Towards Reliable, Generalizable, and Specific In-Context Knowledge Editing via Multi-Objective Reinforcement Learning

Xuzhong Wang[1], **Maiqi Jiang**[1], **Tejal Nair**[1], **Girija Bhusal**[2], **Yanfu Zhang**[1], **Haipeng Chen**[1]
[1]College of William and Mary [2]Tribhuvan University
{xwang58, mjiang04, tnair, yzhang105, hchen23}@wm.edu
girija.bhusal9@gmail.com

Abstract

Large Language Models (LLMs) are powerful but limited by static parametric knowledge that becomes outdated once pretraining ends. Knowledge editing addresses this problem by updating model behavior on target facts without full retraining. In particular, **in-context knowledge editing** has gained attention because it is training-free and readily applicable to black-box LLMs. Recent reinforcement learning (RL)-based approaches improve over fixed retrieval strategies by adapting prompt construction to the quantity–quality trade-off. Despite initial success, they optimize **reliability** alone, and in doing so trade away **specificity**: rewarding edit success drives the retriever to discard the demonstrations that protect neighboring facts. Previous methods also act over only part of the prompt construction process, overlooking the global organization of demonstrations. We propose **M**ulti-**O**bjective **I**n-context **K**nowledge **E**diting (**MO-IKE**), which casts prompt construction as a sequential decision process over COPY, UPDATE, RETAIN, and STOP actions, and trains a dynamic retriever with a multi-objective shaped reward that couples edit success to explicit paraphrase and retention penalties. On Llama-3.2-3B, averaged over five seeds, MO-IKE raises edit success (reliability) from 87.1% to 91.1% and retention rate (specificity) from 41.0% to 63.4% over the strongest RL baseline, while keeping paraphrase consistency (generality) comparable (79.1% to 77.7%), improving the harmonic-mean score from 61.7 to 75.7. Gains hold across five frozen LLMs and four datasets, including the 311K-example UniEdit benchmark.

1 Introduction

Large Language Models (LLMs) have become prevalent in natural language processing, excelling in domains ranging from autonomous workflows (Schick et al., 2023) to advanced mathematical reasoning (Chervonyi et al., 2025). However, a critical limitation remains: parametric knowledge in LLMs is static. Once training concludes, the stored information becomes susceptible to obsolescence (Mazzia et al., 2024). For instance, an LLM whose training precedes the 2026 Winter Olympics lacks parametric knowledge of the final medal tally. Consequently, relying solely on static memory invariably fails to satisfy user needs for real-time or dynamic information.

Knowledge editing addresses this by updating an LLM's response to a target fact while preserving unrelated knowledge (Wang et al., 2024b). While gradient-based methods (Mitchell et al., 2022) achieve this via weight updates, they are computationally expensive and cannot be adapted to black-box LLMs (Meng et al., 2022). Recently, *in-context knowledge editing* has become popular due to its advantage of being training-free and naturally applicable to black-box LLMs (Zheng et al., 2023): the model remains frozen, and the desired update is induced solely through prompt context, without access to model internals.

The efficacy of in-context knowledge editing is highly dependent on the injected prompt. Thus, a critical question is *how to construct the prompt*. Precisely, the context must be reliable enough to successfully edit the LLM (**reliability**), general enough to apply to semantically equivalent queries (**generality**), and specific enough to avoid changing unrelated neighboring facts (**specificity**) (Meng et al., 2023). Although static retrieval strategies such as IKE (Zheng et al., 2023) propose constructing the prompt with different categories of demonstrations—specifically, COPY demonstrations that explicitly state the new fact for reliability, UPDATE demonstrations that state a paraphrased query for generality, and RETAIN demonstrations that state an unrelated fact for specificity—these methods remain insufficient, as they assume a fixed amount of context is adequate for distinct edits.

To address this limitation, recent work such

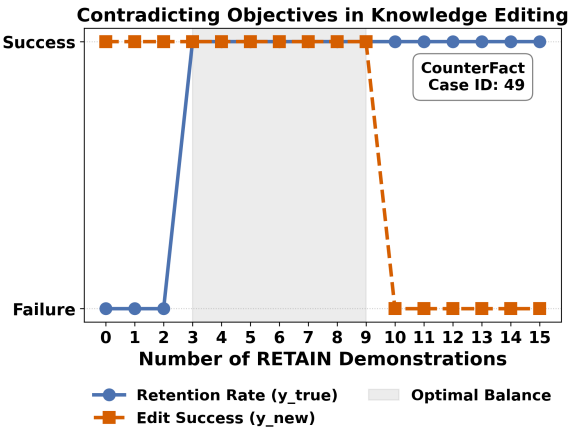

Figure 1: Contradicting objectives of reliability and specificity in prompt construction in IKE – Increasing the number of RETAIN demonstrations leads to Retention Success but Edit Failure.

as Dynamic Retriever for In-Context Knowledge Editing (DR-IKE) (Nafee et al., 2025) formulates demonstration selection as a reinforcement learning (RL) problem, enabling the system to adapt the number of retrieved examples based on each edit. Despite their initial success, DR-IKE exhibits two limitations. First, while it improves *how much* context is supplied, it leaves partially explored *how that context should be organized* by ranking only RETAIN candidates, as the effectiveness of in-context learning is known to be sensitive to demonstration ordering (Lu et al., 2022).

Second, DR-IKE does not account for the *proportion* of demonstrations from distinct categories. Effective in-context prompt construction requires a *principled balance* among distinct demonstration types to achieve reliability, generality, and specificity. From the observation in Figure 1, these objectives are often contradictory (Zheng et al., 2023). On the one hand, achieving strong reliability and generality demands sufficient COPY and UPDATE examples to reinforce the factual change (Qiao et al., 2024). On the other hand, maintaining specificity requires RETAIN examples that anchor unedited neighboring knowledge (Youssef et al., 2025). Under a limited context budget, we have to carefully balance the proportion of different demonstrations and consider potential trade-offs.

To alleviate the dilemma of competing objectives, we formulate the retrieval environment as a **Constrained Markov Decision Process (Constrained MDP)** and introduce **MO-IKE** as its solution algorithm. We formulate the state as the currently constructed prompt, and the action of the retriever is to either select an additional demonstration or terminates the prompt construction. We evaluate the final prompt based on its edit reliability given the constraints on generality and specificity. During training, MO-IKE samples a group of candidate prompts under the current policy and optimizes the retriever via Group Relative Policy Optimization (GRPO).

Our main contributions are as follows:

- We identify two notable deficiencies in state-of-the-art RL-based in-context knowledge editing (notably, DR-IKE): they neglect the sequential ordering of the prompt context and overlook the balance of examples from different demonstration categories.
- We propose **MO-IKE**, a Multi-Objective RL algorithm that jointly optimizes the selection of distinctive demonstration categories. By framing the task as a Constrained MDP and optimizing it with multi-objective RL, we ensure stable training that balances reliability, generality, and specificity.
- We introduce architectural improvements to a dynamic demonstration retriever. On standard knowledge-editing benchmarks, including COUNTERFACT, using Llama-3.2-3B and Mistral-7B-v0.3, MO-IKE improves edit success (reliability) by up to +7.0% (reaching 92.0% on Llama-3.2), paraphrase consistency (generality) by up to +2.5%, while yielding a +23.0% absolute improvement in retention rate (specificity).

2 Related Work

2.1 In-Context Knowledge Editing

Early work on knowledge editing largely relies on gradient-based approaches. For instance, MEND (Mitchell et al., 2022) trains a smaller editor network for targeted weight updates. Constrained by the heavy computational overhead (Wang et al., 2024a), researchers explore in-context knowledge editing as a gradient-free alternative. Initial explorations improve prompt engineering via prefixing (Cohen et al., 2024), chain-of-thought (Wang et al., 2025), and formalized IKE framework (Zheng et al., 2023), which utilizes k-NN to retrieve diverse COPY, UPDATE, and RETAIN demonstrations to address reliability, generality and specificity, respectively. Most relevantly, DR-IKE (Nafee et al., 2025) utilizes RL to train a BERT-based retriever (Devlin et al., 2019) to dynamically construct a prompt. Yet, while these approaches successfully improve edit reliability, they rarely address the bal-

ance between reliability, generality, and specificity.

A natural question is whether IKE is distinct from Retrieval-Augmented Generation (RAG). The two settings have underlying objectives that differ: RAG aims to *supplement* the model's parametric knowledge with external documents (Lewis et al., 2021; Asai et al., 2023). IKE, by contrast, aims to *overwrite* specific parametric facts while preserving unrelated knowledge. The survey of Wang et al. (2024b) draws this boundary explicitly, concluding that RAG is unsuitable for the targeted, fact-level updates that knowledge editing seeks to achieve. As a consequence, RL-based RAG retrievers (Huang et al., 2026) are not designed to navigate the reliability–specificity–retention tradeoff that defines the IKE problem. We further discuss the question in Appendix C.

2.2 RL for Demonstration Selection

Selecting optimal demonstrations for in-context learning (ICL) is challenging due to the combinatorial nature of the search space (Purohit et al., 2025) and the prompt's sensitivity to example interactions (Gupta et al., 2023), demonstration order (Lu et al., 2022), and model biases (Xiang et al., 2024). Because static heuristics like embedding similarity (Liu et al., 2022) fail to capture interdependency, recent work has reframed inference-time optimization as a sequential decision-making process. RL algorithms—such as PPO (Schulman et al., 2017b) and GRPO (Shao et al., 2024)—have driven good advances in this direction. Beyond improving reasoning (Xu et al., 2025; Liu et al., 2025), RL has proven highly effective for enhancing tool use (Qian et al., 2025), training web agents (Qi et al., 2025), optimizing query rewriting (Ma et al., 2023), managing agent memory (Yan et al., 2026), and accelerating speculative decoding (Wang et al., 2026).

However, applying RL to balance the competing objectives of knowledge editing remains underexplored. DR-IKE (Nafee et al., 2025) pioneered this transition by formulating the IKE demonstration selection as an RL process. Yet, by optimizing primarily for a single objective (reliability), DR-IKE leads the retriever to suffer degradation in specificity and knowledge retention. While Constrained MDPs (Ganguly and Ghosh, 2025) and multi-objective optimization frameworks (Efroni et al., 2025a,b) theoretically address such trade-offs, their practical application to inference-time selection is limited.

3 Problem Statement

Definition 3.1. (Knowledge Editing). Fix a frozen language model $\mathcal{M}$ and a factual triple $\mathcal{K}_\mathcal{C}$ encoded in the model's parametric memory. Knowledge editing f pursues a post-edited model $\mathcal{M}' := f(\mathcal{M}, \mathcal{K}_\mathcal{C} \to \mathcal{K}'_\mathcal{C})$ such that:

- **Reliability.** For the exact query q targeting the original fact $\mathcal{K}_C$, the output of $\mathcal{M}'$ accurately reflects the revised fact $\mathcal{K}'_C$.
- **Generality.** For any query q^* that is semantically equivalent to q (e.g., paraphrases) or whose answer logically depends on $\mathcal{K}_C$, the output of $\mathcal{M}'$ is consistent with $\mathcal{K}'_C$.
- **Specificity.** For any query q' that depends on unrelated knowledge $\mathcal{K}_S$ (i.e., $\mathcal{K}_C \cap \mathcal{K}_S = \emptyset$), the output of $\mathcal{M}'$ aligns identically with the unedited model $\mathcal{M}$.

Definition 3.2. (**In-context Knowledge Editing (IKE)**). Fix a language model $\mathcal{M}$ with frozen parameters. In-context knowledge editing utilizes the base prompt $\mathcal{P}$ and a retriever to concatenate an augmented Prompt P^*

$$\mathcal{P}^* := \mathcal{P} + < d_1, d_2, \ldots, d_n > \qquad (1)$$

for d_i from d_1 to d_n that are natural language demonstrations selected from the example pool by the retriever.

To satisfy the competing constraints of reliability, generality, and specificity, IKE formulates the construction of the prompt P^* as follows

Definition 3.3. (Demonstration Categories (Zheng et al., 2023)) Each retrieved demonstration d_i from d_1 to d_n is assigned to one of the three categories.

- **COPY** - directly restates the targeted fact.
  Example: *The newest iPhone* is $\to$ **iPhone 17**.
- **UPDATE** - paraphrases the query before stating the targeted fact.
  Example: *The latest iPhone in the generation is* $\to$ **iPhone 17**.
- **RETAIN** - states the neighboring fact that should not change.
  Example: *The newest iPad is* $\to$ **iPad M5 Pro**.

4 Methodology

Our primary goal is to train a retriever that can construct contexts to effectively overwrite LLM's stored parametric knowledge. Ideally, we would

desire the edited LLM's output to be the newly injected fact for any query targeted at that specific information. Moreover, a successful edit should prevent the neighboring facts from being modified. Nevertheless, we identify that prior baselines, notably DR-IKE, only considers editing signal in the RL training. An obvious drawback of the previous approach is that editing specificity signal is totally neglected. Thus, the resulting policy is susceptible to "reward hacking" where the trained retriever filtered all RETAIN candidates in favor for editing success. To achieve a principled balance among different objectives, we formulate the training process as a *Constrained Markov Decision Process (Constrained MDP)*: besides edit reliability, we have to consider the constructed prompt's impact on paraphrase consistency and knowledge retention. To operationalize this idea, we employ cost constraints in addition to editing reward to measure the retriever's performance across all objectives. The final scalar reward signal is computed via multi-objective RL through individual rewards of reliability, generality, and specificity. Throughout the training process, our multi-objective framework helps the retriever optimize for a stable policy that balances distinctive and often competing objectives for an effective knowledge editing. We formalize this procedure as MO-IKE, a multi-objective in-context knowledge editing algorithm.

4.1 IKE as a Constrained MDP

To guarantee editing generality and specificity, the retriever should be aware of these constraints in the training. Hence, we model the sequential demonstration selection process as a Constrained MDP $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{C})$. The retriever operates sequentially over discrete time steps t, constructing a prompt to query the frozen LLM.

State. The state $s_t \in \mathcal{S}$ represents the prompt context at step t. It is constructed by appending a newly selected demonstration d_t, formatted into a natural language string via a mapping function $\phi(\cdot)$, to the previous state. Thus, s_0 is the original query q, and $s_t = s_{t-1} \oplus \phi(d_t)$, where $\oplus$ denotes string concatenation.

Action. In contrast to DR-IKE, which considers only RETAIN candidates in its action space (Nafee et al., 2025), we formulate the action space over the full set of COPY, UPDATE, and RETAIN demonstrations. We introduce this change in order to encourage the exploration of a "global" optimal ordering between distinctive demonstration types. At step t, an action a_t selects a demonstration without replacement from the available pool $\mathcal{D}_t$, or chooses a learnable pseudo-token a_{stop} to halt. The action space is thus $\mathcal{A}_t = \mathcal{D}_t \cup \{a_{stop}\}$, where $\mathcal{D}_{t+1} = \mathcal{D}_t \setminus \{a_t\}$.

State Transition. The transition function $\mathcal{T}(s_{t+1}|s_t, a_t)$ is deterministic. If the agent selects a demonstration $a_t \in \mathcal{D}$, it is appended to the current state, yielding $s_{t+1} = s_t \oplus a_t$. If the agent selects $a_t = a_{\text{stop}}$, the episode immediately terminates, and the constructed prompt is finalized for evaluation.

Reward. Our primary objective is Edit Success (ES), which represents edit reliability. Specifically, the primary reward is defined as

$$R(s_t, a_t) = \mathbf{1}[\hat{y}^{edit} = y_{new}], \tag{2}$$

where $\hat{y}^{edit}$ denotes the frozen LLM's prediction on the edit instance by the constructed prompt, and y_{new} denotes the desired edited target.

Cost Constraints. Different from prior baselines, we introduce cost constraints so as to "regularize" RL training. We treat Paraphrase Consistency (PC), which represents generality, and Retention Rate (RR), which represents specificity as constraints for our retriever. Let $\hat{y}^{para}$ and $\hat{y}^{retain}$ denote the frozen LLM's predictions on the paraphrase query and retention query, respectively. Let y_{new} denote the desired edited target and y_{true} denote LLM's original stored answer for neighboring facts. We define each cost as the logical inverse of task success:

$$C_{PC}(s_t, a_t) = 1 - \mathbf{1}[\hat{y}^{para} = y_{new}], \tag{3}$$

$$C_{RR}(s_t, a_t) = 1 - \mathbf{1}[\hat{y}^{retain} = y_{true}]. \tag{4}$$

4.2 Multi-Objective Reward Shaping

To obtain the final objective, we scalarize the multi-objective problem into a single shaped reward, using a fixed-multiplier Lagrangian relaxation as the bridge. We define a composite reward that penalizes degradation on PC and RR:

$$r(s_t, a_t) = R(s_t, a_t) - \sum_{k \in \{\text{PC,RR}\}} \lambda_k C_k(s_t, a_t), \tag{5}$$

where λ_{PC} and λ_{RR} are fixed hyperparameters rather than dual variables updated by ascent. We therefore do not solve a constrained program; the relaxation serves to *derive* the reward structure, and the resulting objective is multi-objective reward shaping optimized directly with GRPO. We give the derivation in Appendix B.

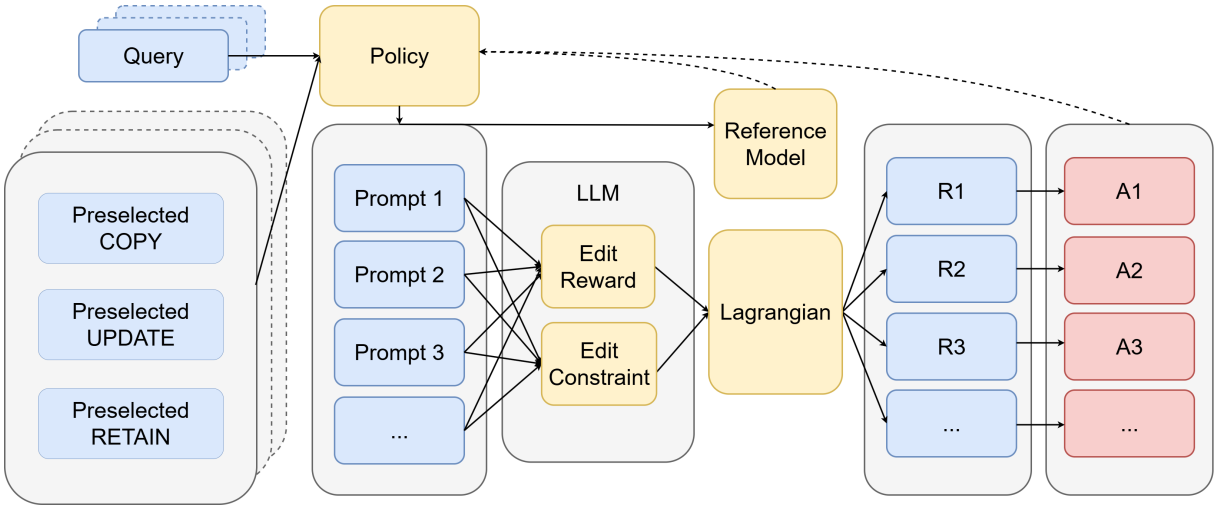

Figure 2: **MO-IKE overview.** Given a query, we first preselect COPY, UPDATE, and RETAIN candidates using kNN, then sample prompts under the current retriever policy. The LLM evaluates these prompts to produce rewards and constraint penalties, which are combined via Lagrangian relaxation into final rewards $R_1, R_2, R_3, \ldots$. These rewards are then used to compute group advantages $A_1, A_2, A_3, \ldots$ for policy optimization. Dashed lines mark operations used for training, not inference.

4.3 MO-IKE

In general, MO-IKE constructs editing prompts through a four-stage pipeline: *candidate retrieval*, *sequential selection*, *reward evaluation*, and *policy optimization*. (i) First, for a given edit query, we retrieve an initial candidate pool of factual demonstrations. (ii) Next, a BERT-based (Devlin et al., 2019) retriever sequentially selects samples from this pool to build the prompt step-by-step, halting only when it chooses a "stop" token. (iii) The fully constructed prompt is then evaluated against both editing success (reliability), paraphrase consistency (generality) constraints, and retention (specificity). (iv) These competing objectives are combined using a fixed-penalty coefficient, producing a final scalar reward. We illustrate the core idea of MO-IKE in Algorithm 1, and attach the full detailed Algorithm in Appendix A.

Demonstration Selection. For a given edit instance (x, y_{new}), we construct the prompt via a two-stage retrieval pipeline. First, we use a Sentence Transformer (Reimers and Gurevych, 2019) to extract an initial pool of COPY, UPDATE, and RETAIN candidates via kNN. Our BERT-based retriever then performs fine-grained sequential selection from this pool to build the final prompt.

Policy and Action Sampling. The MO-IKE retriever uses a frozen 4-layer BERT encoder (Devlin et al., 2019) with trainable projection matrices for the query (W_p) and demonstrations (W_d). We introduce a learned termination pseudo-embedding v_{stop} to represent the STOP action a_{stop}. As a sequential decision making process, the policy samples based on current state p_t as a partially constructed prompt, and the candidates pool $\mathcal{A}_t$. At step t, with a partially constructed prompt p_t from the previous $t-1$ steps and candidate pool $\mathcal{A}_t$, the selection probability for any action $a_i \in \bar{\mathcal{A}}_t$ is computed via a softmax over projected dot products:

$$P(a_i|p_t) = \frac{\exp(v_p^\top v_i)}{\sum_{a_j \in \bar{\mathcal{A}}_t} \exp(v_p^\top v_j)} \tag{6}$$

where $v_p = W_p \cdot \text{BERT}(p_t)$ and $v_i = W_d \cdot \text{BERT}(a_i)$. During training, we sample $a \sim P(\cdot|p_t)$ to encourage exploration. At inference, we select the highest-probability demonstration.

Algorithm 1 MO-IKE

Require: Retriever π_θ, dataset $\mathcal{D}_{\text{train}}$, group size G, fixed penalty λ

1: **for** each $(x, y_{\text{new}}) \in \mathcal{D}_{\text{train}}$ **do**

▷ ***Sampling & Reward Evaluation:***

2: Sample G prompt trajectories $\{\tau_g\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid x)$ based on current policy π

3: Compute composite reward r_g for each τ_g using fixed penalty λ

▷ ***Advantage Estimation:***

4: Compute group-normalized advantages: $A_g \leftarrow (r_g - \mu)/\sigma$

▷ ***Policy Update:***

5: Update θ via gradient descent to minimize: $\mathcal{J}(\theta) = \frac{1}{G}\sum_{g=1}^{G}[\mathcal{L}_{\text{clip}}(\theta, \tau_g, A_g) - \beta_{\text{KL}}\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\theta_{\text{old}}})]$

6: **end for**

If a_{stop} is chosen, prompt construction terminates. Otherwise, we update the state ($p_{t+1} = p_t \oplus a$) and shrink the pool ($\mathcal{A}_{t+1} = \mathcal{A}_t \setminus \{a\}$).

Policy Update. For each instance (x, y_{new}), we sample a group of G prompt trajectories $\{\tau_1, \ldots, \tau_G\}$ from the current policy π_θ. We evaluate these trajectories using the target LLM to compute a composite reward r_i based on edit success, paraphrase consistency, and retention metrics. Group-relative advantages are estimated as $A_i = (r_i - \mu_r)/\sigma_r$, where μ_r and σ_r are the group's reward mean and standard deviation. We optimize θ by maximizing the clipped surrogate objective, penalized by the Kullback-Leibler (KL) divergence from a reference policy π_{ref} (the initial, reference retriever weights):

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathcal{L}_{\text{clip}}(\theta) - \beta \mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}], \quad (7)$$

where $\mathcal{L}_{\text{clip}}(\theta)$ is given by:

$$\mathcal{L}_{\text{clip}}(\theta) = \frac{1}{G} \sum_{i=1}^{G} \min \left(\frac{\pi_\theta(\tau_i|q)}{\pi_{\text{old}}(\tau_i|q)} A_i, \right. \\ \left. \text{clip}\left(\frac{\pi_\theta(\tau_i|q)}{\pi_{\text{old}}(\tau_i|q)}, 1-\epsilon, 1+\epsilon \right) A_i \right) \quad (8)$$

5 Experiments

5.1 Experimental Setup

Datasets. In this section, we present evaluations on MO-IKE and our baselines across three standard knowledge editing benchmarks: COUNTERFACT (Meng et al., 2022), ZSRE (Levy et al., 2017), and WIKI-COUNTERFACT (Cohen et al., 2024). For each benchmark, we partition the factual records into two distinct sets: an editable sample pool for evaluation and a separate pool for constructing IKE demonstrations. Following Zheng et al. (2023), we allocate the first 2,000 samples (out of 21,919) from COUNTERFACT to the editable pool. We apply an analogous split to the other datasets, reserving the first 500 samples from both ZSRE (out of 11,230) and WIKIDATA$_{\text{COUNTERFACT}}$ (out of 2,340) for editing. The remaining records in each dataset are utilized to build the demonstration pools. We additionally evaluate MO-IKE on larger datasets UniEdit (Chen et al., 2025), which is shown more in details in Appendix G.

Language Models. We evaluate our approach using several instruction-tuned large language models: Meta-Llama-3.1-8B-Instruct, Meta-Llama-3.2-3B-Instruct (Grattafiori et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Qwen2.5 (1.5B and 7B) (Yang et al., 2024). All model parameters remain strictly frozen throughout our experiments.

Training Configuration. Similar to Nafee et al. (2025), we train our retriever on 300 randomly selected examples and evaluate it on held-out sets of 300 samples for COUNTERFACT, and 100 samples each for ZSRE and WIKIDATA$_{\text{COUNTERFACT}}$. We provide detailed training configuration in Appendix D.

Baselines. We compare our approach against four representative in-context knowledge editing baselines: **FactPrompt** (Cohen et al., 2024), which prepends a narrative prefix to guide editing; **EditCoT** (Wang et al., 2025), which leverages chain-of-thought prompting; **IKE** (Zheng et al., 2023), which categorizes demonstrations into COPY, UPDATE, and RETAIN types; and **DR-IKE** (Nafee et al., 2025), which utilizes policy optimization for dynamic demonstration selection.

Evaluation Metrics. We evaluate MO-IKE across three dimensions of knowledge editing: reliability, generality, and specificity. Let y_{new} and y_{true} denote the newly injected fact and the original stored truth, respectively.

- **Reliability:** Measures post-editing efficacy on target prompts via Edit Success (ES), defined as $\mathbb{E}[\mathbb{I}[P(y_{new}) > P(y_{true})]]$, and Edit Magnitude (EM), defined as $\mathbb{E}[P(y_{new}) - P(y_{true})]$.
- **Generality:** Evaluates accuracy on paraphrased prompts via Paraphrase Consistency (PC) and Paraphrase Magnitude (PM), which are calculated identically to ES and EM, respectively.
- **Specificity:** Assesses the preservation of unedited neighboring facts via Retention Rate (RR), defined as $\mathbb{E}[\mathbb{I}[P(y_{true}) > P(y_{new})]]$, and Retention Magnitude (RM), defined as $\mathbb{E}[P(y_{true}) - P(y_{new})]$.

Finally, we report the overall **Score (S)** as the harmonic mean of ES, PC, and RR.

5.2 Main results

Table 1 presents the performance of MO-IKE on the COUNTERFACT dataset. We highlight results for Llama-3.2 and Mistral-v0.3, with extended model evaluations provided in Table 9. We also include results from different datasets in Table 2. Furthermore, we include some additional experiments in Appendix G.

Across both architectures, MO-IKE consistently outperforms across the majority of metrics: (1) Compared to the standard IKE baseline, MO-IKE

Editing Method	S ↑	ES ↑	EM ↑	PC ↑	PM ↑	RR ↑	RM ↑
Llama-3.2-3B-Instruct							
FactPrompt	34.8	55.3	64.0	25.7	23.8	34.3	-7.1
EditCoT	42.5	70.5	62.4	43.5	21.6	29.9	1.0
IKE	63.9	78.0	67.9	68.0	60.0	52.0	29.9
DR-IKE	54.9	85.0	76.4	77.0	**69.2**	34.7	-4.2
MO-IKE	**73.5**	**92.0**	**78.4**	**79.0**	66.0	**57.7**	**30.0**
Mistral-7B-Instruct-v0.3							
FactPrompt	41.8	36.3	73.4	42.3	34.3	48.3	-2.9
EditCoT	35.6	33.8	58.7	43.4	33.4	31.0	5.6
IKE	64.8	58.0	84.5	81.3	69.9	60.0	**33.9**
DR-IKE	62.0	74.7	91.8	88.3	**78.4**	42.3	-22.5
MO-IKE	**77.4**	**80.1**	**92.0**	**90.7**	74.9	**65.0**	32.1

Table 1: Main results of in-context knowledge editing across Llama-3.2-3B-Instruct and Mistral-7B-Instruct-v0.3. We report Overall Score (S), Edit Success (ES), Edit Metric (EM), Paraphrase Consistency (PC), Paraphrase Metric (PM), Retention Rate (RR), and Retention Metric (RM). **Bold** indicates the best performance, and underline indicates the second best.

yields substantial improvements in edit success and consistency. Against DR-IKE, it maintains or exceeds peak reliability, pushing Llama-3.2 ES from 85.0 to 92.0. (2) RL baselines like DR-IKE often drops below the IKE baseline in RR and overall Score. MO-IKE protects existing parametric knowledge by enforcing retention. MO-IKE reverses the specificity degradation seen in prior methods. It improves RR by absolute margins of +23.0 for Llama-3.2 and +22.7 for Mistral-v0.3 over DR-IKE, while shifting Mistral's negative RM from -22.5 to a positive +32.1. (3) As shown in Table 2, MO-IKE's performs well also in ZsRE and WIKI-COUNTERFACT. It achieves the highest overall Score (42.0 on ZsRE and 59.0 on Wiki) while simultaneously achieving the strongest RR.

Method	S ↑	ES ↑	PC ↑	RR ↑
ZsRE				
IKE	34.0	64.0	50.0	19.0
DR-IKE	33.0	**77.0**	65.0	16.0
MO-IKE	**42.0**	75.0	**79.0**	**22.0**
$\text{Wiki}_{\text{Counterfact}}$				
IKE	55.0	70.0	54.0	46.0
DR-IKE	53.0	**86.0**	**68.0**	33.0
MO-IKE	**59.0**	80.0	59.0	**47.0**

Table 2: Editing performance across the ZsRE and $\text{Wiki}_{\text{Counterfact}}$ datasets.

Zero-Shot Cross Model Evaluation. We observe that retriever trained via MO-IKE on one language model can effectively generalize to a completely different language model without any retraining. To prove this, we training the retriever purely on Llama-3.2, and evaluating it zero-shot on Mistral-7B. As demonstrated in Table 3, MO-IKE suffers no performance drop when transferred zero-shot to Mistral-7B. In fact, it performs practically identically to the natively trained version, outperforming all other baselines.

Method	S ↑	ES ↑	PC ↑	RR ↑
IKE	64.9	58.0	81.3	60.0
DR-IKE	62.3	74.7	88.3	42.3
MO-IKE (Native)	77.1	80.1	**90.7**	65.0
MO-IKE (Zero-Shot)	**77.3**	**81.0**	89.3	**65.7**

Table 3: Zero-shot transfer performance: the retriever was trained exclusively on Llama-3.2 and evaluated zero-shot on Mistral-7B.

5.3 Reward Constrained Ablation

We conduct an ablation study on the reward function to evaluate four configurations: (i) no constraints, (ii) paraphrase constraint only, (iii) retention constraint only, and (iv) all constraints. As shown in Table 4, omitting the retention penalty (row i and ii) leads to a noticeable drop in RR. This confirms that explicitly penalizing retention degradation is necessary to prevent editing success from negatively impacting the model's pre-existing knowledge. Furthermore, the results implies a positive correlation between edit success and paraphrase consistency: optimizing for editing reliability inherently benefits generality.

Reward Constraint	S↑	ES↑	PC↑	RR↑
No Constraint	65.5	91.3	77.0	50.0
Only Paraphrase	68.8	**92.0**	**79.0**	52.0
Only Retention	70.1	86.0	71.0	**57.7**
All Constraints	**73.5**	**92.0**	**79.0**	**57.7**

Table 4: Ablation study on the components of the MO-IKE scalarized reward function using Llama-3.2. **Bold** indicates the best performance.

It is worth noting that the full multi-objective formulation (all constraints) achieves better Score compared to only retention constraint (row iii and iv). We hypothesize that it may comes from the mechanics of group relative rewards. When the reward signal lacks multiple constraints, the group rewards

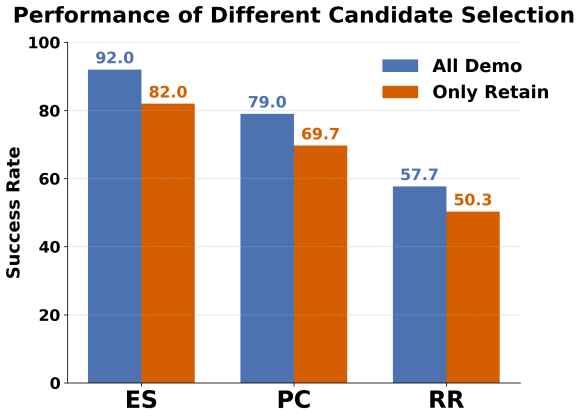

Figure 3: Performance of Llama-3.2 using different demonstration selection scopes.

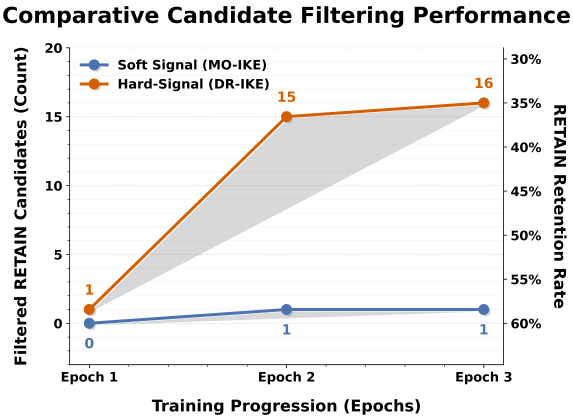

Figure 4: Left axis tracks filtered candidates; the right axis shows inverted retention degradation. The shaded gap highlights DR-IKE's hard constraint failing (over-filtering by Epoch 3) versus MO-IKE's soft embedding preserving stable retention.

tend to exhibit lower variance within a group. Consequently, the group-relative advantages become uninformative, which may downgrade the effect of the policy gradient updates. By incorporating all constraints, the reward landscape becomes more nuanced, providing the variance in advantage signals necessary for RL training.

In addition, we discuss reward weights sensitivity in Appendix F.

5.4 Action Space Ablation

Previous approaches to in-context knowledge editing have largely treated demonstration organization as a secondary concern. IKE (Zheng et al., 2023) constructs prompts using static format, essentially treating demonstrations as a fixed template. While DR-IKE (Nafee et al., 2025) introduces a dynamic retrieval mechanism, it restricts its optimization to the ranking and construction of only RETAIN candidates, ignoring the inter-dependencies between different demonstration types. In contrast, MO-IKE's action space is expanded to include all demonstrations across all categories. By conducting RL training across the entire sequence of demonstration types—COPY, UPDATE, and RETAIN—our method captures the global, synergistic effects of context construction. As illustrated in Figure 3, our approach achieves better performance across all three metrics compared to optimizing solely for RETAIN. These empirical results confirms our decision to expand the action space across all demonstration categories.

5.5 Demonstration Structure and Analysis

The performance of our approach can be attributed to the resulting structural proportions of the demonstration candidates. DR-IKE (Nafee et al., 2025) force a greedy optimization for edit success, which skews the prompt distribution by discarding RETAIN candidates. To resolve this structural imbalance, MO-IKE integrates a multi-objective RL framework and a soft stopping mechanism through a stop signal in the action space that is evaluated alongside other demonstration candidates. As shown in Figure 4, this allows the model to maintain contextual heterogeneity rather than aggressively filtering out non-edit demonstrations. Consequently, MO-IKE achieves better editing specificity overall (Table 1). We provide detailed Case Study of the context structure in Appendix H. By maintaining a balanced proportion of COPY, UPDATE, and RETAIN types, our method achieves robust results without sacrificing the overall integrity of the model's existing knowledge.

Conclusion

We introduce **MO-IKE**, a multi-objective RL algorithm for in-context knowledge editing that trains a retriever to dynamically construct prompts. We formulate demonstration retrieval as a Constrained MDP, casting prompt construction as a sequential decision-making problem. MO-IKE explicitly optimizes the retriever over the often competing objectives of reliability, generalization, and specificity. Experiments across multiple LLMs show that MO-IKE consistently outperforms existing retrieval strategies, achieving strong effectiveness across all three objectives and cross-model generalization. Ablation studies further demonstrate the importance of our constraint formulation, as well as the embedding and ranking designs, to MO-IKE's

overall performance.

Limitations

Several limitations still exist in our work. First, the dataset scale for knowledge editing remains relatively small; for example, we train and evaluate MO-IKE only on the first 2,000 records of the COUNTERFACT dataset, which may not be sufficient to examine the model's overall capabilities. Second, by formulating the problem as a Constrained MDP, we incorporate all constraints directly into the reward function. Because we utilize fixed coefficients (e.g., a static lambda value) for these constraints, the precise impact of these hyperparameters on the RL optimization process remains unexplored. Third, although MO-IKE achieves a better retention rate overall, its absolute performance is bottlenecked by the inherent limits of in-context learning. The retention rate is still relatively low (around 60%) compared to other metrics, suggesting this gradient-free method still has room for improvement. Future work may consider different approaches to surpass this limitation. Finally, we evaluate MO-IKE solely on a specific knowledge editing task, leaving it unclear whether this framework generalizes to other in-context learning applications, such as model reasoning.

Acknowledgements

This research was supported by the National Science Foundation under IIS-2348405 and the William & Mary Semester Research Grant.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Qizhou Chen, Dakan Wang, Taolin Zhang, Zaoming Yan, Chengsong You, Chengyu Wang, and Xiaofeng He. 2025. Uniedit: A unified knowledge editing benchmark for large language models. *Preprint*, arXiv:2505.12345.

Yuri Chervonyi, Trieu H. Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Junsu Kim, Vikas Verma, Quoc V. Le, and Thang Luong. 2025. Gold-medalist performance in solving olympiad geometry with alphageometry2. *Preprint*, arXiv:2502.03544.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yonathan Efroni, Ben Kretzu, Daniel Jiang, Jalaj Bhandari, Zheqing, Zhu, and Karen Ullrich. 2025a. Aligned multi objective optimization. *Preprint*, arXiv:2502.14096.

Yonathan Efroni, Ben Kretzu, Daniel Jiang, Jalaj Bhandari, Zheqing Zhu, and Karen Ullrich. 2025b. Aligned multi objective optimization. In *Forty-second International Conference on Machine Learning*.

Sourav Ganguly and Arnob Ghosh. 2025. Provably efficient sample complexity for robust cmdp. *Preprint*, arXiv:2511.07486.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 2023. Coverage-based example selection for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13924–13950, Singapore. Association for Computational Linguistics.

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Hao Peng, Julia Hockenmaier, and Tong Zhang. 2026. Tackling distractor documents in multi-hop qa with reinforcement and curriculum learning. In *Findings of the Association for Computational Linguistics: EACL 2026*, page 5548–5561. Association for Computational Linguistics.

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. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language*

Learning (CoNLL 2017), pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Fan Liu, Wen-Shuo Chao, Naiqiang Tan, and Hao Liu. 2025. Bag of tricks for inference-time computation of LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2024. A survey on knowledge editing of neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11759–11775.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372.

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

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

Mahmud Wasif Nafee, Maiqi Jiang, Haipeng Chen, and Yanfu Zhang. 2025. Dynamic retriever for in-context knowledge editing via policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16744–16757, Suzhou, China. Association for Computational Linguistics.

Kiran Purohit, Venktesh V, Sourangshu Bhattacharya, and Avishek Anand. 2025. Sample efficient demonstration selection for in-context learning. In *Forty-second International Conference on Machine Learning*.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. 2025. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *The Thirteenth International Conference on Learning Representations*.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. 2025. ToolRL: Reward is all tool learning needs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Shanbao Qiao, Xuebing Liu, and Seung-Hoon Na. 2024. COMEM: In-context retrieval-augmented mass-editing memory in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2333–2347, Mexico City, Mexico. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2017a. Trust region policy optimization. *Preprint*, arXiv:1502.05477.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017b. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Changyue Wang, Weihang Su, Qingyao Ai, Yichen Tang, and Yiqun Liu. 2025. Knowledge editing through chain-of-thought. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10673–10693, Suzhou, China. Association for Computational Linguistics.

Chenan Wang, Daniel H. Shi, and Haipeng Chen. 2026. Speculative sampling with reinforcement learning. *Preprint*, arXiv:2601.12212.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024a. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024b. Knowledge editing for large language models: A survey. *Preprint*, arXiv:2310.16218.

Yanzheng Xiang, Hanqi Yan, Lin Gui, and Yulan He. 2024. Addressing order sensitivity of in-context demonstration examples in causal language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6467–6481, Bangkok, Thailand. Association for Computational Linguistics.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *Preprint*, arXiv:2501.09686.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. 2026. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *Preprint*, arXiv:2508.19828.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Paul Youssef, Zhixue Zhao, Jörg Schlötterer, and Christin Seifert. 2025. How to make LLMs forget: On reversing in-context knowledge edits. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12656–12669, Albuquerque, New Mexico. Association for Computational Linguistics.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

A Training Procedure for Multi-objective GRPO

Algorithm 2 outlines the detailed MO-IKE algorithm.

Algorithm 2 Multi-Objective In-Context Knowledge Editing (MO-IKE)

Require: initial retriever parameters θ, training set $\mathcal{D}_{\text{train}}$, group size G, clipping ϵ, KL weight β_{KL}

for epoch $= 1$ to N_{epochs} **do**
  for each $(x, y_{\text{new}}) \in \mathcal{D}_{\text{train}}$ **do**
    Initialize storage lists: $\mathcal{D} \leftarrow []$, rewards $\leftarrow []$

    for $g = 1$ to G **do**
      $x_{\text{curr}} \leftarrow x$
      Sequence $R_g \leftarrow []$; LogProbs $\pi_g^{\text{old}} \leftarrow []$
      for step $j = 1$ to K **do** ▷ Sequentially select demonstrations
        $z \leftarrow S_\theta(x_{\text{curr}}, C)$; $p \leftarrow \text{softmax}(z)$
        Sample action $a_j \sim p$
        Append a_j to R_g; Append $p[a_j]$ to π_g^{old}
        Remove selected action a_t from candidate pool $\mathcal{C}_t$: $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_t \setminus \{a_t\}$
        $x_{\text{curr}} \leftarrow \text{Concat}(x_{\text{curr}}, a_j)$ ▷ Update prompt state
      end for
      Prompt LLM to get $\hat{y}_g$ and reward r_g ▷ Compute composite reward
      Append r_g to rewards
      Store trajectory data $(R_g, \pi_g^{\text{old}})$ in $\mathcal{D}$
    end for

    $\mu_{\text{group}} \leftarrow \frac{1}{G}\sum r_g$ ▷ Compute group baseline
    $\sigma_{\text{group}} \leftarrow \sqrt{\frac{1}{G}\sum(r_g - \mu_{\text{group}})^2 + \delta}$
    $A_g \leftarrow (r_g - \mu_{\text{group}})/\sigma_{\text{group}}$ for each group g

    for inner epoch $= 1$ to N'_{epochs} **do**
      $\mathcal{L}_{\text{total}} \leftarrow 0$
      for $g = 1$ to G **do**
        Reconstruct states $x_{g,0}, \ldots, x_{g,K}$ using x and R_g
        for step j in R_g **do**
          $p_{g,j}^{\text{new}} \leftarrow \text{softmax}(S_\theta(x_{g,j-1}, C))_{\text{action}_j}$
          $p_{g,j}^{\text{old}} \leftarrow \pi_g^{\text{old}}[j]$
          $ratio \leftarrow p_{g,j}^{\text{new}}/p_{g,j}^{\text{old}}$
          $L^{\text{CLIP}} \leftarrow -\min(ratio \cdot A_g, \text{clip}(ratio, 1-\epsilon, 1+\epsilon) \cdot A_g)$ ▷ Compute clipped surrogate objective
          $L^{\text{KL}} \leftarrow \beta_{\text{KL}} \cdot \log(p_{g,j}^{\text{new}}/p_{g,j}^{\text{old}})$ ▷ Approximate KL penalty
          $\mathcal{L}_{\text{total}} \mathrel{+}= (L^{\text{CLIP}} + L^{\text{KL}})$
        end for
      end for
      $\theta \leftarrow \theta - \alpha\nabla_\theta(\frac{1}{G \cdot K}\mathcal{L}_{\text{total}})$ ▷ Gradient descent update
    end for
  end for
end for

B Proof of the Final Reward via Lagrangian Relaxation

We model the problem as Constrained MDP. Our goal is to find a policy π_θ that maximizes the Edit Success (ES) while ensuring that the expected costs associated with violating Paraphrase Consistency (PC) and Retention Rate (RR) remain below acceptable thresholds.

B.1 Problem Formulation

Let $J_R(\pi)$ denote the expected return for the primary editing task based on the reward $R(s_t, a_t)$. We introduce cost constraints for PC and RR, where $J_{C_k}(\pi)$ represents the expected cumulative cost for constraint $k \in \{\text{PC}, \text{RR}\}$. Let d_k represent the maximum tolerable expected cost for that metric.

The CMDP optimization problem is defined as:

$$\begin{aligned} \max_{\pi_\theta} \quad & J_R(\pi_\theta) \\ \text{s.t.} \quad & J_{C_k}(\pi_\theta) \leq d_k, \quad \forall k \in \{\text{PC}, \text{RR}\} \end{aligned} \tag{9}$$

where the expectations are taken over the trajectory distribution induced by the policy π_θ.

B.2 Lagrangian Relaxation

We apply the method of Lagrange multipliers. We define the **Lagrangian objective function** $\mathcal{L}(\pi, \lambda)$ as:

$$\mathcal{L}(\pi, \lambda) = J_R(\pi) - \sum_{k \in \{\text{PC,RR}\}} \lambda_k (J_{C_k}(\pi) - d_k) \tag{10}$$

Here, $\lambda = (\lambda_{\text{PC}}, \lambda_{\text{RR}})$ is a vector of Lagrange multipliers, with $\lambda_k \geq 0$. The primal constrained problem is equivalent to the following unconstrained **min-max** dual problem:

$$\min_{\lambda \geq 0} \max_{\pi_\theta} \mathcal{L}(\pi, \lambda) \tag{11}$$

B.3 Equivalence to Scalarized Reward

We can rewrite the Lagrangian function by expanding the expected return and expected cost terms over the trajectories $\tau \sim \pi$. Since expectations are linear, we have:

$$\begin{aligned} \mathcal{L}(\pi, \lambda) &= \mathbb{E}\tau \sim \pi \left[\sum tR(s_t, a_t) \right] - \sum_{k \in \text{PC,RR}} \lambda_k \left(\mathbb{E}\tau \sim \pi \left[\sum tC_k(s_t, a_t) \right] - d_k \right) \\ &= \mathbb{E}\tau \sim \pi \left[\sum t \left(R(s_t, a_t) - \sum_{k \in \text{PC,RR}} \lambda_k C_k(s_t, a_t) \right) \right] + \sum_{k \in \text{PC,RR}} \lambda_k d_k \end{aligned} \tag{12}$$

Notice that the term $\sum_k \lambda_k d_k$ is constant with respect to the policy optimization step (the inner maximization loop over π). Therefore, maximizing $\mathcal{L}(\pi, \lambda)$ with respect to π is mathematically equivalent to maximizing the expected return of a new, **composite reward function**:

$$r(s_t, a_t) = R(s_t, a_t) - \sum_{k \in \{\text{PC,RR}\}} \lambda_k C_k(s_t, a_t) \tag{13}$$

Thus, by optimizing this composite reward using GRPO with fixed hyperparameters λ_{PC} and λ_{RR}, we are equivalently solving the inner loop of the Lagrangian dual problem. The fixed coefficients λ_k act as the Lagrange multipliers that apply static penalty weightings to balance the trade-off between editing performance and the cost constraints.

C Comparison to RL-Based RAG Retrieval

To test whether RL-based RAG retrievers can substitute for an IKE-specific approach, we evaluate RAG-RL (Huang et al., 2026) on CounterFact with Qwen-2.5-7B-Instruct. RAG-RL was originally trained to optimize answer-F1 and citation-F1 on open-domain QA; we apply it directly to the IKE setting without modification. We observe that the LLM frequently produces refusals ("I cannot determine the answer based on the prompt") when RAG-RL's retrieval forces it into contradiction with its parametric knowledge.

Methods	S ↑	ES ↑	PC ↑	RR ↑
RAG-RL	20.0	83.0	79.3	8.0
MO-IKE	**78.1**	**96.0**	**88.3**	**60.0**

Table 5: RAG-RL evaluated on CounterFact with Qwen-2.5-7B-Instruct. RAG-RL achieves moderate edit success but collapses on retention (8.0% RR), consistent with our claim in Section 2.

D Training Configuration

To ensure reproducibility, Table 6 details the complete set of hyperparameters used during the training of MO-IKE.

Hyperparameter	**Value**
Learning Rate	1×10^{-5}
Epochs	3
Group Size	8
Random Seed	42
GRPO Clipping Parameter (ϵ)	0.1
KL Coefficient (β)	0.001
Lagrangian Penalty Weights	1

Table 6: Hyperparameters used for MO-IKE training.

E Training Dynamics

To understand the learning efficiency of our RL-based retrieval optimization, we analyze the performance metrics of MO-IKE across training epochs. Figure 5 illustrates the progression of ES, PC and RR over the duration of the training process.

A key observation is the rapid convergence of the policy. By the conclusion of the first epoch, the model establishes a strong, balanced demonstration ordering, evidenced by the sharp initial peak in both ES and RR. Throughout subsequent epochs

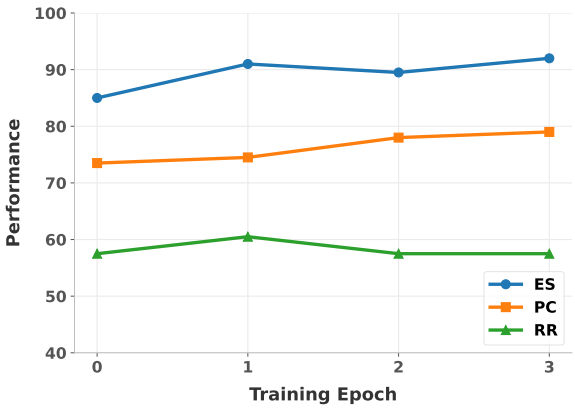

Figure 5: Training dynamics with Llama-3.2. We evaluate the performance of demonstration selection given different training epochs.

(Epoch 2 and 3), the performance largely plateaus; ES and PC demonstrate stable refinements, while RR maintains its rate without degrading. Because the reward includes a penalty for retention, policy updates are regularized toward smaller deviations, which can reduce unstable exploratory shifts during early RL training (Schulman et al., 2017a). The model rapidly identifies a near-optimal structural proportion for the demonstrations, proving that MO-IKE is not only effective at preventing structural collapse but also highly sample-efficient to train.

F Hyperparameter Sensitivity.

We conduct additional experiments to check hyperparameter sensitivity for knowledge editing reward weight λ_{ES} and constraint weights $\lambda_{PC}, \lambda_{RR}$. We swept λ_{PC} and λ_{RR} by $100\times$ each, holding the other fixed. Shown by Table 7, ESR varies by ≤ 2.0 points, PC by ≤ 3.3, RR is unchanged. MO-IKE still outperforms all baselines at every setting, with no monotonic degradation in any metric. The directional pattern matches the design: upweighting PC modestly improves PC (+1.0), while upweighting RR modestly degrades PC (-2.3) without further improving RR, indicating (1, 1) sits in a flat region balancing both objectives.

Configuration	**S ↑**	**ES ↑**	**PC ↑**	**RR ↑**
$\lambda_{PC} = 100, \lambda_{RR} = 1$	<u>73.2</u>	90.0	**80.0**	**57.7**
$\lambda_{PC} = 1, \lambda_{RR} = 100$	72.7	<u>91.7</u>	76.7	**57.7**
$\lambda_{PC} = 1, \lambda_{RR} = 1$	**73.4**	**92.0**	<u>79.0</u>	**57.7**

Table 7: Hyperparameter sensitivity sweep over the Lagrangian penalty weights.

G Additional Experiments

Evaluation on Large Datasets. We expand experiments with the UNIEDIT (Chen et al., 2025) dataset, a larger dataset composed of 311K editing examples drawn from a diverse mixture of relation types and knowledge domains, which is roughly a 15× increase in scale over COUNTERFACT. This setting provides a stricter test of whether MO-IKE's gains generalize beyond the testing of a small benchmark. As indicated by Table 8, MO-IKE achieves the best Score and best RR. FactPrompt's high ESR (81.7%) is misleading: its Score (15.0%), PC (22.7%), and RR (7%) are the lowest in the table, indicating FactPrompt does not let the model effectively learn factual updates — the model parrots the injected editing prompt rather than internalizing new knowledge.

Method	S ↑	ES ↑	PC ↑	RR ↑
FactPrompt	15.0	**81.7**	22.7	7.0
EditCoT	<u>21.5</u>	43.0	**36.0**	<u>11.0</u>
IKE	19.6	39.3	27.6	<u>11.0</u>
DR-IKE	17.2	38.0	27.3	9.0
MO-IKE	**24.5**	<u>51.3</u>	<u>28.3</u>	**14.3**

Table 8: Evaluation on the UniEdit dataset using Llama-3.2-3B-Instruct.

LLM Performance. Table 9 presents the performance of MO-IKE across various frozen LLMs. We observe a general positive scaling trend, where performance of Score (S) improves as model size increases. The highest overall rates are achieved by models with larger parameter scales ($\geq$ 7B).

Model (Parameters)	S↑	ES↑	PC↑	RR↑
Qwen 2.5 (7B)	75.0	**96.0**	88.3	60.0
Qwen 2.5 (1.5B)	63.7	69.7	67.7	54.7
Llama 3.1 (8B)	76.7	88.0	77.7	**67.3**
Llama 3.2 (3B)	73.5	92.0	79.0	57.7
Mistral v0.3 (7B)	**77.4**	80.1	**90.7**	65.0

Table 9: Editing performance across different LLMs.

Natural Baselines. We provide natural baselines that showcase the performance of the retriever, as we claim that selection and ordering and demonstrations matter. Precisely, we compare MO-IKE with random shuffling baseline and a duplication baseline where the target edit is repeated multiple times, while keeping UPDATE and RETAIN unchanged. Table10 shows that simple duplication or shuffling yields moderate performance but does not match the balance of ES/PC/RR achieved by our multi-objective RL framework.

Method	ES ↑	PC ↑	RR ↑
Random Shuffling	77.0–91.0	70.0–74.0	52.0–60.0
Duplicate Target 2×	79.3	72.3	51.0
Duplicate Target 4×	76.7	73.7	50.7
Duplicate Target 8×	78.3	<u>77.0</u>	<u>52.0</u>
Duplicate COPY 2×	78.3	72.7	<u>52.0</u>
Duplicate COPY 4×	79.0	73.7	49.7
Duplicate COPY 8×	<u>79.7</u>	70.0	51.0
MO-IKE	**96.0**	**88.3**	**60.0**

Table 10: Performance of heuristic prompt engineering strategies on COUNTERFACT.

Evaluation on 2025-era-SOTA LLMs. To directly test the long-term applicability of MO-IKE, we conducted an additional evaluation on a recently released 2025-era model, Qwen3-4B-Instruct. Results are summarized in Table 11. We observe that base models exhibit high baseline robustness—for instance, standard static IKE achieves an impressive 98.67% Edit Success Rate (ESR) and 94.67% Paraphrase Consistency (PC) on Qwen-3. Since ESR and PC are near the ceiling in this model, RR becomes the key metric in differentiating MO-IKE from all other baselines. We can observe that MO-IKE exceeds the single objective baseline DR-IKE by approximately 23% on RR, confirming its effectiveness in preserving neighboring facts in knowledge editing. Moreover, MO-IKE successfully preserves the high ESR and PC of the newer models. Therefore, in terms of the overall performance, MO-IKE achieves the highest overall score (S) of 70.2%.

Method	S ↑	ES ↑	PC ↑	RR ↑
FactPrompt	50.4	91.0	75.3	31.3
EditCoT	42.1	86.1	76.5	24.6
IKE	<u>65.6</u>	<u>98.7</u>	<u>94.7</u>	<u>46.3</u>
DR-IKE	49.6	**99.3**	**96.0**	30.0
MO-IKE	**70.2**	98.3	94.3	**53.0**

Table 11: Editing performance on Qwen3-4B-Instruct. While stronger base models exhibit high baseline robustness, MO-IKE remains the only framework capable of preserving knowledge specificity (RR), achieving the highest overall score.

Computation Cost. We evaluate inference latency Llama-3.2-3B-Instruct. The results are detailed in Table 12.

Component	Mean (ms)	Median (ms)	Std (ms)	p95 (ms)
Retrieval	812.2	808.5	36.2	871.8
Generation	112.3	112.8	45.6	181.3
End-to-End	924.4	915.4	55.4	1027.3

Table 12: Inference latency per query evaluated on COUNTERFACT using Llama-3.2-3B-Instruct. Benchmarked on a single NVIDIA A100-SXM4-40GB GPU at batch size 1.

H Case Study

Table 13 use the record 676 in COUNTERFACT to illustrate the relationship between the structural composition of retrieved demonstrations and the resulting performance tradeoffs in different in-context knowledge editing frameworks. We use edit success to represent Reliability Metric, and retention rate to represent Specificity Metric.

IKE. The baseline IKE framework relies on fixed prompt construction—statically ordering COPY, UPDATE, and RETAIN demonstrations. As shown in the Reliability Metric, this static template approach fails to successfully execute the target edit ($y_{true} \rightarrow y_{new}$). Because the retriever does not optimize for the edit signal, the model defaults to its pre-trained knowledge (Russian) rather than adopting the new fact.

DR-IKE. While DR-IKE successfully forces the edit (outputting "French" for Ivanov-Vano), it suffers from a "over-optimization" issue. By penalizing only on non-edit-maximizing signals, the policy optimization entirely discards RETAIN candidates. This creates an unbalanced prompt heavily skewed toward the target concept. Consequently, DR-IKE exhibits fact forgetting, failing the Specificity metric by incorrectly altering the language of a related but distinct entity (Vladimir Smirnov) to French.

MO-IKE. MO-IKE resolves the tradeoff between edit success and knowledge retention. By avoiding rigid templates and maintaining a balanced structural proportion of demonstration types, MO-IKE allows the retriever to dynamically interleave UPDATE, RETAIN, and COPY examples. This unfixed ordering provides sufficient contextual protection for unrelated facts while still providing a strong enough edit signal to update the target knowledge. As a result, MO-IKE is the only framework in the case study that successfully achieves both Reliability (updating Ivanov-Vano to French) and Specificity (retaining Russian for Smirnov).

Component	IKE	DR-IKE	MO-IKE
Edit Request	*Target Edit:* Ivan Ivanov-Vano spoke the language **Russian** (y_{true}) → **French** (y_{new})		
Selected Demonstrations	**[COPY]** Andrey Malakhov spoke the language French. **[COPY]** Boris Shaposhnikov spoke the language French. ... **[UPDATE]** Andrey Malakhov is a native speaker of French. **[UPDATE]** Fyodor Pavlovich Reshetnikov, a native French. ... **[RETAIN]** *Roger Vitrac spoke the language Russian.* **[RETAIN]** *Christophe Moreau spoke the language Russian.* ... *(Note: **Fixed category order.**)*	**[COPY]** Andrey Malakhov spoke the language French. **[COPY]** Boris Shaposhnikov spoke the language French. ... **[UPDATE]** Andrey Malakhov is a native speaker of French. **[UPDATE]** Fyodor Pavlovich Reshetnikov, a native French. ... *(Note: **Fixed COPY and UPDATE order.** No RETAIN candidates, as they are filtered due to penalty on non-edit-maximizing signals)*	**[COPY]** Andrey Malakhov spoke the language French. **[RETAIN]** *Christophe Moreau spoke the language Russian.* **[UPDATE]** Fyodor Pavlovich Reshetnikov, a native French. ... **[UPDATE]** Andrey Malakhov is a native speaker of French. ... **[RETAIN]** *Roger Vitrac spoke the language Russian.* **[COPY]** Boris Shaposhnikov spoke the language French. *(Note: **Unfixed order.** Balanced structural proportion maintained via soft embedding)*
Reliability Metric *Ivan Ivanov-Vano spoke the language?*	Russian. (*Unsuccessful edit due to heuristic prompt construction*)	French. (*Success*)	French. (*Success*)
Specificity Metric *Vladimir Smirnov, speaker of?*	Russian. (*Success*)	French. (*Unsuccessful retention due to unbalanced RETAIN demonstrations*)	Russian. (*Success*)

Table 13: Comparison of in-context editing frameworks highlighting the structure of the constructed prompt.