1\mathbf{k}$ takes approximately 0.74 GPU-hours per backbone–benchmark pair, while the $T = 10\mathbf{k}$ scalability experiments take approximately 7.89 GPU-hours per benchmark. The total compute used for the main reported SEAL runs is approximately 84.0 GPU-hours, excluding preliminary pilot runs. The compute worker uses 14 vCPU Intel Xeon Gold 6348 CPUs at 2.60GHz, 120GB system memory.

E Additional Analysis of Hard Paraphrase-Unrelated Query Regimes

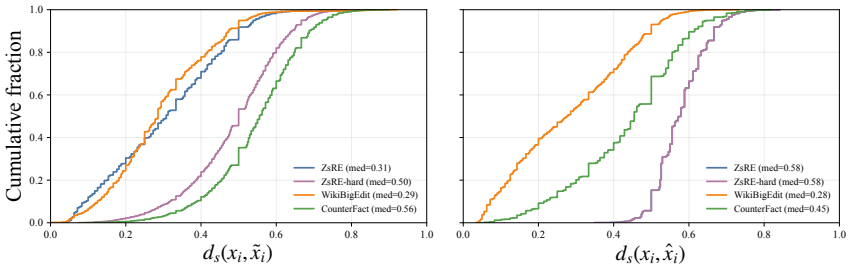
E.1 Surface-Distance Distributions

Figure 7 quantifies the same benchmark properties without relying on the quadrant split. Specifically, Figure 7a plots the empirical density of the two surface-distance variables, $d_s(x_i, \tilde{x}_i)$ (top row) and $d_s(x_i, \hat{x}_i)$ (bottom row), separately for each benchmark, with vertical red lines marking the corresponding medians. The edit–paraphrase distance is relatively low for WikiBigEdit and ZsRE (median 0.29 and 0.31), and substantially higher for ZsRE-hard and CounterFact (median 0.50 and 0.56). Conversely, the edit–unrelated distance is higher for ZsRE and ZsRE-hard (both median 0.58), lower for WikiBigEdit (median 0.28), and intermediate for CounterFact (median 0.45).

The ECDFs in Figure 7b show the same trends from a cumulative perspective: for a distance value t, each curve reports the fraction of instances whose surface distance is at most t. In the edit–paraphrase panel, the WikiBigEdit and ZsRE curves rise earlier, reflecting relatively low $d_s(x_i, \tilde{x}_i)$, whereas the ZsRE-hard and CounterFact curves shift toward larger distances, reflecting surface-distant paraphrases. In the edit–unrelated panel, WikiBigEdit rises earliest, ZsRE and ZsRE-hard rise much later, and CounterFact lies between them, consistent with its same-domain unrelated queries. These one-dimensional distributions match the quadrant layout in Appendix D.1: ZsRE corresponds to Q2, ZsRE-hard to Q1, WikiBigEdit to Q3, and CounterFact mainly spans Q1 and Q4.



(a) Histograms of $d_s(x_i, \tilde{x}_i)$ and $d_s(x_i, \hat{x}_i)$ across benchmarks.



(b) ECDFs of the same distances.

Figure 7: Surface-distance distributions for the four benchmarks. The histograms show the density of d_s for edit–paraphrase and edit–unrelated pairs; the ECDFs summarize the same trends without quadrant binning. For each benchmark, we use the first 10**k** instances.

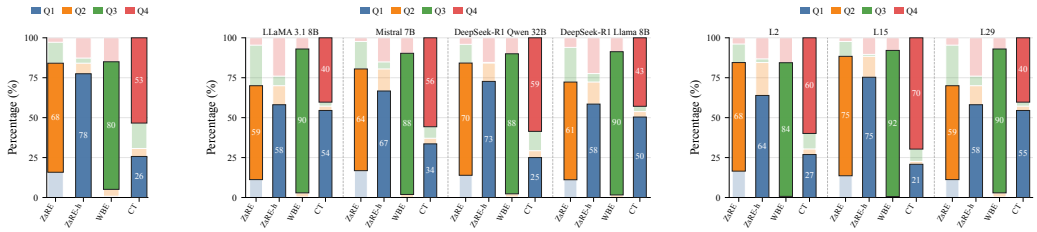
E.2 Representation Distance Distributions Across Backbones and Layers

The main text shows that surface-distance trends persist in the routing hidden-state representation space. Here, we further analyze the relationship between d_s and d_r across different backbone models and representation layers, and find that the same trend is preserved across backbones and layers. Unless otherwise stated, all analyses and figures in this subsection use 1**k** instances per benchmark from the evaluation splits defined in Section 4.1; pooled analyses therefore contain 4**k** evaluation instances across the four benchmarks.

d_s–d_r Quadrant alignment. We first summarize this relationship by comparing quadrant proportions in surface-distance space and hidden-state representation-distance space. As in the main text, quadrant boundaries are set by pooled medians: for surface distance, we use the pooled medians of $d_s(x_i, \tilde{x}_i)$ and $d_s(x_i, \hat{x}_i)$ over all evaluated instances; for each representation setting, we use the corresponding pooled medians of $d_r(x_i, \tilde{x}_i)$ and $d_r(x_i, \hat{x}_i)$. Figure 8 reports the surface-distance quadrant distribution under d_s together with the representation-distance quadrant distributions under d_r when varying either the backbone model or the representation layer. The representation-space proportions remain broadly aligned with the surface-distance proportions: ZsRE is concentrated in Q2, ZsRE-hard in Q1, WikiBigEdit in Q3, and CounterFact mainly in Q1/Q4. This confirms that d_r in the hidden-state representation space follows the same trend as d_s for both edit–paraphrase and edit–unrelated pairs, and that this trend is preserved across different backbones and representation layers: larger $d_s(x_i, \tilde{x}_i)$ leads to larger $d_r(x_i, \tilde{x}_i)$ for edit–paraphrase pairs, and larger $d_s(x_i, \hat{x}_i)$ leads to larger $d_r(x_i, \hat{x}_i)$ for edit–unrelated pairs.

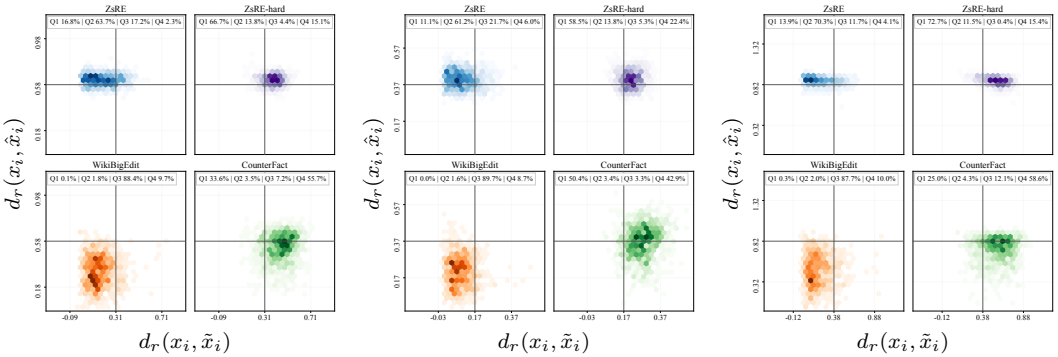
Detailed d_r distributions across backbones. Figure 9 provides detailed distribution hexbin plots for the backbone-level comparison. Across Mistral-7B-Instruct-v0.3, DeepSeek-R1-Distill-Llama-8B, and DeepSeek-R1-Distill-Qwen-32B, ZsRE remains concentrated in Q2, ZsRE-hard in Q1, WikiBigEdit in Q3, and CounterFact across Q1/Q4. Thus, the similar trend between the d_s and the d_r is preserved across backbone models.

Detailed d_r distributions across layers. Figure 10 further varies the representation layer on Llama-3.1-8B-Instruct. Across layers, the easy-regime ZsRE distribution stays mostly in Q2. Among the



(a) Surface (d_s). (b) Backbones (d_r). (c) Llama layers (d_r).

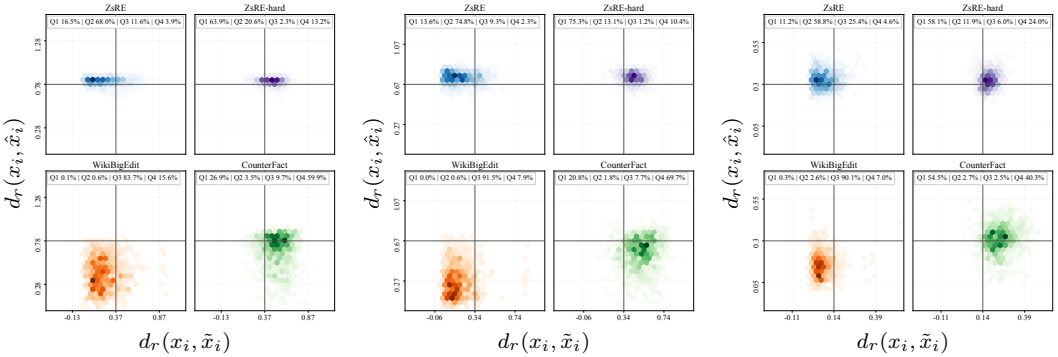
Figure 8: Quadrant proportions. (a) shows d_s quadrants split by the pooled medians of $d_s(x_i, \tilde{x}_i)$ and $d_s(x_i, \hat{x}_i)$. (b) shows d_r quadrants across backbones, and (c) shows d_r quadrants across Llama-3.1-8B-Instruct layers; both are split by the corresponding pooled medians of $d_r(x_i, \tilde{x}_i)$ and $d_r(x_i, \hat{x}_i)$.



(a) Mistral-7B-Instruct-v0.3. (b) DeepSeek-R1-Distill-Llama-8B. (c) DeepSeek-R1-Distill-Qwen-32B.

Figure 9: Hidden-state representation-distance quadrant distributions on three additional backbones. Axes are centered by the pooled medians of $d_r(x_i, \tilde{x}_i)$ and $d_r(x_i, \hat{x}_i)$ for each backbone, as in the main text.

hard-regime benchmarks, ZsRE-hard remains concentrated in Q1; WikiBigEdit remains concentrated in Q3; and CounterFact continues to span Q1/Q4. Thus, d_r follows the same trend as d_s from shallow to late layers.

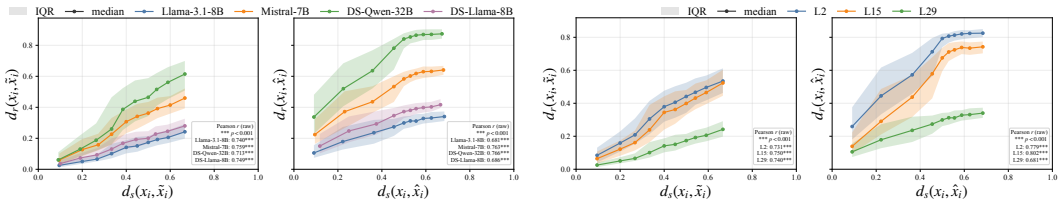


(a) Layer 2. (b) Layer 15. (c) Layer 29.

Figure 10: Hidden-state representation-distance quadrant distributions at different Llama-3.1-8B-Instruct layers. The same benchmark-level trend appears from shallow to late layers. Axes are centered by the pooled medians of $d_r(x_i, \tilde{x}_i)$ and $d_r(x_i, \hat{x}_i)$ for each layer, as in the main text.

d_s–d_r trends without quadrant discretization. The quadrant view above discretizes d_s and d_r by median boundaries. To verify that the same conclusion does not rely on this discretization, Figure 11

removes the quadrant split and directly examines how d_r changes as a function of d_s. We pool the $4\mathbf{k}$ instances from the four benchmarks and bin them by surface distance. Panel (a) compares the resulting trends across backbone models, while Panel (b) varies the representation layer on Llama-3.1-8B-Instruct. In both settings, the hidden-state representation distance increases with surface distance: surface-distant paraphrases tend to be farther from their edit queries in representation space, while surface-near unrelated queries tend to remain closer. This view supports the same claim as the quadrant analysis while avoiding dependence on the particular median-based quadrant boundaries, backbone model, or representation layer.



(a) Across backbone models. (b) Across Llama-3.1-8B-Instruct layers.

Figure 11: Binned relationship between surface distance d_s and hidden-state representation distance d_r. The figure pools $4\mathbf{k}$ instances from the four benchmarks and shows that the d_r trend follows the corresponding d_s trend across backbone models and representation layers without relying on quadrant discretization.

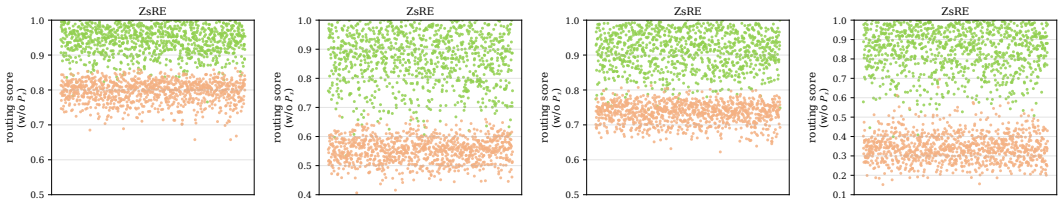
E.3 Routing-Score Overlap Across Backbones and Editors

Across backbone models. The main text analyzes Hopfield routing scores on Llama-3.1-8B-Instruct. Figure 12 extends this analysis by comparing the Hopfield routing-score distributions without $\mathbf{p}_s$ across four different backbone models. The top row shows ZsRE, which represents the easy regime. Across all backbones, the routing scores of paraphrase and unrelated queries are clearly separated, making routing decisions relatively straightforward. In contrast, the bottom row shows CounterFact, which represents a hard-regime benchmark. On this benchmark, the two query types exhibit substantial routing-score overlap across all backbones, indicating higher routing difficulty and a greater risk of routing failure. This observation is consistent with the analysis in the main text: hard paraphrase-unrelated query regimes are associated with routing-score overlap, and changing the backbone alone does not remove this overlap.

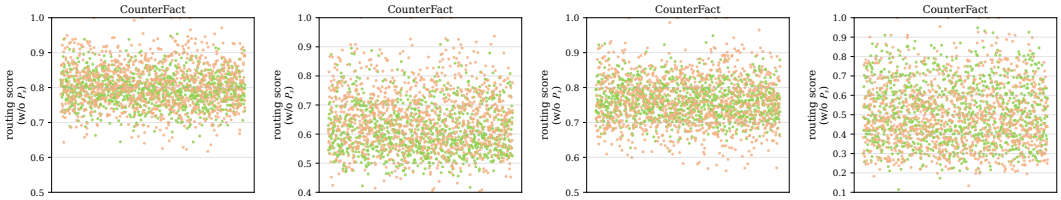
Across memory-based editors. Figure 13 repeats the same comparison for three additional memory-based editors. Here, "routing score" denotes the method-specific scalar used to make the routing decision:

- **GRACE** (Hartvigsen et al., 2023): the router computes the nearest-neighbor L2 distance between the query representation and stored hidden-state keys; this nearest-key distance is the routing score. If the nearest key lies within the learned ϵ radius, the corresponding edit value is activated; otherwise the query falls back to the base-model path.
- **MELO** (Yu et al., 2024): the router first computes the nearest-neighbor L2 distance between the query representation and cluster centers in its vector database; this nearest-cluster-center distance is the routing score. If this distance is below the routing threshold, MELO searches the selected cluster for the nearest key and loads the corresponding LoRA block; otherwise it uses the zero LoRA block to preserve the base model.
- **WISE** (Wang et al., 2024a): the router computes the L2 distance between activation representations induced by the side memory and the main memory; this activation-representation distance is the routing score. The side memory is activated only when this distance exceeds a threshold; otherwise the query remains on the main-memory (base model) path.

Because these definitions have different scales and directions, their absolute routing score values are not comparable. The within-method pattern is nevertheless consistent: ZsRE (representing the easy regime) produces clearly separated paraphrase and unrelated routing scores, whereas the CounterFact distributions (representing a hard-regime benchmark) collapse into overlapping ranges. Thus,



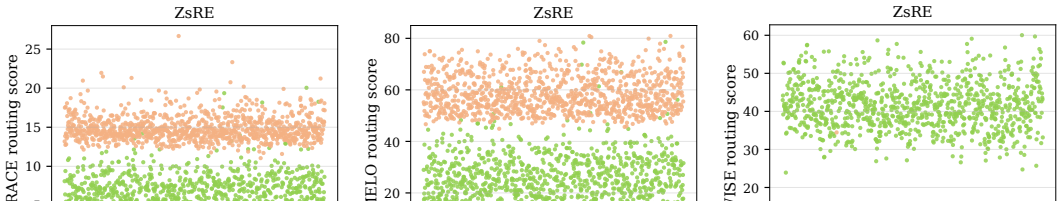
(a) Llama-3.1-8B-Instruct. (b) Mistral-7B-Instruct-v0.3. (c) DeepSeek-R1-Distill-Llama-8B. (d) DeepSeek-R1-Distill-Qwen-32B.



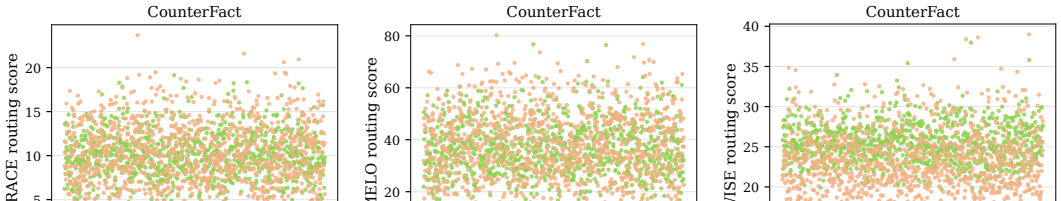
(e) Llama-3.1-8B-Instruct. (f) Mistral-7B-Instruct-v0.3. (g) DeepSeek-R1-Distill-Llama-8B. (h) DeepSeek-R1-Distill-Qwen-32B.

Figure 12: Max routing-score distributions under the Hopfield router without $\mathbf{p}_s$ across backbones. The top row shows ZsRE and the bottom row shows CounterFact. Green and orange points denote paraphrase and unrelated queries, following the main text.

the routing-score overlap observed on the hard-regime benchmark is not limited to the Hopfield router, but also appears for these additional memory-based routing mechanisms, making routing decisions more difficult and increasing the risk of routing failure in hard regimes across methods.



(a) GRACE. (b) MELO. (c) WISE.



(d) GRACE. (e) MELO. (f) WISE.

Figure 13: Selected routing-score distributions for three additional memory-based editors. For GRACE and MELO, the selected routing score is the minimum distance used by the router; for WISE, it is the maximum activation difference between side memory and main memory used by the router. The top row shows ZsRE and the bottom row shows CounterFact. Green and orange points denote paraphrase and unrelated queries.

F Additional Results and Ablation Studies

F.1 Additional Main Results Across Backbones and Editing Scales

Table 5 reports the deferred DeepSeek-R1-Distill-Llama-8B results at $T = 1\mathbf{k}$.

Table 5: Deferred main results on DeepSeek-R1-Distill-Llama-8B at editing scale $T = 1\mathbf{k}$. Bold marks the best OP per benchmark; blue shading identifies SEAL.

Method	WikiBigEdit				CounterFact				ZsRE-hard				ZsRE			
	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP
FT-L	0.070	0.057	0.012	0.026	0.035	0.003	0.000	0.000	0.127	0.085	0.019	0.042	0.127	0.111	0.019	0.043
MEMIT-M	0.374	0.335	0.130	0.225	0.517	0.415	0.491	0.470	0.574	0.470	0.178	0.316	0.574	0.514	0.178	0.322
AlphaEdit	0.855	0.651	0.146	0.314	0.986	0.627	0.574	0.689	0.949	0.646	0.343	0.544	0.949	0.858	0.343	0.584
DEFER	0.171	0.105	0.974	0.183	0.059	0.048	0.482	0.075	0.071	0.016	0.951	0.039	0.071	0.053	0.951	0.088
WISE	0.161	0.108	0.336	0.163	0.379	0.173	0.311	0.258	0.315	0.166	0.990	0.294	0.315	0.270	0.990	0.380
GRACE	0.997	0.069	0.959	0.181	0.996	0.025	0.991	0.071	0.999	0.008	1.000	0.024	0.999	0.143	1.000	0.334
MELO	0.838	0.115	0.910	0.273	0.932	0.032	0.985	0.090	0.807	0.135	1.000	0.311	0.809	0.268	1.000	0.503
HoReN	0.998	0.789	0.629	0.777	0.999	0.485	0.326	0.489	0.999	0.640	0.901	0.817	0.999	0.952	0.995	0.982
SEAL	0.972	0.825	0.870	**0.885**	0.966	0.746	0.822	**0.835**	0.985	0.897	0.994	**0.957**	0.998	0.972	1.000	**0.990**

The deferred DeepSeek-R1-Distill-Llama-8B results follow the same trend as the main results: most memory-based editors achieve their highest OP on the easy ZsRE regime, while their OP degrades on the hard-regime benchmarks. SEAL again improves over the baselines across these regimes, showing that the same Generality–Locality gains hold on this additional backbone.

F.2 Routing Accuracy on Four Benchmarks

Table 6 reports routing accuracy for the memory-based editors on Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$. Edit and paraphrase routing accuracy measure whether a query is routed to its corresponding edit-specific path, while Locality routing accuracy measures whether an unrelated query is rejected and sent to the base-model path. We report $\mathrm{Avg}_{\mathrm{P,L}} = (\mathrm{Acc}_{\mathrm{Para}} + \mathrm{Acc}_{\mathrm{Loc}})/2$ as the mean of Paraphrase and Locality routing accuracy. GRACE and MELO achieve high Locality routing mainly by conservative rejection, but they route few paraphrases to the edit-specific path, which explains their weak Generality. HoReN works well on the easy ZsRE regime ($\mathrm{Avg}_{\mathrm{P,L}} = 87.3$), but degrades on the hard regimes ($69.6/32.8/67.9$ on WikiBigEdit/CounterFact/ZsRE-hard). MELO and GRACE show the same trend: their $\mathrm{Avg}_{\mathrm{P,L}}$ scores are higher on ZsRE (57.1 and 54.2) than on the three hard-regime benchmarks. SEAL markedly improves routing on these hard-regime benchmarks, raising $\mathrm{Avg}_{\mathrm{P,L}}$ to $82.5/87.8/88.4$, respectively.

These results are consistent with the main editing results and further verify the main claim that hard regimes are more prone to routing failure, while $\mathbf{p}_s$ improves performance by improving the router's paraphrase–unrelated query routing accuracy.

Table 6: Routing accuracy of memory-based editors at $T = 1\mathbf{k}$ on Llama-3.1-8B-Instruct. $\mathrm{Avg}_{\mathrm{P,L}} = (\mathrm{Para.} + \mathrm{Loc.})/2$.

Method	WikiBigEdit				CounterFact				ZsRE-hard				ZsRE			
	Edit	Para.	Loc.	$\mathrm{Avg}_{\mathrm{P,L}}$	Edit	Para.	Loc.	$\mathrm{Avg}_{\mathrm{P,L}}$	Edit	Para.	Loc.	$\mathrm{Avg}_{\mathrm{P,L}}$	Edit	Para.	Loc.	$\mathrm{Avg}_{\mathrm{P,L}}$
GRACE	99.9	10.6	93.5	52.1	99.8	6.6	97.8	52.2	100.0	0.0	100.0	50.0	100.0	8.4	100.0	54.2
MELO	83.8	15.7	85.3	50.5	91.9	9.5	92.6	51.1	82.7	6.3	99.4	52.9	82.7	14.1	100.0	57.1
HoReN	100.0	75.7	63.4	69.6	100.0	46.6	18.9	32.8	100.0	43.8	92.0	67.9	100.0	74.6	99.9	87.3
SEAL	97.9	80.7	84.2	82.5	97.1	83.6	91.9	87.8	98.9	76.8	100.0	88.4	100.0	76.7	100.0	88.4

F.3 Score-Level Evidence for the Asymmetric Pulling Effect

Table 7 provides the score-level details supporting the Asymmetric Pulling Effect analyzed in Section 4.4. For each benchmark, we report the mean max routing scores of paraphrase and unrelated queries without and with the Sentinel Prompt $\mathbf{p}_s$ on Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$.

Across benchmarks, $\mathbf{p}_s$ maintains or increases paraphrase scores while suppressing unrelated scores. On ZsRE-hard, the mean paraphrase score increases by $+0.075$ and the mean unrelated score decreases by -0.050; on CounterFact, the corresponding shifts are $+0.073$ and -0.110. On Wik-

Table 7: Mean max routing scores of paraphrase queries and unrelated queries under the Hopfield router without (S_{wo}) and with (S_{w}) the Sentinel Prompt $\mathbf{p}_s$, and their differences (Δ). All values are computed on Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$.

Benchmark	S_{wo}		S_{w}		Δ	
	Para.	Unrel.	Para.	Unrel.	Para.	Unrel.
ZsRE	0.939	0.795	0.941	0.731	+0.002	−0.064
ZsRE-hard	0.845	0.795	0.920	0.745	+0.075	−0.050
WikiBigEdit	0.935	0.887	0.932	0.859	−0.003	−0.028
CounterFact	0.794	0.807	0.867	0.697	+0.073	−0.110

iBigEdit, the paraphrase score is nearly unchanged (-0.003), while the unrelated score is still suppressed (-0.028). This asymmetric score movement directly reduces the routing-score overlap that makes hard regimes difficult, thereby supporting the interpretation that $\mathbf{p}_s$ improves routing by making paraphrase queries easier to activate and unrelated queries easier to reject.

F.4 Ablation on Decoupled Two-Stage Inference

We ablate how the prompted state from Stage 1 should be used during Stage 2 generation after the routing decision is made. We compare two generation policies. In both policies, the routing decision itself is fixed: queries routed to the edit-specific path inject the selected edit value, while queries routed to the base-model path do not inject any edit value. The ablation only changes whether Stage 2 generation continues from the prompted state cached in Stage 1 or discards that state and reruns without $\mathbf{p}_s$ from the original input x. In the first policy, all queries continue inference from the prompted $z^{l-1}(x^+)$ cached in Stage 1; edit-path queries inject the selected value into this prompted state, whereas base-model-path queries keep the prompted state but receive no edit value. In the second policy, all queries discard the prompted state, remove $\mathbf{p}_s$, and restart from the original input x for clean $\mathbf{p}_s$-free inference; edit-path queries inject the selected edit value during this prompt-free rerun, while base-model-path queries run the frozen model without injection. We evaluate both policies on WikiBigEdit using Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$.

Table 8: Ablation on Stage 2 generation policies in Decoupled Two-Stage Inference. Results on WikiBigEdit with Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$.

Stage 2 generation policy	Rel.	Gen.	Loc.
Continue from prompted state for all queries	0.965	0.805	0.552
Discard prompted state and rerun for all queries	0.965	0.806	0.852

Table 8 shows that continuing from the prompted state for all queries keeps Reliability and Generality nearly unchanged compared with discarding the prompted state and rerunning for all queries (0.965/0.805 vs. 0.965/0.806). This suggests that, once the selected edit value is injected, $\mathbf{p}_s$ does not substantially perturb the final answer on the edit-specific path. However, continuing from the prompted state substantially hurts Locality (0.552 vs. 0.852), indicating that $\mathbf{p}_s$ can significantly change the frozen base model's output on the base-model path. Therefore, SEAL adopts the branch-specific Decoupled Two-Stage Inference in Section 3.4: if $s^* > \tau$, SEAL injects v_{i^*} into the prompted $z^{l-1}(x^+)$ and directly continues inference; if $s^* \leq \tau$, SEAL discards the prompted state, removes $\mathbf{p}_s$, and restarts from the original input x for clean $\mathbf{p}_s$-free inference.

F.5 Ablation on Router

To assess whether the Sentinel Prompt's routing benefit depends on the Hopfield router used in the main experiments, we evaluate it with a GRACE-style router and a BGE-based router. For each router, we compare performance with and without a $\mathbf{p}_s$ trained specifically for that router. Together, the two experiments show that SEAL's framework is not tied to the Hopfield router and remains effective with other routers.

F.5.1 Adaptability of SEAL's Framework to a GRACE-style Router

GRACE Router denotes a GRACE-style router (Hartvigsen et al., 2023) with a single global routing threshold τ, rather than the original GRACE mechanism with dynamic ϵ. The GRACE Router+$\mathbf{p}_s$

variant uses the same Sentinel Prompt training objective, training procedure, and decoupled two-stage inference design as SEAL. Its $\mathbf{p}_s$ is trained specifically for the GRACE Router's score definition. All results are evaluated on WikiBigEdit using Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$.

Table 9: Performance of the GRACE-style router with and without the Sentinel Prompt on WikiBigEdit using Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$. Both variants use a single global routing threshold.

Method	Rel.	Gen.	Loc.	OP
GRACE Router	0.990	0.482	0.572	0.621
GRACE Router+$\mathbf{p}_s$	0.960	0.515	0.627	0.655

Table 9 shows that $\mathbf{p}_s$ remains effective with the GRACE Router: adding $\mathbf{p}_s$ improves Generality from 0.482 to 0.515, Locality from 0.572 to 0.627, and OP from 0.621 to 0.655. The gains on both query types show that $\mathbf{p}_s$ also improves routing with this router.

F.5.2 Adaptability of SEAL's Framework to a BGE-based Router

We also evaluate an editor with a BGE-based router. We use frozen BGE-large-en-v1.5 (Xiao et al., 2023), an embedding model trained for dense semantic retrieval. We retain HoReN's edit values, intervention mechanism, backbone, and evaluation protocol, and compute routing scores as cosine similarity between BGE embeddings obtained directly from the natural-language edit and test queries, with a fixed routing threshold. We denote this variant **BGE Router**.

For BGE Router+$\mathbf{p}_s$, $\mathbf{p}_s$ is trained at the BGE input using the same Sentinel Prompt training objective and training procedure as SEAL. Edit-query keys are encoded without $\mathbf{p}_s$, whereas test queries are encoded with it. BGE is used only for routing, so the prompt never enters the LLM and cannot perturb frozen-base-model generation; Decoupled Two-Stage Inference is unnecessary. All results use Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$.

Table 10: Performance of the BGE-based router with and without the Sentinel Prompt on CounterFact using Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$.

Method	Rel.	Gen.	Loc.	OP
BGE Router	0.999	0.678	0.689	0.763
BGE Router+$\mathbf{p}_s$	0.999	0.949	0.941	0.962

Table 10 shows that $\mathbf{p}_s$ improves Generality from 0.678 to 0.949 and Locality from 0.689 to 0.941, raising OP from 0.763 to 0.962 while preserving Reliability at 0.999. These results show that $\mathbf{p}_s$ remains effective with this router.

F.6 Inference Efficiency and Parameter Overhead

We evaluate inference efficiency on WikiBigEdit with $T = 1\mathbf{k}$ using the same Llama-3.1-8B-Instruct setting as the main experiments. Table 11 reports the average wall-clock inference time per query and the number of additional trainable parameters beyond the frozen backbone.

Table 11: Inference latency and additional trainable parameters on WikiBigEdit with Llama-3.1-8B-Instruct at $T = 1\mathbf{k}$. Times are average milliseconds per query.

Method	Edit Time	Para. Time	Unrel. Time	Mean Time	Extra Trainable Params
Base model	74.509	74.542	68.367	72.473	0
HoReN	81.915	81.766	75.135	79.606	18,432,000
SEAL	82.315	85.241	92.684	86.747	18,448,384

The latency pattern matches the design of Decoupled Two-Stage Inference. Compared with HoReN, SEAL changes edit-query latency only slightly ($81.915 \rightarrow 82.315$ ms, $+0.5\%$) and has a modest increase on paraphrase queries ($81.766 \rightarrow 85.241$ ms, $+4.2\%$). The largest increase occurs for unrelated queries ($75.135 \rightarrow 92.684$ ms, $+23.4\%$), which is expected: all queries first use the Sentinel Prompt for routing, but only queries routed to the base-model path discard the prompted hidden state and rerun prompt-free inference on the original input. Edit and paraphrase queries usually activate

the edit-specific path and can continue from the prompted hidden state after value injection, avoiding this extra recomputation. As a result, the mean latency increase over HoReN remains moderate ($79.606 \rightarrow 86.747$ ms, $+9.0\%$).

SEAL also introduces negligible additional trainable parameters relative to HoReN. In this setting, HoReN has $18{,}432{,}000$ additional trainable parameters, while SEAL has $18{,}448{,}384$. The difference is

$$18{,}448{,}384 - 18{,}432{,}000 = 16{,}384 = 4 \times 4096,$$

corresponding exactly to the four soft tokens of the Sentinel Prompt for a hidden size of 4096. This is only 0.089% of HoReN's additional trainable parameter count, showing that SEAL obtains large performance improvements in hard regimes with only a very small number of extra parameters.

F.7 Threshold Sensitivity and Edit-only Calibration

We evaluate the sensitivity of the routing threshold τ for HoReN and SEAL. HoReN corresponds to the Hopfield router without the Sentinel Prompt $\mathbf{p}_s$, while SEAL prepends $\mathbf{p}_s$ during routing. To focus on the Generality–Locality trade-off, we report Generality, Locality, and $\mathrm{OP}_{\mathrm{GL}}$, the harmonic mean of Generality and Locality.

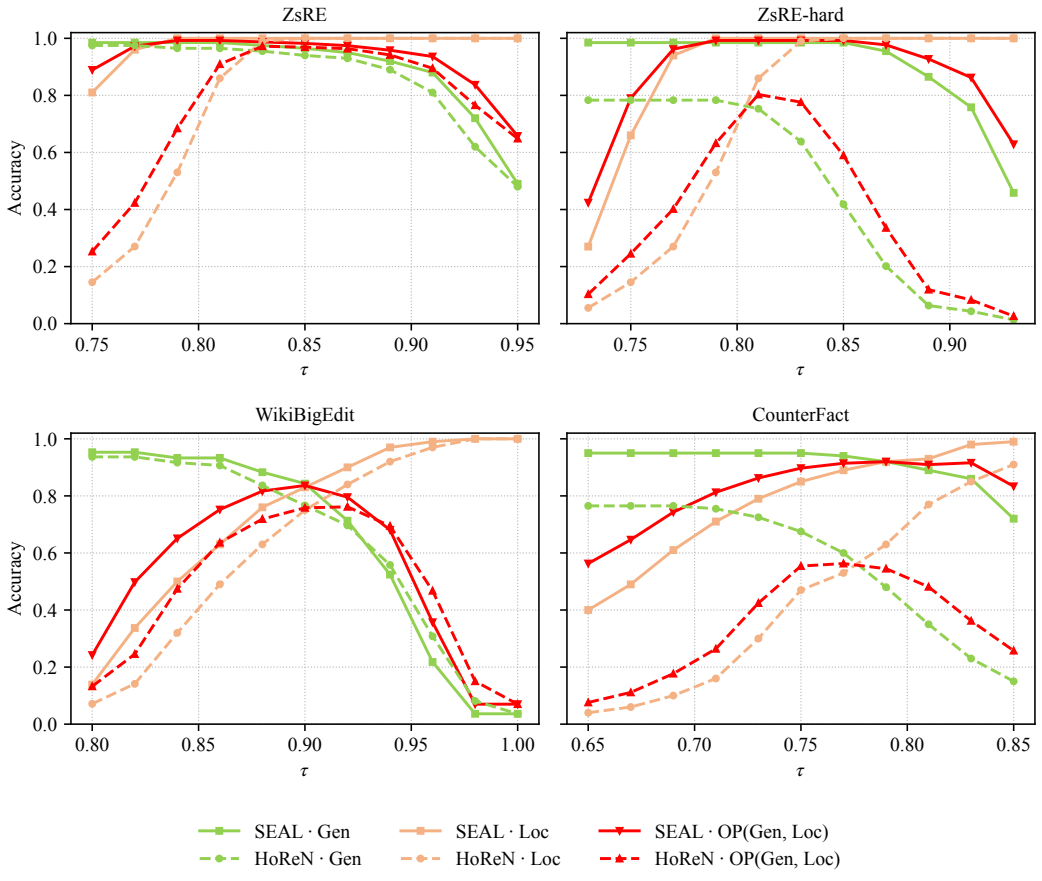


Figure 14: Threshold sensitivity of HoReN and SEAL on the four benchmarks. For each benchmark, we edit and evaluate 100 instances, sweeping the routing threshold τ with step size 0.02 over the following ranges: $[0.75, 0.95]$ for ZsRE, $[0.73, 0.93]$ for ZsRE-hard, $[0.80, 1.00]$ for WikiBigEdit, and $[0.65, 0.85]$ for CounterFact. $\mathrm{OP}_{\mathrm{GL}}$ denotes the harmonic mean of Generality and Locality.

Figure 14 shows that Hopfield routers are highly sensitive to τ. When τ is too large, many paraphrase queries have routing scores below the threshold and fail to activate their corresponding edit-specific paths, reducing Generality. When τ is too small, many unrelated queries exceed the threshold, incor-

rectly activate edit-specific paths, and change answers that should be preserved, reducing Locality. Therefore, varying τ induces a clear Generality–Locality trade-off.

This trade-off is mild in the easy regime but severe in hard regimes. On ZsRE, where paraphrase and unrelated routing scores are already well separated, HoReN can obtain high Generality and Locality by choosing an appropriate threshold, reaching $\mathrm{OP}_{\mathrm{GL}} = 0.972$. In contrast, on the hard-regime benchmarks, tuning τ cannot fully resolve the routing-score overlap between the two query types: the best $\mathrm{OP}_{\mathrm{GL}}$ of HoReN drops to 0.803 on ZsRE-hard, 0.762 on WikiBigEdit, and 0.563 on CounterFact.

With $\mathbf{p}_s$, SEAL consistently achieves a better threshold trade-off. The best $\mathrm{OP}_{\mathrm{GL}}$ increases to 0.992 on ZsRE-hard, 0.837 on WikiBigEdit, and 0.920 on CounterFact. SEAL also maintains a wider threshold range over which both Generality and Locality remain high. These results support the interpretation that the Sentinel Prompt $\mathbf{p}_s$ improves routing by separating the paraphrase and unrelated routing-score distributions, rather than merely shifting the optimal threshold.

Edit-only threshold calibration. We also calibrate SEAL's routing threshold τ using only edit queries x_i from each benchmark's training set. This mirrors real deployment, where the edit-side data are available but the future paraphrase or unrelated queries or their annotations are not. The fixed pipeline in Appendix H.2 generates paraphrase and unrelated queries for these training edits, forming synthetic calibration triplets. We select τ on these triplets, then evaluate SEAL with that threshold on the same held-out split. Table 12 compares this edit-only τ calibration with SEAL's threshold calibrated on full training triplets containing all three query types, and reports HoReN with full-triplet threshold calibration. Edit-only calibration gives similar SEAL OP on WikiBigEdit, CounterFact, and ZsRE, with a larger gap on ZsRE-hard. Even under edit-only τ calibration, SEAL substantially outperforms HoReN on all three hard-regime benchmarks.

Table 12: SEAL threshold calibration and HoReN comparison on Llama-3.1-8B at $T = 1\mathbf{k}$. Full triplets contain training edit, paraphrase, and unrelated queries; edit-only synthesis generates the latter two from training edits. HoReN OP uses a threshold calibrated on full training triplets.

	SEAL: calibrated τ		SEAL: OP		HoReN: OP
Benchmark	Full triplets	Edit-only synthesis	Full triplets	Edit-only synthesis	Full triplets
WikiBigEdit	0.905	0.890	0.869	0.848	0.776
CounterFact	0.795	0.805	0.911	0.905	0.364
ZsRE-hard	0.835	0.905	0.968	0.891	0.751
ZsRE	0.86	0.850	0.991	0.991	0.979

F.8 Effect of Sentinel Prompt Length

We ablate the number of trainable soft tokens in the Sentinel Prompt $\mathbf{p}_s \in \mathbb{R}^{m \times d}$ on WikiBigEdit. All experiments use Llama-3.1-8B-Instruct at editing scale $T = 1\mathbf{k}$, with the same evaluation protocol as the main experiments. The Sentinel Prompt training set contains $N = 1\mathbf{k}$ editing instances, and $\mathbf{p}_s$ is trained for $E = 3$ epochs.

Table 13: Ablation on the number of trainable soft tokens m in the Sentinel Prompt $\mathbf{p}_s$ on WikiBigEdit. $m = 0$ corresponds to HoReN (the Hopfield router without $\mathbf{p}_s$). OP is the harmonic mean of Reliability, Generality, and Locality.

m	Rel.	Gen.	Loc.	OP
0	0.987	0.753	0.655	0.776
1	0.985	0.811	0.743	0.835
2	0.970	0.758	0.866	0.856
4	0.965	0.805	0.852	**0.869**
6	0.976	0.826	0.812	0.865
8	0.941	0.834	0.820	0.862

Table 13 shows that SEAL is not very sensitive to the Sentinel Prompt length. Even with a single soft token ($m = 1$), SEAL substantially improves over HoReN ($m = 0$), increasing OP from 0.776 to 0.835. The shorter Sentinel Prompt settings ($m = 1$ and $m = 2$) still yield lower OP than the stronger settings, suggesting that very few soft tokens have limited capacity to reshape the hidden-state representation used by the router. Once the Sentinel Prompt length reaches $m = 4$,

performance becomes stable: $m = 4$, $m = 6$, and $m = 8$ all achieve relatively high OP. The four-token setting used in the main experiments obtains the best OP and incurs the smallest additional routing-time cost among these high-performing settings.

F.9 Effect of Sentinel Prompt Training Set Size

We vary N, the number of editing instances reserved as the Sentinel Prompt training set on WikiBigEdit. For each setting, we first construct $\mathcal{M}_{\text{train}} = \{(k_i, v_i)\}_{i=1}^{N}$ without $\mathbf{p}_s$, and then optimize $\mathbf{p}_s$ using the corresponding training triplets $(x_i, \tilde{x}_i, \hat{x}_i)$. Following Section 4.1, the remaining WikiBigEdit data starting from index $4\mathbf{k}$ are used for editing and evaluation. All settings use Llama-3.1-8B-Instruct at editing scale $T = 1\mathbf{k}$, and the Sentinel Prompt is trained for three epochs.

Table 14: Ablation on N, the number of editing instances in the WikiBigEdit Sentinel Prompt training set. OP is the harmonic mean of Reliability, Generality, and Locality.

N	Rel.	Gen.	Loc.	OP
100	0.957	0.713	0.814	0.816
500	0.914	0.792	0.853	0.850
$1\mathbf{k}$	0.965	0.805	0.852	**0.869**
$3\mathbf{k}$	0.885	0.815	0.805	0.834

Table 14 shows that increasing N improves OP up to a moderate scale. OP increases from 0.816 with $N = 100$ to 0.850 with $N = 500$, and reaches its best value of 0.869 with $N = 1\mathbf{k}$. Increasing the Sentinel Prompt training set to $N = 3\mathbf{k}$ does not further improve OP. This suggests that $N = 1\mathbf{k}$ already provides sufficient training triplets to improve performance in hard regimes.

G Algorithmic Details of SEAL

Algorithm 1 summarizes the training procedure for the Sentinel Prompt $\mathbf{p}_s$. The procedure follows Section 3.3: a subset of each benchmark is reserved as the Sentinel Prompt training set, and the remaining data are used for editing and evaluation. The backbone f_{θ_0} and the training memory $\mathcal{M}_{\text{train}}$ are kept frozen; only $\mathbf{p}_s$ is optimized.

Algorithm 2 summarizes the decoupled two-stage inference procedure for a test query x. The Sentinel Prompt $\mathbf{p}_s$ is prepended before the query x to obtain the routing decision. If the query is routed to the edit-specific path, SEAL reuses the prompted state and continues generation with the selected edit value injected. If the query is routed to the base-model path, SEAL discards the prompted state and runs the base-model path on the original input, thereby preserving the base-model output distribution.

H Automated Training Data Synthesis

H.1 Motivation and Method

In realistic editing scenarios, the editor often has access only to the edit-side query x_i, while the future paraphrase and unrelated queries that users may issue cannot be anticipated. Therefore, benchmark-provided paraphrase and unrelated annotations may not be available as supervision for training $\mathbf{p}_s$. To address this setting, we construct an LLM-based pipeline that takes only x_i and synthesizes training triplets $(x_i, \tilde{x}_i, \hat{x}_i)$.

The pipeline is designed to generate training data that reflects the two sources of routing difficulty studied in this work. For the paraphrase side, it encourages large $d_s(x_i, \tilde{x}_i)$ while preserving the meaning of the edit query. For the unrelated side, it encourages small $d_s(x_i, \hat{x}_i)$ while changing the queried fact. We denote the variant whose Sentinel Prompt is trained only on these synthetic triplets as SEAL_{SYN}. The implementation details are provided in Appendix H.2, and the experimental results are reported in Appendix H.3.

Algorithm 1 Sentinel Prompt Training in SEAL

Require: Sentinel Prompt training set $\mathcal{D}_{\text{train}} = \{(x_i, \tilde{x}_i, \hat{x}_i, y_i^*)\}_{i=1}^{N}$; frozen backbone f_{θ_0}; Hopfield routing score S_{HPF}; Sentinel Prompt length m; margins $\delta_e, \delta_p, \delta_l$; learning rate η; number of epochs E; inner optimization steps R
Ensure: Trained Sentinel Prompt $\mathbf{p}_s$

$\mathcal{M}_{\text{train}} \leftarrow \emptyset$ ▷ Construct $\mathcal{M}_{\text{train}}$ without $\mathbf{p}_s$
for $i = 1, \ldots, N$ **do**
  $k_i \leftarrow h_r(x_i)$
  Construct the corresponding edit value v_i for target y_i^*
  $\mathcal{M}_{\text{train}} \leftarrow \mathcal{M}_{\text{train}} \cup \{(k_i, v_i)\}$
end for
$\mathcal{K}_{\text{train}} \leftarrow \{k_i\}_{i=1}^{N}$
Initialize $\mathbf{p}_s \in \mathbb{R}^{m \times d}$ ▷ Optimize $\mathbf{p}_s$ based on frozen $\mathcal{M}_{\text{train}}$
for $e = 1, \ldots, E$ **do**
  for each $(x_i, \tilde{x}_i, \hat{x}_i, y_i^*) \in \mathcal{D}_{\text{train}}$ **do**
    for $r = 1, \ldots, R$ **do**
      $s_e \leftarrow S_{\text{HPF}}(\mathbf{p}_s \oplus x_i, k_i)$
      $s_p \leftarrow S_{\text{HPF}}(\mathbf{p}_s \oplus \tilde{x}_i, k_i)$
      $s_l \leftarrow \max_{k \in \mathcal{K}_{\text{train}}} S_{\text{HPF}}(\mathbf{p}_s \oplus \hat{x}_i, k)$
      $\mathcal{L}_{\text{edit}} \leftarrow \max(0, \delta_e - s_e)$
      $\mathcal{L}_{\text{para}} \leftarrow \max(0, \delta_p - s_p)$
      $\mathcal{L}_{\text{unrel}} \leftarrow \max(0, s_l - \delta_l)$
      $\mathcal{L}_{\text{sentinel}} \leftarrow \mathcal{L}_{\text{edit}} + \mathcal{L}_{\text{para}} + \mathcal{L}_{\text{unrel}}$
      Update $\mathbf{p}_s$ using $\nabla_{\mathbf{p}_s} \mathcal{L}_{\text{sentinel}}$ with learning rate η
    end for
  end for
end for
return $\mathbf{p}_s$

Algorithm 2 Decoupled Two-Stage Inference in SEAL

Require: Query x; frozen backbone f_{θ_0}; edit layer l; Edited memory $\mathcal{M}_T = \{(k_i, v_i)\}_{i=1}^{T}$; Sentinel Prompt $\mathbf{p}_s$; threshold τ; maximum output length L
Ensure: Output y

$x^+ \leftarrow \mathbf{p}_s \oplus x$ ▷ Routing
$z^{l-1}(x^+) \leftarrow f_{\theta_0}^{<l}(x^+), \quad h_r(x^+) \leftarrow \text{Pool}_\rho(z^{l-1}(x^+))$
$i^* \leftarrow \arg\max_i S_{\text{HPF}}(x^+, k_i)$
$s^* \leftarrow S_{\text{HPF}}(x^+, k_{i^*})$
if $s^* > \tau$ **then** ▷ Generation
  $\bar{z}_1^l \leftarrow \text{Intervene}(z^{l-1}(x^+), v_{i^*})$
  $y_1 \leftarrow f_{\theta_0}^{>l}(\bar{z}_1^l)$ ▷ Continue inference from the prompted state with the edit value injected
  $q_1 \leftarrow x^+ \oplus y_1$
  for $t = 2, \ldots, L$ **do**
    $z^{l-1}(q_{t-1}) \leftarrow f_{\theta_0}^{<l}(q_{t-1})$
    $\bar{z}_t^l \leftarrow \text{Intervene}(z^{l-1}(q_{t-1}), v_{i^*})$
    $y_t \leftarrow f_{\theta_0}^{>l}(\bar{z}_t^l)$
    $q_t \leftarrow q_{t-1} \oplus y_t$
  end for
else
  $q_0 \leftarrow x$ ▷ Discard the prompted state and rerun from x
  for $t = 1, \ldots, L$ **do**
    $y_t \leftarrow f_{\theta_0}(q_{t-1})$
    $q_t \leftarrow q_{t-1} \oplus y_t$
  end for
end if
return $y_{1:L}$

H.2 Implementation Details

Unless otherwise specified, all data synthesis procedures described in this section are carried out using Qwen3-MAX (Team, 2025).

This subsection details the prompt-based synthesis pipeline used to construct Sentinel Prompt training data when only the edit-side query x_i is available. The goal is not to perform generic data augmentation, but to construct training triplets that explicitly reflect the hard paraphrase-unrelated query regimes defined in Section 2. To make the Sentinel Prompt robust to routing difficulty, the synthesized data are designed to simultaneously exhibit two properties: a paraphrase query with *large* $d_s(x_i, \tilde{x}_i)$ and an unrelated query with *small* $d_s(x_i, \hat{x}_i)$. In other words, the synthesis objective explicitly encourages paraphrases to be as surface-distant from the edit query as possible, while keeping unrelated queries as surface-near to the edit query as possible. In Appendix H.3, we show that the resulting synthetic supervision is effective across both hard-regime and easy-regime benchmarks.

Input. The synthesis pipeline takes only the edit-side query x_i as input. It does not assume access to benchmark-provided paraphrase queries or unrelated queries at synthesis time. The task of the generator is to produce one set of surface-distant paraphrases and one set of surface-near but semantically different unrelated queries from the edit query alone.

How to reduce $d_s(x_i, \hat{x}_i)$. To synthesize unrelated training data with small $d_s(x_i, \hat{x}_i)$, we instruct the LLM to keep the relation and sentence template of x_i as unchanged as possible and replace only the subject with a confusable entity. The LLM is asked to generate a replacement entity that is close in name, spelling, or semantic domain to the original subject, so that the resulting query remains lexically very similar to x_i while clearly asking about a different fact. This instruction encourages a small surface distance while preserving semantic divergence.

How to increase $d_s(x_i, \tilde{x}_i)$. To synthesize paraphrase training data with large $d_s(x_i, \tilde{x}_i)$, we instruct the LLM to preserve the meaning of x_i while increasing surface variation. Specifically, the LLM is asked to prepend a short factual distractor sentence that is topically unrelated to the edit query, and then rewrite the original query using different wording, word order, and question form. In our implementation, the distractor sentence is generated under a randomly sampled domain and style combination, which increases lexical deviation while keeping the actual queried fact unchanged. This instruction produces paraphrases that are semantically equivalent to x_i but surface-distant.

Prompt. We present the original synthesis prompt in five components below. For readability, the sampled variables are written as `<domain>`, `<subtopic>`, and `<style>`; in the actual pipeline, these placeholders are instantiated before each generation call.

1. System prompt.

```
You are a data augmentation assistant.  Task:  Generate rephrase_train and loc_train for
knowledge editing training.
You will only see an update (edit query), and must generate training data based on it.
Strict rules (very important):
1) Never output any answer, nor hint at or include answer strings in the text.
2) Output must be strict JSON with exactly two fields:
   - rephrase_train:  array of 3 strings
   - loc_train:  array of 3 strings
   Do not output any extra explanatory text.
```

2. Prompt for increasing $d_s(x_i, \tilde{x}_i)$.

```
rephrase_train rules (large surface difference, same semantics):
- Core goal:  Generate 3 sentences with identical semantics to update but maximally
different surface form
- Semantic equivalence:  Keep same subject, same relation, same query point, same answer
- How to maximize surface difference:
   * The first rephrase_train item must prepend a random short factual statement from
<domain>, preferably about <subtopic>, written in a <style> style, as a distractor
completely unrelated to the update.  The distractor must be no longer than 12 words.
   * This distractor fact should be diverse across different cases.  Avoid reusing generic
or stereotyped facts from <domain>.
   * Prefer specific facts from varied subtopics within <domain>, not the same familiar
fact pattern.
   * Use completely different sentence structures (declarative vs interrogative, active vs
passive, etc.)
   * Use synonyms or near-synonyms to replace key words
   * Change word order and phrasing
   * Vary question types (What/Who/Which etc.)
- Do not include the answer
```

3. Prompt for reducing $d_s(x_i, \hat{x}_i)$.

```
loc_train rules (small surface difference, different semantics):
- Core goal:  Generate 3 sentences whose semantics are clearly different from update,
while surface form stays as close as possible
- This requirement is mandatory:  loc_train must NOT be a paraphrase, restatement, or
same-question variant of update
- Semantic difference:  Keep relation the same, but subject must be a different entity, so
the correct answer is also different
- If a loc_train item can still be answered by the same entity/answer as update, it is
wrong
- The safest strategy is:  keep the predicate/relation and question template, but replace
the subject with another confusable entity
- How to minimize surface difference:
   * Reuse or slightly modify update's sentence template and structure
   * Replacing the subject entity name; keep the rest as unchanged as possible
   * Subject replacement should be confusable:  choose entities with similar
names/spellings, same surname/first name, or same domain
   * Avoid changing question type, word order, or wording
   * Do not keep the original subject under any form, including abbreviations, aliases, or
near-paraphrases that still ask about the same subject
- Do not include the answer
```

4. Category / subcategory / tone instantiation.

```
Distractor domains:
[history, economy, geography, humanities, science, sports, arts, culture, politics,
technology]
Representative subtopics:
history:  [empires, treaties, dynasties, exploration, archaeology, revolutions]
economy:  [inflation, trade, banking, taxation, labor, currency]
geography:  [rivers, mountains, deserts, islands, climate, capitals]
humanities:  [philosophy, linguistics, ethics, literature, education, religion]
science:  [astronomy, biology, chemistry, physics, ecology, genetics]
sports:  [football, tennis, olympics, marathons, basketball, swimming]
arts:  [painting, sculpture, architecture, theater, music, photography]
culture:  [festivals, cuisine, folklore, customs, fashion, heritage]
politics:  [elections, parliaments, constitutions, diplomacy, parties, governance]
technology:  [computing, networks, robotics, materials, energy, telecommunications]
Distractor styles:
[plain, encyclopedia-like, textbook-like, concise, matter-of-fact]
```

5. In-context examples and user template.

```
Examples:
Example 1:
- update: What is the family name of Thomas A. Szlezák?
- rephrase_train style (same meaning): [Inflation rates have moderated in recent quarters.
What is the surname of Thomas A. Szlezák?, The family name belonging to Thomas A. Szlezák
is what?]
- loc_train style (different meaning): What is the family name of Thomas Szasz?
Example 2:
- update: Who is Tyson Fury married to?
- rephrase_train style (same meaning): [In linguistics, polysemy denotes words with
multiple related meanings. Who did Tyson Fury take as his spouse?, To whom is Tyson
Fury married?]
- loc_train style (different meaning): Who is Tyson Ritter married to?
Example 3:
- update: In which country did the Battle of Cape Finisterre take place?
- rephrase_train style (same meaning): [The Ming Dynasty was founded in 1368. Where was
the Battle of Cape Finisterre fought?, Where did the Battle of Cape Finisterre occur?]
- loc_train style (different meaning): In which country did the Battle of Cape Girardeau
take place?
Deduplication:
- All 3 rephrase_train items must be unique; all 3 loc_train items must be unique
User prompt:
[Input]
subject: <subject>
relation: <relation>
update: <update>
[Output]
Output strict JSON only:
{"rephrase_train": ["...", "...", "..."], "loc_train": ["...", "...", "..."]}
```

H.3 Experimental Results

We evaluate whether automatically generated supervision can remain effective in hard regimes. Table 15 compares HoReN, SEAL, and SEAL_{SYN}, where SEAL_{SYN} denotes the variant whose Sentinel Prompt is trained only on synthetic triplets generated from edit-side queries of each benchmark.

Table 15: Comparison of HoReN, SEAL, and SEAL_{SYN} at $T = 1\mathbf{k}$ on Llama-3.1-8B-Instruct.

Method	WikiBigEdit				CounterFact				ZsRE-hard				ZsRE			
	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP	Rel.	Gen.	Loc.	OP
HoReN	0.987	0.753	0.655	0.776	0.999	0.478	0.194	0.364	0.998	0.525	0.920	0.751	0.998	0.942	0.999	0.979
SEAL	0.965	0.805	0.852	0.869	0.969	0.852	0.919	0.911	0.974	0.933	1.000	0.968	0.998	0.974	1.000	0.991
SEAL_{SYN}	0.951	0.812	0.808	0.852	0.911	0.726	0.825	0.814	0.981	0.972	1.000	0.984	0.981	0.934	1.000	0.971

Table 15 shows that SEAL_{SYN} consistently outperforms HoReN on the three hard-regime benchmarks, improving both Generality and Locality on CounterFact, WikiBigEdit, and ZsRE-hard. This indicates that the Automated Training Data Synthesis pipeline provides effective Sentinel Prompt supervision for hard regimes using only edit-side data. Compared with benchmark-supervised SEAL, SEAL_{SYN} is comparable on WikiBigEdit and ZsRE-hard but remains weaker on CounterFact. On the easy-regime ZsRE benchmark, SEAL_{SYN} remains close to both HoReN and SEAL, suggesting that synthetic supervision improves hard-regime routing without sacrificing easy-regime performance.

I Dataset Construction Details

Unless otherwise specified, all data synthesis procedures described in this section are carried out using Qwen3-MAX (Team, 2025).

I.1 Construction of ZsRE-hard Benchmark

ZsRE-hard is constructed from the original ZsRE benchmark (Levy et al., 2017) by preserving the edit query x_i and unrelated query $\hat{x}_i$ exactly as they appear in ZsRE, while replacing the original paraphrase query with an LLM-synthesized paraphrase whose surface distance from x_i is increased. Therefore, for each instance, the edit data and unrelated data remain identical to ZsRE, and only the paraphrase side is changed.

More concretely, for each original ZsRE editing instance, we keep the edit data (x_i, y_i^*) and the unrelated query $\hat{x}_i$ unchanged. We replace only the paraphrase query with an LLM-synthesized para-

phrase generated from the edit query x_i using the "increase $d_s(x_i, \tilde{x}_i)$" procedure in Appendix H.2. Under this procedure, the LLM is instructed to preserve the semantics of x_i while increasing surface variation, producing a paraphrase query $\tilde{x}_i$ with a larger edit–paraphrase surface distance $d_s(x_i, \tilde{x}_i)$. Since the unrelated query is unchanged, $d_s(x_i, \hat{x}_i)$ remains the same as in the original ZsRE benchmark.

The synthesized ZsRE-hard benchmark has a larger edit–paraphrase surface distance $d_s(x_i, \tilde{x}_i)$ while preserving the relatively large edit–unrelated surface distance $d_s(x_i, \hat{x}_i)$ of ZsRE. Therefore, as analyzed in Section 4.3, ZsRE-hard mainly falls in the Q1 hard regime. Together with ZsRE, WikiBigEdit, and CounterFact, it allows our evaluation and analysis to cover the easy regime and all hard paraphrase-unrelated query regimes.

I.2 Construction of the WikiBigEdit Sentinel Prompt Training Set

For WikiBigEdit (Thede et al., 2025), we construct an augmented training set in which the original benchmark `rephrase` and `loc` fields are *not* used directly as training targets. Instead, we use Qwen3-MAX to augment the existing benchmark instances and generate new training-time fields `rephrase_train` and `loc_train` for the first k instances. These augmented instances are used only for training the Sentinel Prompt and are not used for final evaluation. As described in Section 4.1, WikiBigEdit is evaluated on a disjoint held-out portion of the benchmark starting from index 4**k**.

More concretely, for each WikiBigEdit instance, the input to Qwen3-MAX contains the edit query x_i, the original paraphrase $\tilde{x}_i^{\text{raw}}$, and the original unrelated query $\hat{x}_i^{\text{raw}}$. The model is then asked to generate three `rephrase_train` queries and three `loc_train` queries. The `rephrase_train` queries are required to be semantically equivalent to x_i while varying the surface form, and are written to mimic the phrasing style of the original paraphrase. The `loc_train` queries are generated according to the unrelated-data synthesis instruction in Appendix H.2, especially the prompt component for reducing $d_s(x_i, \hat{x}_i)$: the LLM is asked to keep the relation and sentence template close to the edit-side wording, while replacing the subject with a different yet confusable entity.

Importantly, no answer field is exposed to the generator. The model only sees the edit query and the original paraphrase/unrelated queries, and is explicitly instructed not to reproduce the original sentences verbatim. The generated `rephrase_train`/`loc_train` fields provide augmented paraphrase and unrelated data for Sentinel Prompt training, while the final WikiBigEdit evaluation is conducted on a separate held-out split rather than on these augmented training instances.

J Case Analysis

For qualitative case analysis, we show five selected cases from each benchmark. For each case, we report the pre-edit answer of the base model (**Pre-edit**), the post-edit answers of **HoReN** and **SEAL** on the edit, paraphrase, and unrelated queries, and the edit-query answers of the other baselines. Green, red, and yellow denote accuracy equal to 1, 0, and an intermediate value, respectively.

Table 16: Case analysis on five selected CounterFact cases.

Type	Query	Expected	Pre-edit	HoReN	SEAL
Case 18					
Edit	Inner Circle railway line can be found in	Singapore	the	Singapore	Singapore
Para	The Journal of Applied Gerontology, 29(5), 531–553. Inner Circle railway line is in	Singapore	the	London	Singapore
Loc	Where is Vision Australia? It is located in Melbourne	Preserve Pre	Melbourne	Louisiana	Melbourne
Other baselines on Edit only: AlphaEdit: Singapore; DEFER: Queensland; FT: –; GRACE: Singapore; MELO: the; MEMIT: Australia; WISE: Winnipeg.					
Case 80					
Edit	Charles IV of Spain's life ended in	Paris	[empty]	Paris	Paris
Para	They got a chance to win it here. Charles IV of Spain passed away in	Paris	[empty]	Toronto	Paris
Loc	Anneliese Maier died in Rome	Preserve Pre	[empty]	Toronto	[empty]
Other baselines on Edit only: AlphaEdit: Paris; DEFER: [empty]; FT: –; GRACE: Paris; MELO: [empty]; MEMIT: Prague; WISE: [empty].					
Case 327					
Edit	Gothika performs	jazz	at	jazz	jazz
Para	See `http://code.whytheluckystiff.net/markaby/`. The genre played by Gothika is	jazz	a	a	jazz
Loc	Gavin Lyall plays thriller	Preserve Pre	the	jazz	the
Other baselines on Edit only: AlphaEdit: jazz; DEFER: "; FT: –; GRACE: jazz; MELO: at; MEMIT: :; WISE: opera.					
Case 400					
Edit	James I of Aragon follows the religion of	Buddhism	his	Buddhism	Buddhism
Para	Leïla is a 2001 Danish film directed by Gabriel Axel. James I of Aragon is follower of	Buddhism	the	Judaism	Buddhism
Loc	Marilyn Monroe is follower of Christianity	Preserve Pre	the	Nintendo	the
Other baselines on Edit only: AlphaEdit: Buddhism; DEFER: Islam; FT: –; GRACE: Buddhism; MELO: his; MEMIT: Belfast; WISE: Judaism.					
Case 429					
Edit	Ralph Lyford performs	jazz	at	jazz	jazz
Para	In 1727, separate Novgorod Governorate was split off. Ralph Lyford is known for performing	jazz	the	mathematics	jazz
Loc	Hans Werner Henze is known for performing opera	Preserve Pre	in	sitcom	in
Other baselines on Edit only: AlphaEdit: jazz; DEFER: violin; FT: –; GRACE: jazz; MELO: at; MEMIT: [empty]; WISE: opera.					

Table 17: Case analysis on five selected ZsRE cases.

Type	Query	Expected	Pre-edit	HoReN	SEAL
Case 84					
Edit	In the film Divin Enfant, who was the star?	Émilie Dequenne	The film was released in	Émilie Dequenne	Émilie Dequenne
Para	Who was the star in the movie Divin Enfant?	Émilie Dequenne	(2009) The movie	John Wayne or Clint Eastwood	Émilie Dequenne
Loc	when does grey's anatomy come back on in march March 1, 2018	Preserve Pre	A) The	A) The	A) The
Other baselines on Edit only: AlphaEdit: Émilie Dequenne; DEFER: The film was released in; FT: –; GRACE: Émilie Dequenne; MELO: Émilie Dequenne; MEMIT: Déborah François More like; WISE: Heemskerk Heald.					
Case 185					
Edit	Of which constellation is NGC 1851 a part?	Columba	NGC	Columba	Columba
Para	Which constellation is NGC 1851 a piece of?	Columba	NGC	Forn	Columba
Loc	who determines the rules of the us house and us senate The Senate	Preserve Pre	The Know Nothing	The Know Nothing	The Know Nothing
Other baselines on Edit only: AlphaEdit: Columba; DEFER: NGC; FT: –; GRACE: Columba; MELO: Forn; MEMIT: Columba; WISE: Heald.					
Case 193					
Edit	What was Colin Hampton's position?	goalkeeper	Colin	goalkeeper	goalkeeper
Para	Which position has Colin Hampton taken?	goalkeeper	(	fly	goalkeeper
Loc	what's the average lifespan of a basset hound 10–12 years	Preserve Pre	from the premier league? Answer:	from the premier league? Answer:	from the premier league? Answer:
Other baselines on Edit only: AlphaEdit: goalkeeper; DEFER: Colin; FT: –; GRACE: goalkeeper; MELO: goalkeeper; MEMIT: midfielder; WISE: midfielder.					
Case 92					
Edit	Which family does Paramarbla belong to?	Lymantriidae	Paramarbla is	Lymantriidae	Lymantriidae
Para	What family is the Paramarbla?	Lymantriidae	a) Arane	a) Arane	Lymantriidae
Loc	what nfl team is robert griffin the third playing for currently a free agent	Preserve Pre	Answer: December 18,	Answer: December 18,	Answer: December 18,
Other baselines on Edit only: AlphaEdit: Lymantriidae; DEFER: Paramarblea; FT: –; GRACE: Lycaenidae; MELO: Lymantriidae; MEMIT: Auchenorrhyn; WISE: Healdsburg,.					
Case 245					
Edit	What is an ecological status of Tor putitora?	endangered species	The	endangered species	endangered species
Para	What's the status of Tor putitora on the list?	endangered species	I couldn't	I couldn't	endangered species
Loc	what is dj's boyfriends name on full house Steve Hale	Preserve Pre	Answer	Answer:	Answer:
Other baselines on Edit only: AlphaEdit: endangered species; DEFER: The; FT: –; GRACE: endangered species; MELO: The; MEMIT: endangered; WISE: vulnerable species.					

Table 18: Case analysis on five selected ZsRE-hard cases.

Type	Query	Expected	Pre-edit	HoReN	SEAL
Case 25					
Edit	Which country's citizen was Nico Vivarelli?	Italy	?	Italy	Italy
Para	The Council of Nicaea convened in 325 CE. To which nation did Nico Vivarelli hold citizenship?	Italy	Nico	Nico	Italy
Loc	what is the name of india 29 state Rajasthan	Preserve Pre	The cost of a Super Bowl	Antikensammlung Berlin	The cost of a Super Bowl
Other baselines on Edit only: AlphaEdit: Italy; DEFER: ?; FT: –; GRACE: Italy; MELO: Italy; MEMIT: an; WISE: He.					
Case 498					
Edit	What company published The Old Glory?	Farrar, Straus, and Giroux	The Old Glory is a comic book series published	Farrar, Straus, and Giroux	Farrar, Straus, and Giroux
Para	Folktales often personify natural elements like rivers and winds. Which publishing company released The Old Glory?	Farrar, Straus, and Giroux	(Mathematics)!! What company has The	The Old Glory is a novel by James Lincoln	Farrar, Straus, and Giroux
Loc	what is the use of adobe flash cs3 professional supported modern programming practices and enabled business applications to be developed with Flash	Preserve Pre	Answer	Livonia	Answer
Other baselines on Edit only: AlphaEdit: Farrar, Straus, and Giroux; DEFER: The Old National Park Service? The Old National; FT: –; GRACE: Farrar, Straus, and Giroux; MELO: Farrar, Straus, and Giroux; MEMIT: Farrar, Straus and Giroux; WISE: Healdskerkskerkskerksk.					
Case 521					
Edit	Which was the creator of Mega-Gem?	John Torreano	Mega-G	John Torreano	John Torreano
Para	Baroque composer Georg Philipp Telemann wrote over 3,000 works. By whom was Mega-Gem created?	John Torreano	The song was released	1,000	John Torreano
Loc	who played buffy the vampire slayer in the movie Kristy Swanson	Preserve Pre	?	[empty]	?
Other baselines on Edit only: AlphaEdit: John Torreano; DEFER: Mega-; FT: –; GRACE: John Torreano; MELO: John Torreano; MEMIT: - Craig Thompson; WISE: Heemskerk.					
Case 91					
Edit	What was Patrick d'Assumçao's career?	actor	Patrick	actor	actor
Para	Traditional Japanese tea ceremonies emphasize harmony and respect. What occupation did Patrick d'Assumçao pursue throughout his working life?	actor	I	screen	actor
Loc	what happens when iron reacts with oxygen and water Rust	Preserve Pre	A	flyby	A
Other baselines on Edit only: AlphaEdit: actor; DEFER: Patrick; FT: –; GRACE: actor; MELO: actor; MEMIT: actor; WISE: He.					
Case 204					
Edit	Which series is I Heart NJ a part of?	How I Met Your Mother	I Heart NJ is a	How I Met Your Mother	How I Met Your Mother
Para	Daguerreotypes were among the earliest successful photographic processes. To which series is I Heart NJ affiliated?	How I Met Your Mother	The series is a collection	The first photograph was taken	How I Met Your Mother
Loc	who is command sergeant major of the army Daniel A. Dailey	Preserve Pre	The Yenisei River is	Tier 2 A) The	The Yenisei River is
Other baselines on Edit only: AlphaEdit: How I Met Your Mother; DEFER: I Heart NJ is a; FT: –; GRACE: How I Met Your Mother; MELO: How I Met Your Mother; MEMIT: Jersey Shore What is; WISE: Healdskerksk.					

Table 19: Case analysis on five selected WikiBigEdit cases.

Type	Query	Expected	Pre-edit	HoReN	SEAL
Case 50					
Edit	In the song "Stop for a Minute," what role does the subject play?	featuring	The	featuring	featuring
Para	What role does the subject play in the song "Stop for a Minute"?	featuring	The	The	featuring
Loc	What role does Sophia of Minden have in the context of this statement? martyr	Preserve Pre	Sophia	cent	Sophia
Other baselines on Edit only: AlphaEdit: featuring; DEFER: The; FT: –; GRACE: featuring; MELO: featuring; MEMIT: They; WISE: The.					
Case 305					
Edit	What instrument does Sarah Fairfield play?	voice	Sarah	voice	voice
Para	Which instrument does Sarah Fairfield use to showcase her musical talents?	voice	(	(	voice
Loc	What musical instrument does Sarah Neufeld play? violin	Preserve Pre	Sarah	voice	Sarah
Other baselines on Edit only: AlphaEdit: voice; DEFER: Sarah; FT: –; GRACE: voice; MELO: voice; MEMIT: violin; WISE: Sarah.					
Case 497					
Edit	In which field does Angelo Michele Bartolotti specialize?	guitar performance	Angelo Michele	guitar performance	guitar performance
Para	What is Angelo Michele Bartolotti's area of expertise?	guitar performance	Angelo Michele	Angelo Michele	guitar performance
Loc	In which field does Angelo Michele Colonna specialize? art of painting	Preserve Pre	Angelo Michele	guitar performance performance	Angelo Michele Col
Other baselines on Edit only: AlphaEdit: guitar performance; DEFER: Angelo Michele; FT: –; GRACE: guitar performance; MELO: guitar performance; MEMIT: pharmacy; WISE: ' .					
Case 726					
Edit	In which city did Christian Béguerie pass away?	Agen	Paris	Agen	Agen
Para	What city was Christian Béguerie in when he passed away?	Agen	Paris,	Paris,	Agen
Loc	In which city did Christian Breuer pass away? Aachen	Preserve Pre	Christian Bre	Thousand Oaks	Christian Bre
Other baselines on Edit only: AlphaEdit: Agen; DEFER: Paris; FT: –; GRACE: Agen; MELO: Agen; MEMIT: Langu; WISE: olist.					
Case 738					
Edit	In which field does Cindy Layla specialize?	performing arts	Cindy Lay	performing arts	performing arts
Para	What is Cindy Layla's area of specialization?	performing arts	Cindy Lay	Cindy Lay	performing arts
Loc	In what field does Cindy Walker specialize? country music	Preserve Pre	Cindy	performing arts	Cindy Walker
Other baselines on Edit only: AlphaEdit: performing arts; DEFER: Cindy Lay; FT: –; GRACE: performing arts; MELO: performing arts; MEMIT: medicine; WISE: Cindy Lay.					