

Beyond Paraphrases: Generalization Across Task Formats in Parametric Knowledge Editing

Cheongwoong Kang^{1,2}, Jongeun Baek¹, Yeonjea Kim¹, Jaesik Choi^{1,2}

¹KAIST, ²INEEJI

{cw.kang, jongeun.baek, yeon.kim, jaesik.choi}@kaist.ac.kr

Abstract

Knowledge editing aims to update specific facts in language models to ensure reliable and up-to-date behavior. A key challenge is whether edited knowledge generalizes beyond the format in which the edit is applied, yet existing evaluations primarily assess generalization using paraphrases. In this work, we identify a critical blind spot: generalization across heterogeneous task formats, including multiple-choice and verification-based question answering. We show that paraphrase-based evaluations overestimate robustness, as models that perform well on paraphrases exhibit substantial performance drops across task formats. While model predictions are consistent across formats prior to editing, editing introduces format-dependent inconsistencies. We further observe that heterogeneous task formats exhibit distinct decision landscapes, where single-format objectives lead to format-specific local optima with divergent edit directions, resulting in misaligned outputs across formats. To address this issue, we demonstrate that aligning edit directions across task formats improves cross-format generalization and consistency. These findings highlight the need to move beyond paraphrase-based evaluation and to develop editing methods that maintain consistent factual behavior across diverse task formulations.

1 Introduction

Knowledge editing aims to update specific facts stored in language models (Meng et al., 2022, 2023; Fang et al., 2025; Deng et al., 2025). As these methods become increasingly important for correcting factual errors and maintaining up-to-date information, a key question is whether edited knowledge generalizes beyond the format in which the edit is applied, ensuring consistent behavior across di-

verse usage scenarios. However, existing evaluations primarily rely on paraphrase-based variations, overlooking broader forms of variation. For example, a policy update that appears correct in free-form responses may yield inconsistent judgments in structured compliance checks (e.g., true/false), potentially leading to downstream workflow failures and audit risks.

In this work, we study generalization across heterogeneous task formats (e.g., multiple-choice and true/false question answering), an underexplored setting in knowledge editing. We show that editing efficacy drops substantially across task formats, despite strong performance on paraphrase-based evaluations.

While model predictions are largely consistent across formats prior to editing, this consistency breaks down after editing: knowledge edits that succeed in the source format often fail to transfer to others, resulting in inconsistent predictions. To explain this behavior, we analyze editing dynamics at the representation level. We observe that heterogeneous task formats induce distinct decision landscapes, where single-format objectives lead to local optima, preventing the model from learning compatible updates that generalize across task formats. As a result, edit directions diverge across formats, leading to misaligned model behavior.

Motivated by this insight, we investigate aligning edit directions across task formats during the editing process. This approach improves cross-format generalization and consistency while preserving performance on the original task interface, highlighting the importance of explicitly accounting for alignment across heterogeneous task formats in knowledge editing.

Our main contributions are as follows:

- We identify a critical blind spot in existing evaluation protocols in knowledge editing, showing that paraphrase-based evaluations

*Code and data are available at <https://github.com/cheongwoong/mf-memit>.

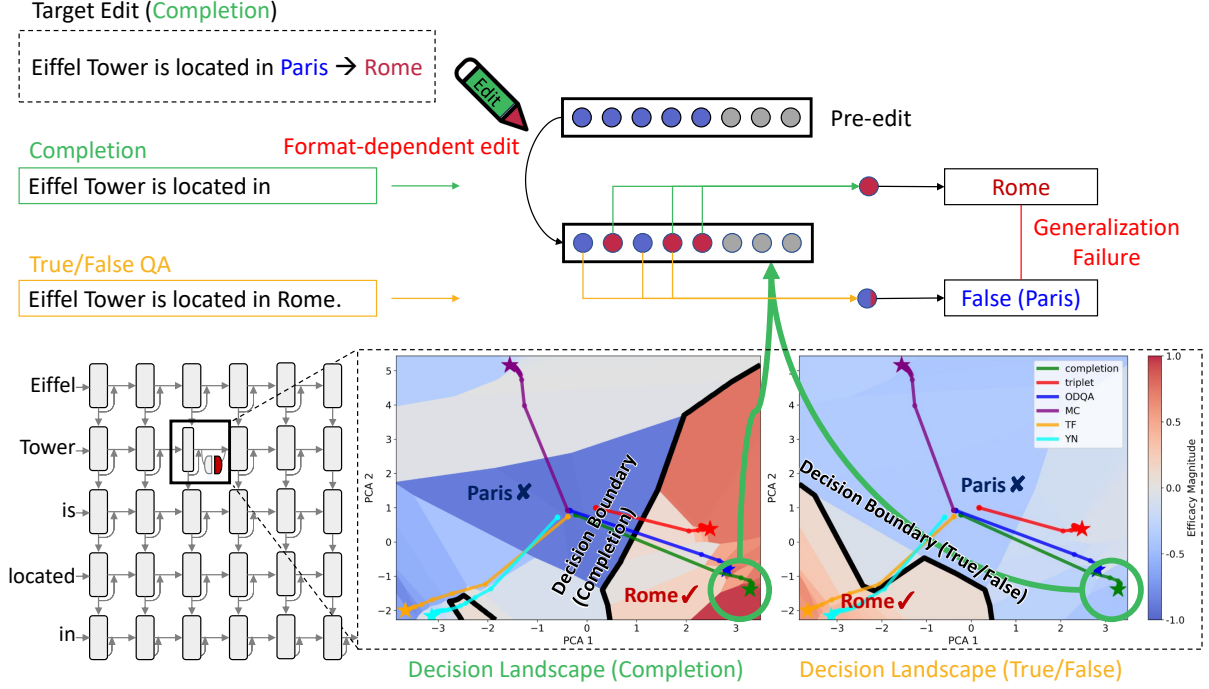


Figure 1: **Format-dependent editing dynamics.** Representations that yield successful edits in one format may fail to transfer to others due to differences in format-specific decision landscapes (background heatmaps; red: high efficacy, blue: low efficacy). These landscape differences guide optimization toward different local optima under each format, leading to divergent optimization trajectories and in turn, format-dependent inconsistent outcomes. For the same factual edit (*Eiffel Tower*, *location*, *Paris* $\Rightarrow$ *Rome*), we visualize MEMIT’s value optimization trajectories from the pre-edit state v^0 to the optimized target v^* under each format. While some formats (e.g., TF and YN) exhibit partially aligned trajectories, heterogeneous interfaces often follow more distinct yet structured paths.

substantially overestimate robustness across heterogeneous task formats.

- We demonstrate that knowledge editing induces format-dependent updates, leading to inconsistent factual behavior across task formats despite successful edits in the source format.
- We identify divergent edit directions as a key mediating factor linking format-specific local optima to cross-format inconsistency.
- We show that explicitly aligning edit directions across heterogeneous task formats improves cross-format generalization and consistency.

2 Background on Model Editing

Following prior work (Meng et al., 2022, 2023), we view W_{proj}^l as a linear associative memory (Kohonen, 1972; Anderson, 1972), where W_{proj}^l is the MLP projection matrix at layer l . This perspective suggests that any linear operation W can associate

a set of vector keys $K = [k_1 \mid k_2 \mid \dots]$ (encoding subject entities) and corresponding values $V = [v_1 \mid v_2 \mid \dots]$ (encoding attributes) by minimizing the squared error $\|\hat{W}K - V\|$. A set of new associations (K_1, V_1) can then be inserted by solving a constrained least-squares problem, leading to the closed-form update:

$$\hat{W} = W_0 + RK_1^\top (C_0 + K_1K_1^\top)^{-1}, \quad (1)$$

where K_1 contains the keys for edited subjects, $R \triangleq V_1 - W_0K_1$ stacks the residuals of their desired value shifts, and $C_0 = \lambda \cdot \mathbb{E}_k[kk^\top]$ is the pre-computed covariance statistic of original keys sampled from text, scaled by a hyperparameter λ that regularizes the weighting of new vs. existing associations.

The editing procedure requires selecting appropriate key and value vectors to insert new associations. To obtain a key vector k^* , we sample multiple prefix contexts ending with the subject token and average their MLP input representations. The target value v^* (MLP output) is then optimized such that inserting (k^*, v^*) shifts the model’s prediction from the original object o^c to the target

object o^* . The resulting (k^*, v^*) pairs are directly inserted into the projection matrix update in Equation 1.

3 Methodology

3.1 Evaluating Knowledge Editing Across Heterogeneous Task Formats

Existing evaluations of knowledge editing primarily assess generalization under paraphrase-level variations (Meng et al., 2022; Wang et al., 2024b; Zhang et al., 2024). While paraphrases introduce surface-level diversity, they preserve the underlying task interface. In practice, however, language models are queried through a variety of formats beyond simple completion, including open-ended question answering and structured decision tasks such as multiple-choice and true/false questions. As a result, prior evaluations may overestimate the robustness of edits by overlooking inconsistencies that arise under task format shifts. To address this limitation, we evaluate knowledge editing across heterogeneous task formats with distinct decision interfaces.

We define a knowledge edit as a tuple $(s, r, o^c \Rightarrow o^*)$, where s is a subject entity, r is a relation, and the original object o^c is updated to the target object o^* . To assess cross-format generalization, we instantiate this edit across multiple task formats that are semantically equivalent, i.e., each format queries the same underlying fact (s, r, o) while varying the form of interaction. We distinguish between a *source format* f_{src} , in which the edit is applied, and a set of *target formats* $\mathcal{F}_{\text{tgt}} = \{f_1, \dots, f_k\}$, on which the edit is evaluated. Cross-format generalization is defined as the extent to which an edit learned in f_{src} transfers to formats in $\mathcal{F}_{\text{tgt}}$.

Format Variation Design. We consider six task formats that differ in input-output structure and decision interfaces. **Completion** prompts the model to generate a free-form continuation. **Triplet** presents a structured $(s, r, \cdot)$ schema with an explicit relational slot. **Open-Domain QA (ODQA)** uses interrogative question templates. **Multiple-Choice QA (MC)** maps candidate objects to discrete options. **True/False QA (TF)** frames the task as binary statement verification. **Yes/No QA (YN)** queries the association between s and a candidate object in a binary question format. These formats span both generative settings (Completion, Triplet, ODQA) and constrained decision settings

(MC, TF, YN), introducing variation not only in surface form but also in how the model expresses or selects answers. Detailed templates are provided in Appendix A.

Metrics. We evaluate edits using three standard metrics from prior work (Meng et al., 2022). **Efficacy** measures whether the edited model assigns a higher probability to the target than to the original object for the edited fact: $\mathbb{1}[\mathbb{P}[o^* | s, r] > \mathbb{P}[o^c | s, r]]$. **Specificity** measures whether unrelated knowledge is preserved. For neighboring subjects s_n that share the original object o^c under relation r , we verify that their facts remain unchanged by checking whether the original object o^c is still assigned a higher probability than the target object o : $\mathbb{1}[\mathbb{P}[o^c | s_n, r] > \mathbb{P}[o^* | s_n, r]]$ for neighborhood subjects s_n sharing the original object o^c . **Generalization** measures whether an edit transfers across prompt formats, defined as the average efficacy over $\mathcal{F}_{\text{tgt}}$. To further capture cross-format behavior, we introduce **Consistency**, which measures the agreement of model predictions across formats. We define consistency as the average pairwise agreement of predictions across all format pairs for a given instance (Elazar et al., 2021). Together, these metrics provide a comprehensive evaluation of knowledge editing, capturing not only local edit success and side effects, but also the extent to which edits transfer and remain consistent across heterogeneous task formats.

3.2 Aligning Edit Directions Across Task Formats

As illustrated in Figure 1, updates derived from a single format may converge to local optima that do not generalize across task formats, as each format induces a distinct decision landscape. Such updates steer representations toward format-specific regions, resulting in inconsistent behavior across formats. This motivates the need for editing strategies that align edit directions across task formats.

To this end, we optimize edit directions that are jointly supported across multiple formats during the editing process. A straightforward extension of single-format editing is to optimize independent key-value updates for each format (*multi-key + multi-value*), yielding multiple format-specific edit directions. However, independently optimized updates may conflict with one another, particularly when key representations are shared across formats (Dong et al., 2025). Instead, we adopt a *multi-key*

+ *joint-value* formulation, where a single shared value vector is optimized across formats. For a candidate value v , let P_v denote the model output distribution obtained by intervening on the target MLP output with v during the forward pass. For a target edit $(s, r, o^c \rightarrow o^*)$ and a set of source formats $\mathcal{F}_{\text{src}}$, we optimize:

$$v^* = \arg \min_v \sum_{f \in \mathcal{F}_{\text{src}}} -\log P_v[o_f^* | s, r, f], \quad (2)$$

which encourages a unified edit direction that is jointly supported across formats, promoting consistent factual behavior.

Iterative Expansion of Source Formats. While aligning edit directions across formats improves consistency, jointly optimizing across all formats can be computationally inefficient and may introduce redundant constraints. To address this, we construct a minimal sufficient set of source formats via an iterative expansion strategy. This procedure is performed once at a dataset-level to identify the subset, after which all edits use the selected formats without additional overhead.

Starting from an initial set $\mathcal{F}_{\text{src}} = \{\text{Completion}\}$, we iteratively expand it based on evaluation performance. At each iteration, we: (1) apply editing using the current $\mathcal{F}_{\text{src}}$; (2) evaluate the edited model on all candidate formats $\mathcal{F}_{\text{cand}}$; (3) identify the weakest format based on lowest efficacy; (4) add this format to $\mathcal{F}_{\text{src}}$ if its performance falls below a threshold τ . The procedure terminates when all candidate formats achieve efficacy above τ , indicating that the selected subset sufficiently covers the considered formats. This approach reduces computational cost while revealing whether a small subset of formats is sufficient to induce consistent factual update across formats.

4 Experiments

4.1 Experimental Setup

Editing Methods. We focus on parametric editing methods, which directly modify model parameters and enable analysis of internal update mechanisms. We compare four editing methods alongside an unedited baseline. **Baseline** refers to the original model without editing. **ROME** (Meng et al., 2022) applies localized updates to MLP layers at subject token positions. **MEMIT** (Meng et al., 2023) extends ROME by distributing updates across multiple MLP layers to support batch

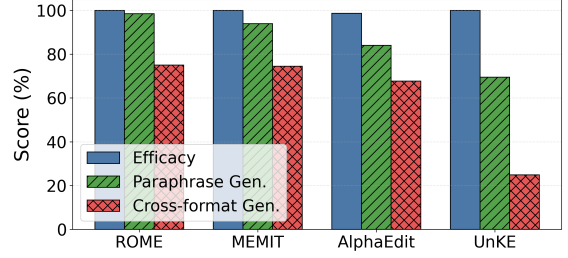


Figure 2: **Paraphrase vs. cross-format generalization gap.** All methods achieve near-perfect efficacy and strong paraphrase-based generalization. However, cross-format generalization drops substantially across all methods, revealing that paraphrase-based evaluations can significantly overestimate robustness.

editing. **AlphaEdit** (Fang et al., 2025) further constrains MEMIT by penalizing deviations from the original model distribution. **UnKE** (Deng et al., 2025) performs more global updates without restricting edits to specific MLP layers or subject token positions. Implementation details are provided in Appendix B.

Datasets. We evaluate on **CounterFact-1k**, a subset of 1,000 triples from CounterFact (Meng et al., 2022). We additionally use three datasets from KnowEdit (Zhang et al., 2024), including **Wiki-CounterFact**, **Wiki-Recent** and **ZsRE**, to assess generalization across domains and evaluation settings. For each fact, we construct instances across the six formats introduced in Section 3.1, enabling systematic cross-format evaluation.

Models. Our primary experiments use **Llama-3.2-3B-Instruct** (Dubey et al., 2024). To evaluate generality, we additionally consider models of varying scales and architectures, including **Llama-3.2-1B-Instruct**, **Llama-3.1-8B-Instruct**, **OLMo2-Instruct** (1B, 7B, 13B) (Walsh et al., 2025) and **Qwen3** (0.6B, 1.7B, 4B, 8B, 14B) (Yang et al., 2025).

4.2 Limits of Cross-Format Generalization

We evaluate whether knowledge edits performed in the *completion* format generalize to alternative task formats. Figure 2 shows results on CounterFact-1k with Llama-3.2-3B-Instruct, comparing paraphrase-based and cross-format evaluation. All methods achieve near-perfect efficacy in the source format ($> 99\%$), confirming that the edited knowledge is successfully retrieved under the completion setting. Paraphrase-based generalization is also

high (86.5% on average), indicating strong transfer across surface-level variations within the same task interface. In contrast, performance drops sharply under cross-format evaluation (60.5% on average), demonstrating that successful edits in one format do not reliably transfer to others. This degradation is more pronounced for constrained formats such as true/false QA (43.6%), while remaining relatively higher for generative formats such as ODQA (78.6%). Detailed results are provided in Appendix C. These results reveal a key limitation of conventional evaluations: paraphrase-based metrics substantially overestimate robustness by failing to capture inconsistencies across task formats. Edits that appear robust under paraphrased inputs often produce inconsistent predictions under alternative task interfaces.

Cross-Format Generalization Across Models and Datasets. We next examine whether this gap persists across models and datasets. Across models from the Llama, OLMo2 and Qwen3 families (0.6B-14B parameters), using MEMIT on CounterFact-1k, we observe consistent gaps between paraphrase-based and cross-format evaluation, averaging 14.9 percentage points (pp). We further evaluate on additional datasets from KnowEdit (Wiki-CounterFact, Wiki-Recent and ZsRE) using Llama-3.2-3B-Instruct, observing the same pattern: high paraphrase generalization but substantially lower cross-format performance. These findings indicate that format-dependent failures are not specific to a particular model or dataset, but reflect a broader limitation of current knowledge editing methods. Detailed results are provided in Appendix C.

Structured Transfer Across Task Formats. To better understand cross-format generalization failures, we analyze transfer patterns between editing and evaluation formats. Figure 3 shows efficacy when MEMIT edits are applied in one format and evaluated in another on CounterFact-1k with Llama-3.2-3B-Instruct. Transfer patterns exhibit clear structure across task formats. Edits applied in open-generation formats transfer relatively well to other generative settings, whereas constrained interfaces such as MC, TF and YN exhibit substantially weaker transfer to generative formats. Notably, TF and YN remain highly compatible with one another despite poor transfer elsewhere, while MC behaves as a partially separate interface. These results suggest that cross-format generalization is structured

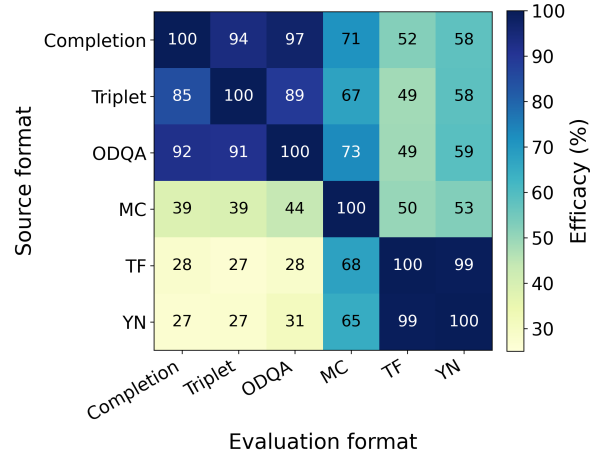


Figure 3: **Single-format edit transfer across task formats.** Transfer patterns cluster by task interface: open-generation formats transfer strongly to one another, TF and YN form a tightly aligned binary cluster, while MC remains only partially compatible with either group.

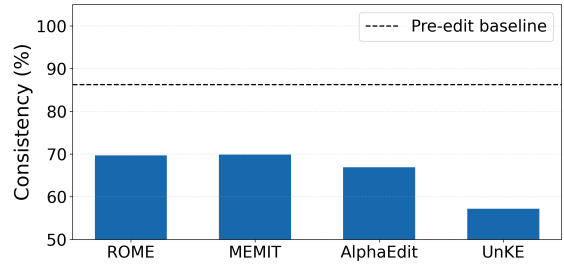


Figure 4: **Pre- vs. post-edit consistency across editing methods.** The dashed line indicates pre-edit consistency, while bars represent post-edit consistency after applying each editing method. These results show that knowledge editing systematically reduces cross-format consistency, indicating that edits introduce format-specific behavior.

by task-interface similarity rather than governed by a single shared format-invariant behavior space.

4.3 Cross-Format Inconsistency from Format-Dependent Editing Behavior

We examine how knowledge editing affects the consistency of model predictions across task formats. Figure 4 compares pre- and post-edit consistency on CounterFact-1k with Llama-3.2-3B-Instruct. Before editing, the model exhibits high cross-format consistency (86%), indicating stable predictions across formats. However, all editing methods substantially reduce consistency, with scores dropping to around 70% for ROME and MEMIT, and further to 57% for UnKE. This result shows that cross-format inconsistencies are not an inherent property of the model, but are introduced

by the editing process. While the model produces coherent predictions across formats prior to editing, targeted updates induce format-specific behavior, leading to divergent predictions for semantically equivalent queries. These findings indicate that existing knowledge editing methods produce updates that are locally effective but globally inconsistent across task formats.

4.4 Divergent Edit Directions Across Task Formats

To explain the limited cross-format generalization and the inconsistencies observed across task formats, we analyze editing dynamics at the representation level. We study how knowledge updates differ across formats by examining the optimization trajectories of MEMIT¹ on Llama-3.2-3B-Instruct. Each trajectory traces the path from the initial value vector v_f^0 to the final edited vector v_f^* for a given format f . Figure 1 visualizes these trajectories in a two-dimensional PCA projection (Pearson, 1901). Edit trajectories derived from different formats follow distinct directions in representation space: formats with similar decision interfaces (e.g., True/False and Yes/No) exhibit partially aligned trajectories, while others diverge substantially. Crucially, the underlying decision landscapes differ across formats, such that updates derived from a single format converge toward format-specific local optima, failing to produce a shared update direction across formats. As a result, edit directions diverge across formats, which can lead to misaligned outputs. These observations suggest that divergence in edit directions serves as a mechanism linking format-specific decision landscapes to output inconsistency.

Edit Direction Divergence and Output Inconsistency. To test whether edit direction divergence drives output inconsistency, we isolate the effect of edit direction under controlled conditions. For a fixed anchor format, we construct edits using directions derived from different source formats and evaluate all resulting models under the same anchor format, holding inputs and model state constant. Divergence is measured as the cosine distance between their edit directions, and inconsistency is defined as disagreement in the resulting predictions. We find that larger divergence is associated with higher inconsistency. While raw correlations

¹This value optimization step is shared across ROME-based editing methods, including ROME and AlphaEdit.

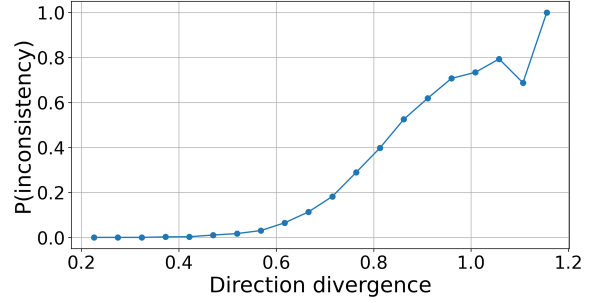


Figure 5: **Edit direction divergence predicts output inconsistency.** Each point shows the probability of inconsistent outputs within bins of edit direction divergence. The strong monotonic trend indicates that larger divergence leads to a higher likelihood of inconsistent predictions across task formats.

are moderate (Spearman’s $\rho = 0.46$, $p < 0.001$), a binned analysis reveals a strong monotonic relationship (Spearman’s $\rho = 0.99$, $p < 0.001$; Figure 5), indicating that the probability of inconsistent outputs increases with divergence.

We further perform intervention-style comparisons between maximally divergent directions and anchor-consistent directions. Averaged across all instances, maximal divergence results in inconsistent outputs in 60.1% of cases, demonstrating that edit direction divergence is a key mediating factor driving output inconsistency. These results provide a mechanistic explanation for the observed generalization failures: format-specific decision landscapes induce divergent edit directions through local optima, and this divergence in turn leads to misaligned outputs across formats. This motivates the need to align edit directions across task formats.

4.5 Aligning Edit Directions Improves Cross-Format Consistency

We evaluate whether aligning edit directions across task formats addresses the limitations identified in previous sections, namely limited cross-format generalization and consistency.

Figure 6 compares MEMIT with its paraphrase-aligned variant (MEMIT-Para) and cross-format aligned variant (MEMIT-XF) on CounterFact-1k using Llama-3.2-3B-Instruct. MEMIT achieves strong paraphrase generalization (94.0), but cross-format generalization remains limited (74.5). MEMIT-Para improves paraphrase generalization, but does not improve cross-format generalization. In contrast, MEMIT-XF maintains strong paraphrase performance (96.0) while substantially improving cross-format generalization (99.8). These

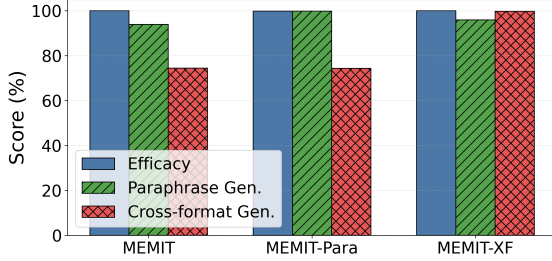


Figure 6: **Effect of aligning edit directions across task formats.** Comparison among MEMIT, its paraphrase-aligned variant (MEMIT-Para) and its cross-format aligned variant (MEMIT-XF). Aligning edit directions across task formats substantially improves cross-format generalization, while paraphrase alignment alone is insufficient.

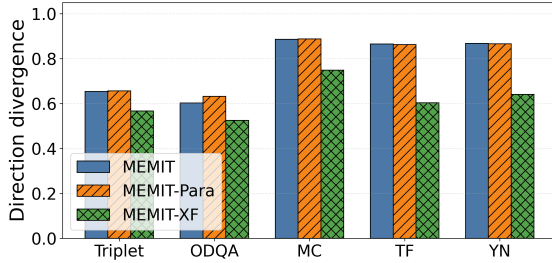


Figure 7: **Cross-format alignment reduces edit direction divergence.** Bars show cosine distance between each method’s edit direction and the single-format MEMIT direction. MEMIT-Para produces divergence patterns similar to MEMIT, whereas MEMIT-XF substantially reduces divergence across formats, providing direct evidence that cross-format alignment reduces representational divergence.

results show that aligning edit directions across heterogeneous task formats produces updates that transfer consistently across formats, substantially reducing the gap between paraphrase-based and cross-format performance while preserving performance on the original interface.

To verify whether the consistency improvements of MEMIT-XF arise from reduced edit direction divergence, we analyze how edit direction divergence changes under different alignment strategies. Figure 7 compares cosine distances between edit directions produced by each strategy and the corresponding single-format MEMIT direction across formats. MEMIT-Para produces divergence patterns similar to MEMIT, indicating that paraphrase alignment alone does not substantially alter the geometry of edit directions across task formats. In contrast, MEMIT-XF consistently reduces edit di-

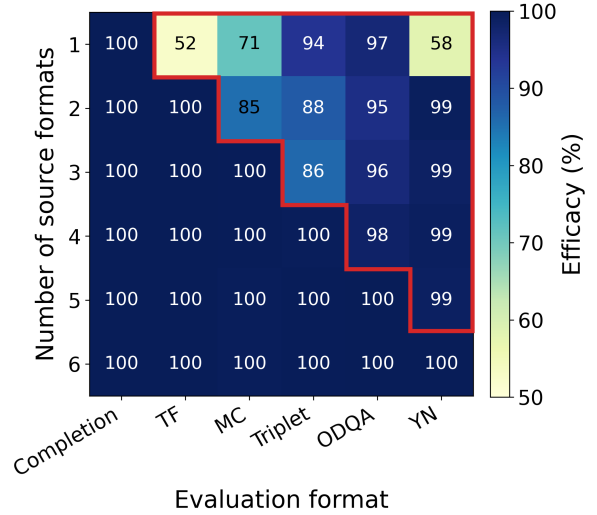


Figure 8: **Cross-format efficacy as a function of the number of source formats (K).** Columns follow the order in which formats are added as edit sources, and the red outline marks formats held out from editing at each K . Held-out performance improves unevenly across formats: adding TF strongly improves YN, suggesting strong transfer between similar verification-oriented interfaces, while also improving MC. Incorporating MC and Triplet further improves coverage, yielding near-saturated performance by $K = 4$.

rection divergence across formats. These results provide direct evidence that cross-format alignment improves cross-format consistency by reducing representational divergence between format-specific edit directions.

Effect of Alignment Components. We analyze the contribution of each component in the alignment objective. Removing value alignment across formats (*w/o multi-value*) leads to a substantial drop in cross-format generalization ($99.8 \rightarrow 75.1$), effectively reverting to single-format behavior. Removing key alignment (*w/o multi-key*) has minimal impact. Removing the joint optimization objective (*w/o joint-loss*) preserves cross-format performance but reduces paraphrase generalization ($96.0 \rightarrow 89.5$), indicating weaker generalization beyond supervised formats. These results suggest that value alignment across formats, together with joint optimization, is critical for achieving consistent updates.

Effect of Number and Type of Source Formats. We further analyze how cross-format generalization changes as the source formats are iteratively expanded. Figure 8 presents the efficacy on each task format when the edit direction is aligned using K

# Edits	Method	Cross-Format Gen.	Cross-Format Cons.	Specificity
10	MEMIT	73.7	69.2	83.3
10	MEMIT-XF ($K = 4$)	99.4	99.1	83.6
10	MEMIT-XF ($K = 6$)	99.9	99.8	83.5
100	MEMIT	72.6	68.2	83.3
100	MEMIT-XF ($K = 4$)	99.4	99.1	83.8
100	MEMIT-XF ($K = 6$)	99.9	99.8	83.5
1,000	MEMIT	70.1	65.8	83.0
1,000	MEMIT-XF ($K = 4$)	97.6	96.1	82.3
1,000	MEMIT-XF ($K = 6$)	98.2	97.2	80.7

Table 1: Robustness under batch editing. We jointly apply batches of 10, 100 and 1,000 CounterFact edits using MEMIT’s batch update.

Method	Efficacy	Cross-Format Gen.	Cross-Format Cons.	Specificity
AlphaEdit	99.7	70.2	65.3	73.4
AlphaEdit-XF ($K = 4$)	99.4	98.5	97.5	73.4

Table 2: Robustness under sequential editing with AlphaEdit. We edit 100 facts per round for 10 consecutive rounds and evaluate all 1,000 edited facts after the final round.

source formats. The improvements are not merely determined by the number of formats, but also by which decision interfaces are covered. Adding TF to completion ($K = 2$) substantially improves held-out YN performance ($58.1 \rightarrow 99.3$), suggesting strong transfer between similar verification-oriented interfaces, while also improving held-out MC performance ($71.1 \rightarrow 85.4$). These results suggest that cross-format transfer is structured rather than uniform: formats with similar decision interfaces exhibit particularly strong transfer, while aligning edit directions on one interface can also partially benefit heterogeneous interfaces. Incorporating MC and Triplet further improves coverage, yielding near-saturated held-out performance by $K = 4$, after which improvements become marginal. In terms of efficiency, runtime increases from 8.1s at $K = 1$ (original MEMIT) to 13.5s at $K = 6$, while $K = 4$ already achieves near-optimal performance with lower cost (11.5s). This represents a moderate computational overhead relative to the substantial improvement in cross-format generalization.

Robustness under Batch Editing To evaluate robustness of the proposed method under batch editing, we evaluate joint batches of 10, 100 and 1,000 CounterFact edits, comparing MEMIT with MEMIT-XF. Table 1 shows that MEMIT-XF main-

tains substantially stronger cross-format generalization and consistency as the batch size increases. Even with 1,000 simultaneous edits, MEMIT-XF ($K = 6$) achieves 98.2% cross-format generalization and 97.2% consistency, compared with 70.1% and 65.8% for MEMIT. Specificity also remains relatively stable: at 1,000 edits, MEMIT-XF achieves 82.3% with $K = 4$ and 80.7% with $K = 6$, compared with 83.0% for MEMIT. These results demonstrate that the proposed method preserves its cross-format advantage under large-scale batch editing with only a modest reduction in completion locality.

Robustness under Sequential Editing We additionally evaluate our approach under sequential editing using AlphaEdit, which is specifically designed to preserve previously updated knowledge across successive edits. Following the original AlphaEdit (Fang et al., 2025), we jointly edit 100 facts in each round, accumulate the model updates across 10 rounds, and evaluate all 1,000 edited facts only after the final round. AlphaEdit-XF remains highly effective under sequential editing. After 10 consecutive rounds comprising 1,000 edits, AlphaEdit-XF achieves 98.5% cross-format generalization and 97.5% cross-format consistency, substantially outperforming AlphaEdit by 28.3 pp and 32.2 pp, respectively. Both methods retain near-

perfect efficacy, while their specificity is effectively identical at 73.4%. These results demonstrate that the proposed cross-format objective remains robust in sequential editing scenarios.

5 Related Work

5.1 Knowledge Editing

Knowledge editing methods (Yao et al., 2023) can be broadly categorized into *parametric approaches*, which directly modify model parameters (Zhu et al., 2020; De Cao et al., 2021; Dai et al., 2022; Mitchell et al., 2022a; Meng et al., 2022, 2023; Hartvigsen et al., 2023; Li et al., 2024; Fang et al., 2025), and *non-parametric approaches*, which instead alter predictions through external conditioning or retrieval (Mitchell et al., 2022b; Zhong et al., 2023; Zheng et al., 2023; Zeng et al., 2024; Cohen et al., 2024). In this work, we focus on parametric editing, which enables direct intervention on model parameters and analysis of internal representations.

Existing evaluations assess knowledge editing across a range of dimensions, including counterfactual editing, cross-lingual editing, multi-hop reasoning and long-form generation (Meng et al., 2022; Zhong et al., 2023; Wang et al., 2024a; Cohen et al., 2024; Rosati et al., 2024; Zhang et al., 2024). However, evaluations of generalization are typically conducted using paraphrases or structurally similar task formats, which may overlook failures that emerge across heterogeneous task formats. Our work introduces *cross-format generalization* as a complementary evaluation dimension, showing that paraphrase-based metrics can substantially overestimate robustness by failing to capture inconsistencies across task formats.

5.2 Semantic Consistency

Semantic consistency concerns whether a model maintains invariant predictions when the input form is altered without changing its meaning (Elazar et al., 2021; Jang et al., 2022). Prior work studies this property under various perturbations, including paraphrasing, syntactic transformations and prompt formatting (Jin et al., 2020; Ravichander et al., 2020; Sclar et al., 2024; Kang et al., 2025).

Our work focuses on the underexplored dimension of *task-format variation*, which goes beyond such perturbations by capturing shifts across heterogeneous task interfaces (e.g., multiple-choice and true/false QA). Whereas most prior studies evaluate semantic consistency in unedited models,

we examine how knowledge editing itself affects consistency across formats. In particular, we show that editing introduces additional inconsistencies across semantically equivalent queries.

5.3 Knowledge Localization

Another line of work investigates where factual knowledge is represented within language models. Early studies suggest that transformer feed-forward layers function as key-value memories (Geva et al., 2021), motivating efforts to localize individual facts to specific neurons or parameters. Building on this view, Dai et al. (2022) propose knowledge neurons, while Meng et al. (2022) introduce causal tracing methods to identify modules involved in factual recall.

More recent findings challenge strong localization assumptions. Chen et al. (2024, 2025b) show that distinct neuron sets can encode the same fact and may also participate in representing multiple facts, while Chen et al. (2025a) argue that knowledge localization is query-dependent, varying even across paraphrased inputs. Our work extends this perspective by identifying *task-format variation* as an additional axis along which knowledge access can differ. Our findings support the view that knowledge is represented in distributed, format-dependent subspaces, where different task formats induce distinct access patterns, contributing to failures in cross-format generalization.

6 Conclusion

This work studies the generalization of knowledge editing across heterogeneous task formats, showing that paraphrase-based evaluations substantially overestimate robustness by overlooking inconsistencies across task formats. Knowledge editing induces format-dependent updates: distinct decision landscapes across task formats lead to format-specific local optima under single-format objectives, resulting in divergent edit directions and inconsistent factual behavior across formats. The results further suggest that explicitly aligning edit directions across heterogeneous task formats can substantially improve cross-format generalization and consistency. Overall, these findings highlight that reliable knowledge editing requires going beyond paraphrases and explicitly accounting for globally consistent behavior across heterogeneous task formulations.

Limitations

While our work identifies a critical blind spot in existing evaluation protocols and proposes a practical approach to improving cross-format consistency, we acknowledge several limitations. Our empirical study is conducted on a limited set of language models, datasets and editing methods. Although these choices cover commonly used benchmarks and representative approaches, they do not exhaust the diversity of models, datasets and editing methods. Evaluating whether the observed behaviors hold across a broader range of models, datasets and editing methods would further strengthen the generality of our findings.

We acknowledge that the templates are manually designed to provide controlled, reproducible interfaces. They are used for both evaluation and cross-format alignment, but are created once at the dataset level and reused across all samples, requiring no per-example design or intervention. Automated format generation is an important future direction.

The improved cross-format generalization comes with additional editing cost. This represents a trade-off between computational efficiency and substantially stronger cross-format generalization, rather than a cost-free improvement. We agree that reducing this overhead is important for practical deployment. Developing more efficient generalizable editing methods is a promising direction.

We hope this work encourages more comprehensive evaluation of knowledge editing and the development of methods that promote consistent behavior across diverse task formulations.

Acknowledgments

This work was partly supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Support Program (KAIST)), (RS-2022-II220984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation), (RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics), (RS-2024-00457882, AI Research Hub Project), (RS-2024-00509258, AI Guardians: Development of Robust, Controllable, and Unbiased Trustworthy AI Technology), by the InnoCORE program of the Min-

istry of Science and ICT (N10250156), and by the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT), Republic of Korea (RQT-25-120227).

References

- James A Anderson. 1972. A simple neural network generating an interactive memory. *Mathematical Biosciences*.
- Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2025a. Knowledge localization: Mission not accomplished? enter query localization! In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Yuheng Chen, Pengfei Cao, Yubo Chen, Yining Wang, Shengping Liu, Kang Liu, and Jun Zhao. 2025b. Cracking factual knowledge: A comprehensive analysis of degenerate knowledge neurons in large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- 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 (TACL)*.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. Everything is editable: Extend knowledge editing to unstructured data in large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Zilu Dong, Xiangqing Shen, and Rui Xia. 2025. Memit-merge: Addressing memit’s key-value conflicts in same-subject batch editing for llms. In *Findings of the Association for Computational Linguistics: ACL (Findings of ACL)*.

- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhिलाशा Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics (TACL)*.
- Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adapters. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Myeongjun Jang, Deuk Sin Kwon, and Thomas Lukasiewicz. 2022. Becel: Benchmark for consistency evaluation of language models. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.
- Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Cheongwoong Kang, Jongeun Baek, Yeonja Kim, and Jaesik Choi. 2025. When format changes meaning: Investigating semantic inconsistency of large language models. In *Findings of the Association for Computational Linguistics: EMNLP (Findings of EMNLP)*.
- Teuvo Kohonen. 1972. Correlation matrix memories. *IEEE Transactions on Computers (ToC)*.
- Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems (NeurIPS)*.
- Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the International Conference on Machine Learning (ICML)*.
- Karl Pearson. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*.
- Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL (Findings of NAACL)*.
- Abhिलाशा Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in bert. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*.
- Domenic Rosati, Robie Gonzales, Jinkun Chen, Xuemin Yu, Yahya Kayani, Frank Rudzicz, and Hassan Sajjad. 2024. Long-form evaluation of model editing. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, and 1 others. 2025. 2 olmo 2 furious. In *Conference on Language Modeling (COLM)*.
- Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, Jiarong Xu, and Fandong Meng. 2024a. Cross-lingual knowledge editing in large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024b. Knowledge editing for large language models: A survey. *ACM Computing Surveys (CSUR)*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Li Zeng, Yingyu Shan, Zeming Liu, Jiashu Yao, and Yuhang Guo. 2024. Fame: Towards factual multi-task model editing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Multi-Format Data Construction

Format	Template	Edit Targets
Completion	{completion_template(s, r)}	$o^c \Rightarrow o^*$
Triplet	(subject, relation, object) {triplet_template(s, r)}	$o^c \Rightarrow o^*$
ODQA	Question: {whqa_template(s, r)} Answer:	$o^c \Rightarrow o^*$
MC	Output only answer letter. Question: {whqa_template(s, r)} Options: A. { o^* / o^c } B. { o^c / o^* } Answer:	{ $B \Rightarrow A$ / $A \Rightarrow B$ }
TF	True or False? Statement: {completion_template(s, r)} { o^* } Answer:	{False $\Rightarrow$ True}
YN	Yes or No? Statement: {ynqa_template(s, r)} { o^* }? Answer:	{No $\Rightarrow$ Yes}

Table 3: Multi-format templates used in our evaluation framework. Each factual edit ($s, r, o^c \Rightarrow o^*$) is instantiated across these formats.

Format	Example Prompt	Edit Targets
Completion	Eiffel Tower is located in	Paris $\Rightarrow$ Rome
Triplet	(subject, relation, object) (Eiffel Tower, location,	Paris $\Rightarrow$ Rome
ODQA	Question: Where is Eiffel Tower located? Answer:	Paris $\Rightarrow$ Rome
MC	Output only answer letter. Question: Where is Eiffel Tower located? Options: A. Rome B. Paris Answer:	B $\Rightarrow$ A
TF	True or False? Statement: Eiffel Tower is located in Rome Answer:	False $\Rightarrow$ True
YN	Yes or No? Question: Is Eiffel Tower located in Rome? Answer:	No $\Rightarrow$ Yes

Table 4: Multi-format examples for the target edit (*Eiffel Tower, location, Paris $\Rightarrow$ Rome*).

Relation	completion_template	triplet_template
P17	{s} is located in	({s}, country,
P19	{s} was born in	({s}, place of birth,
P20	{s} died in	({s}, place of death,
P27	{s} is a citizen of	({s}, country of citizenship,
P30	{s} is located in	({s}, continent,
P36	The capital of {s} is	({s}, capital,
P37	The official language of {s} is	({s}, official language,
P39	{s} has the position of	({s}, position held,
P101	{s} works in the field of	({s}, field of work,
P103	The native language of {s} is	({s}, native language,
P106	The occupation of {s} is	({s}, occupation,
P108	{s} works for	({s}, employer,
P127	{s} is owned by	({s}, owned by,
P131	{s} is located in	({s}, located in the administrative territorial entity,
P136	The genre played by {s} is	({s}, genre,
P138	{s} is named after	({s}, named after,
P140	{s} is affiliated with the religion of	({s}, religion,
P159	The headquarter of {s} is in	({s}, headquarters location,
P176	{s} is produced by	({s}, manufacturer,
P178	{s} is developed by	({s}, developer,
P190	{s} is a twin city of	({s}, twinned administrative body,
P264	{s} is represented by music label	({s}, record label,
P276	{s} is located in	({s}, location,
P364	The original language of {s} is	({s}, original language of film or TV show,
P407	{s} was written in	({s}, language of work or name,
P413	{s} plays in the position of	({s}, position played on team,
P449	{s} was originally aired on	({s}, original network,
P463	{s} is a member of	({s}, member of,
P495	{s} was created in	({s}, country of origin,
P740	{s} was founded in	({s}, location of formation,
P937	{s} used to work in	({s}, work location,
P1303	{s} plays the	({s}, instrument,
P1412	{s} used to communicate in	({s}, languages spoken, written or signed,

Table 5: Relational templates for completion_template and triplet_template.

Relation	whqa_template	ynqa_template
P17	Where is {s} located?	Is {s} located in
P19	Where was {s} born?	Was {s} born in
P20	Where did {s} die?	Did {s} die in
P27	What country is {s} a citizen of?	Is {s} a citizen of
P30	Where is {s} located?	Is {s} located in
P36	What is the capital of {s}?	Is the capital of {s}
P37	What is the official language of {s}?	Is the official language of {s}
P39	What position does {s} hold?	Does {s} have the position of
P101	What is the field of work for {s}?	Does {s} work in the field of
P103	What is the native language of {s}?	Is the native language of {s}
P106	What is the occupation of {s}?	Is the occupation of {s}
P108	Who does {s} work for?	Does {s} work for
P127	Who owns {s}?	Is {s} owned by
P131	Where is {s} located?	Is {s} located in
P136	What genre does {s} play?	Is the genre played by {s}
P138	Who or what is {s} named after?	Is {s} named after
P140	What religion is {s} affiliated with?	Is {s} affiliated with the religion of
P159	Where is the headquarter of {s}?	Is the headquarter of {s} in
P176	Who produced {s}?	Is {s} produced by
P178	Who developed {s}?	Is {s} developed by
P190	What city is twinned with {s}?	Is {s} a twin city of
P264	What music label represents {s}?	Is {s} represented by music label
P276	Where is {s} located?	Is {s} located in
P364	What is the original language of {s}?	Is the original language of {s}
P407	What language was {s} written in?	Was {s} written in
P413	What position does {s} play in?	Does {s} play in the position of
P449	Which network originally aired {s}?	Was {s} originally aired on
P463	What is {s} a member of?	Is {s} a member of
P495	Where was {s} created?	Was {s} created in
P740	Where was {s} founded?	Was {s} founded in
P937	Where did {s} use to work?	Did {s} use to work in
P1303	What instrument does {s} play?	Does {s} play the
P1412	What language does {s} use to communicate?	Does {s} use to communicate in

Table 6: Relational templates for whqa_template and ynqa_template.

B Implementation Details

B.1 MEMIT

We tune the target layers separately for each model by identifying the critical MLP layers following the original implementation of MEMIT (Meng et al., 2023). On Llama-3.2-1B-Instruct, we set the target layers $\mathcal{R} = \{0, 1, 2, 3, 4, 5\}$. On Llama-3.2-3B-Instruct, we set the target layers $\mathcal{R} = \{2, 3, 4, 5, 6, 7\}$. On Llama-3.1-8B-Instruct, we set the target layers $\mathcal{R} = \{2, 3, 4, 5, 6, 7\}$. On OLMo2-1B-Instruct, we set the target layers $\mathcal{R} = \{2, 3, 4, 5, 6, 7\}$. On OLMo2-7B-Instruct, we set the target layers $\mathcal{R} = \{3, 4, 5, 6, 7, 8\}$. On OLMo2-13B-Instruct, we set the target layers $\mathcal{R} = \{2, 3, 4, 5, 6, 7\}$. On Qwen3-0.6B-Instruct, we set the target layers $\mathcal{R} = \{1, 2, 3, 4, 5, 7\}$. On Qwen3-1.7B-Instruct, we set the target layers $\mathcal{R} = \{3, 4, 5, 6, 7, 8\}$. On Qwen3-4B-Instruct, we set the target layers $\mathcal{R} = \{6, 7, 8, 9, 10, 11\}$. On Qwen3-8B-Instruct, we set the target layers $\mathcal{R} = \{5, 6, 7, 8, 9, 10\}$. On Qwen3-14B-Instruct, we set the target layers $\mathcal{R} = \{7, 8, 9, 10, 11, 12\}$.

All other hyperparameters are applied consistently across models to ensure comparability. We set the covariance adjustment hyperparameter $\lambda = 1$. While prior work often employs large regularization values, we find that such strong regularization can overly constrain the update and limit generalization across formats. In our single-edit setting, the update remains sufficiently localized even with a small λ , making large regularization unnecessary and allowing better transfer across formats. Covariance statistics are estimated based on 100,000 instances from Wikitext.

The other hyperparameters follow the original MEMIT configuration. Specifically, the target value vector v^* is optimized for 25 gradient steps with a learning rate of $5e-1$. We clamp the L_2 norm of edit delta such that it does not exceed 0.75 times the norm of the original hidden state. We set the weight decay parameter to 0.5 and KL regularization factor to 0.0625. We restrict the number of edits to one, which is sufficient to reveal the generalization failure. Unless otherwise noted, we follow the original implementation of MEMIT (Meng et al., 2023).

B.2 ROME

We use the final layer of the MEMIT’s target layers as the target layer. We set the clamp norm factor to 4. The other hyperparameters follow the MEMIT.

B.3 AlphaEdit

We set the nullspace threshold to $2e-2$ and L2 factor to 10. The other hyperparameters follow the MEMIT.

B.4 UnKE

We use the final layer of the MEMIT’s target layers as the boundary layer. We set the clamp norm factor to 4. The other hyperparameters follow the MEMIT while UnKE-specific hyperparameters follow the original UnKE implementation.

C Additional Results on Cross-Format Generalization

C.1 Additional Qualitative Examples

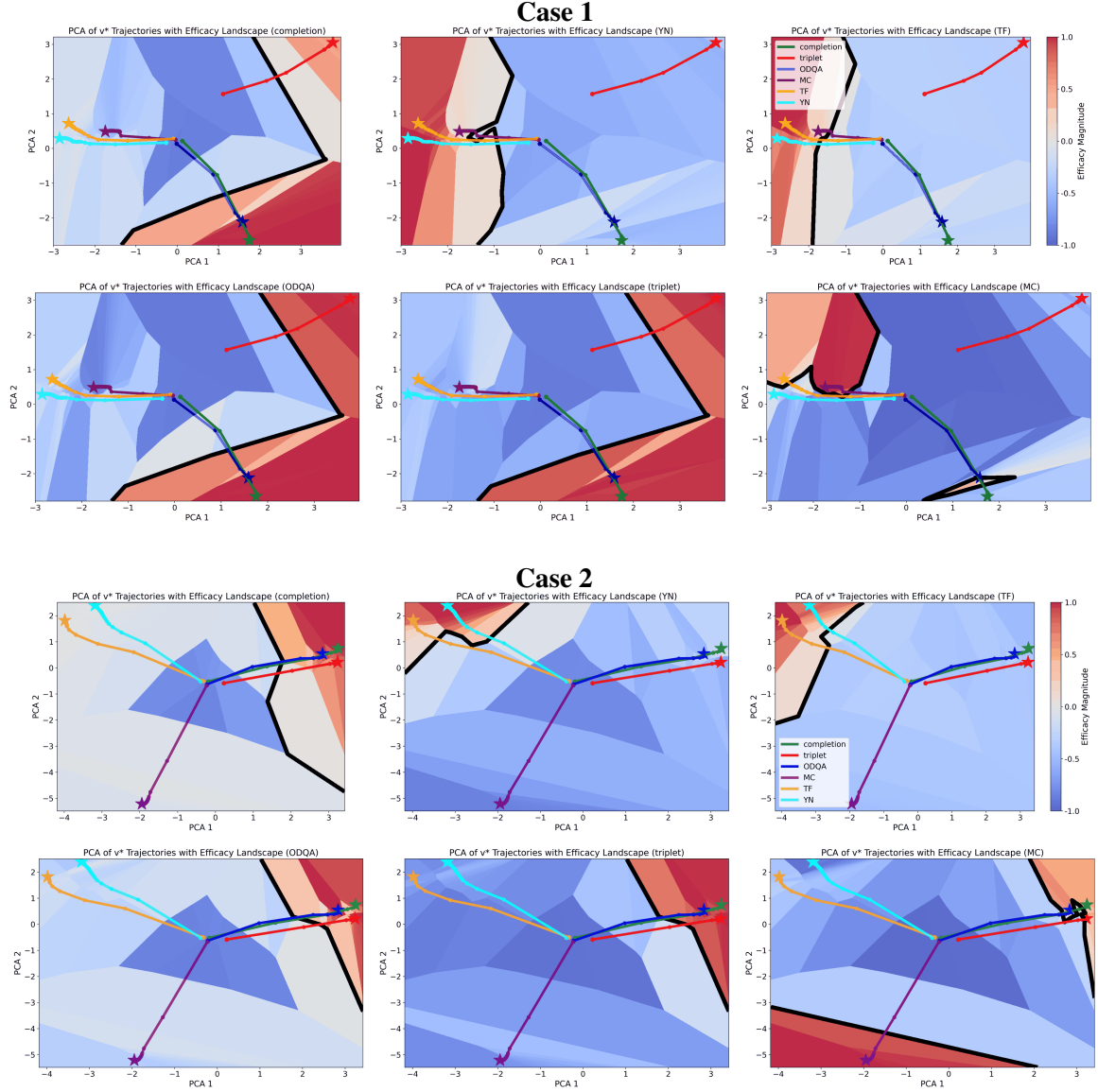


Figure 9: Additional qualitative examples of editing dynamics across prompt formats. Each case visualizes PCA-projected v^* trajectories over the corresponding efficacy landscape for six prompt formats: completion, YN, TF, ODQA, triplet and MC. Across additional examples, we consistently observe structured format-dependent geometries, where semantically or operationally similar task formats (e.g., TF/YN) tend to exhibit more aligned edit trajectories than heterogeneous interfaces such as MC.

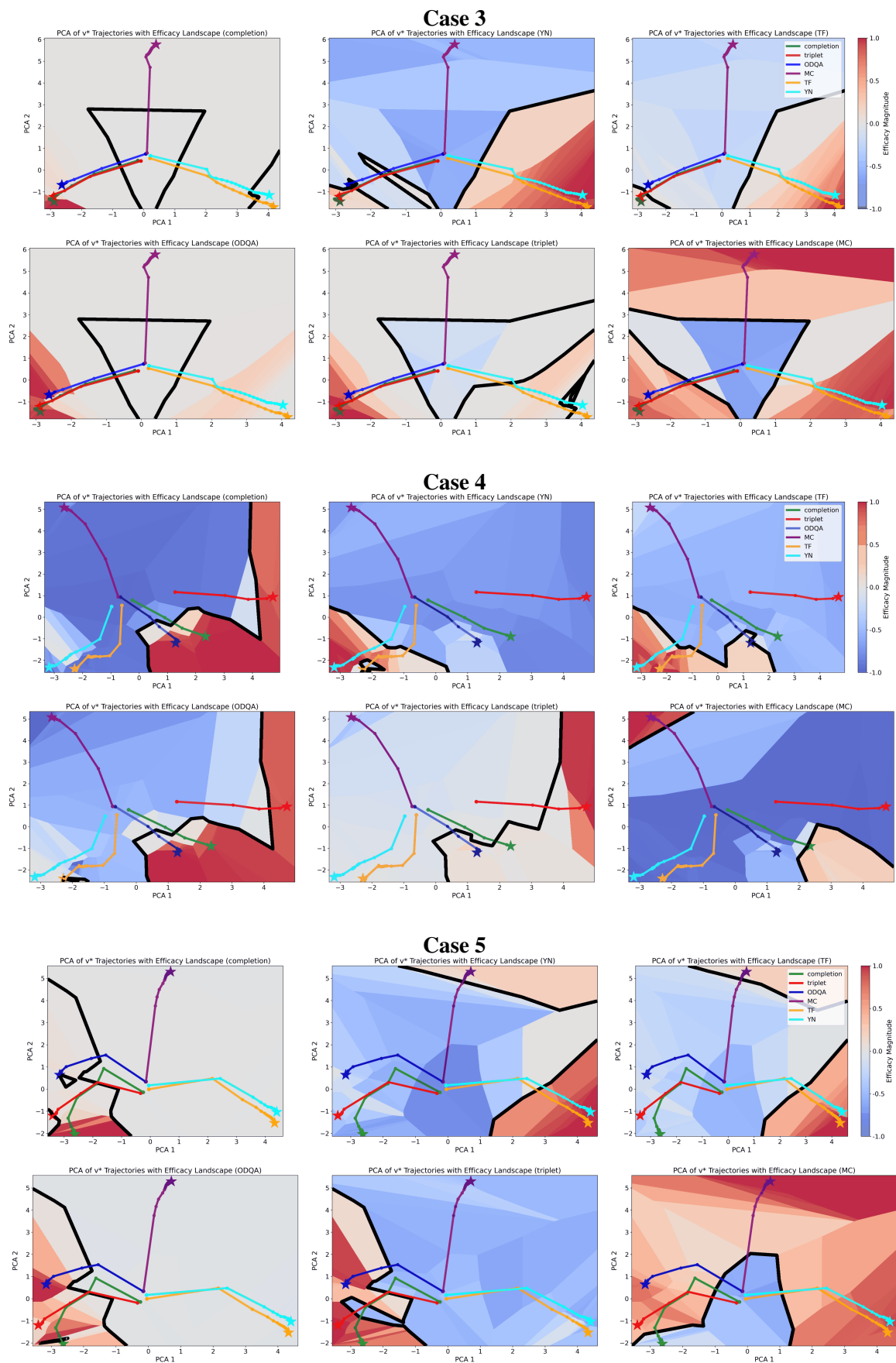


Figure 10: Additional qualitative examples of editing dynamics across prompt formats, continued.

C.2 Additional Quantitative Results

Method	Eff. ↑	Paraphrase			Cross-format							
		Gen. ↑	Spe. ↑	Con. ↑	Gen. ↑	ODQA	Triplet	MC	TF	YN	Spe. ↑	Con. ↑
Pre-edit	12.5 (2.1)	15.7 (2.0)	84.4 (1.5)	90.5 (1.4)	10.9 (1.2)	11.8 (2.0)	15.5 (2.2)	14.5 (2.2)	5.8 (1.4)	6.8 (1.6)	86.7 (0.9)	86.3 (1.3)
ROME	100.0 (0.0)	98.5 (0.6)	74.0 (1.9)	98.1 (0.7)	75.1 (1.5)	98.4 (0.8)	98.6 (0.7)	66.1 (2.9)	54.3 (3.1)	57.9 (3.1)	79.3 (1.1)	69.7 (1.6)
MEMIT	100.0 (0.0)	94.0 (1.2)	83.4 (1.6)	93.4 (1.2)	74.5 (1.5)	96.8 (1.1)	93.8 (1.5)	71.1 (2.8)	52.5 (3.1)	58.1 (3.1)	85.9 (0.9)	69.9 (1.5)
AlphaEdit	98.7 (0.7)	84.0 (1.9)	82.5 (1.6)	85.2 (1.7)	67.7 (1.8)	89.4 (1.9)	84.6 (2.2)	63.7 (3.0)	48.5 (3.1)	52.5 (3.1)	85.0 (0.9)	66.9 (1.5)
UnKE	100.0 (0.0)	69.5 (2.3)	32.5 (1.9)	69.4 (2.1)	24.8 (1.8)	29.9 (2.8)	29.5 (2.8)	23.6 (2.6)	19.1 (2.4)	22.1 (2.6)	71.5 (1.4)	57.2 (0.8)
MEMIT-Para	99.9 (0.2)	99.9 (0.1)	83.2 (1.6)	99.8 (0.2)	74.4 (1.5)	97.2 (1.0)	96.2 (1.2)	72.5 (2.8)	50.7 (3.1)	55.3 (3.1)	85.1 (0.9)	69.1 (1.5)
MEMIT-XF	100.0 (0.0)	96.0 (1.0)	83.5 (1.6)	95.3 (1.1)	99.8 (0.1)	99.9 (0.2)	99.7 (0.3)	99.5 (0.4)	100.0 (0.0)	100.0 (0.0)	85.0 (0.9)	99.7 (0.2)
AlphaEdit-XF	99.6 (0.4)	95.1 (1.0)	82.7 (1.6)	94.3 (1.2)	<u>99.7 (0.2)</u>	100.0 (0.0)	<u>99.5 (0.4)</u>	99.6 (0.4)	<u>99.6 (0.4)</u>	100.0 (0.0)	82.9 (1.0)	<u>99.5 (0.3)</u>

Table 7: Editing baseline comparison (Llama-3.2-3B-Instruct, CounterFact-1k).

Model	Eff. ↑	Paraphrase			Cross-format							
		Gen. ↑	Spe. ↑	Con. ↑	Gen. ↑	ODQA	Triplet	MC	TF	YN	Spe. ↑	Con. ↑
Llama-3.2-1B-Instruct	15.2 (2.2)	19.3 (2.1)	81.8 (1.6)	86.7 (1.7)	38.3 (1.7)	13.6 (2.1)	18.1 (2.4)	43.9 (3.1)	61.1 (3.0)	54.8 (3.1)	52.0 (1.3)	63.0 (1.4)
+ MEMIT	100.0 (0.0)	94.8 (1.1)	80.3 (1.7)	94.4 (1.1)	81.6 (1.2)	97.4 (1.0)	95.0 (1.4)	49.1 (3.1)	85.1 (2.2)	81.6 (2.4)	51.0 (1.3)	75.2 (1.3)
+ MEMIT-XF	99.9 (0.2)	95.7 (1.0)	80.9 (1.7)	95.1 (1.1)	99.2 (0.3)	99.7 (0.3)	99.6 (0.4)	97.1 (1.0)	99.9 (0.2)	99.9 (0.2)	49.3 (1.3)	98.8 (0.4)
Llama-3.2-3B-Instruct	12.5 (2.1)	15.7 (2.0)	84.4 (1.5)	90.5 (1.4)	10.9 (1.2)	11.8 (2.0)	15.5 (2.2)	14.5 (2.2)	5.8 (1.4)	6.8 (1.6)	86.7 (0.9)	86.3 (1.3)
+ MEMIT	100.0 (0.0)	94.0 (1.2)	83.4 (1.6)	93.4 (1.2)	74.5 (1.5)	96.8 (1.1)	93.8 (1.5)	71.1 (2.8)	52.5 (3.1)	58.1 (3.1)	85.9 (0.9)	69.9 (1.5)
+ MEMIT-Para	99.9 (0.2)	99.9 (0.1)	83.2 (1.6)	99.8 (0.2)	74.4 (1.5)	97.2 (1.0)	96.2 (1.2)	72.5 (2.8)	50.7 (3.1)	55.3 (3.1)	85.1 (0.9)	69.1 (1.5)
+ MEMIT-XF	100.0 (0.0)	96.0 (1.0)	83.5 (1.6)	95.3 (1.1)	99.8 (0.1)	99.9 (0.2)	99.7 (0.3)	99.5 (0.4)	100.0 (0.0)	100.0 (0.0)	85.0 (0.9)	99.7 (0.2)
Llama-3.1-8B-Instruct	6.7 (1.6)	9.8 (1.6)	88.7 (1.3)	92.8 (1.3)	6.2 (1.0)	6.3 (1.5)	8.1 (1.7)	6.4 (1.5)	3.9 (1.2)	6.5 (1.5)	89.3 (0.8)	92.0 (1.1)
+ MEMIT	100.0 (0.0)	96.9 (0.8)	87.9 (1.3)	96.3 (1.0)	86.2 (1.3)	95.7 (1.3)	96.5 (1.1)	81.0 (2.4)	78.1 (2.6)	79.9 (2.5)	88.5 (0.8)	83.4 (1.4)
+ MEMIT-XF	100.0 (0.0)	98.0 (0.7)	88.0 (1.3)	97.7 (0.8)	99.8 (0.1)	99.7 (0.3)	99.8 (0.3)	99.9 (0.2)	99.9 (0.2)	99.9 (0.2)	87.4 (0.9)	99.7 (0.2)
OLMo2-1B-Instruct	19.1 (2.4)	18.6 (2.1)	79.7 (1.7)	86.3 (1.7)	13.6 (1.1)	15.2 (2.2)	18.2 (2.4)	31.1 (2.9)	3.1 (1.1)	0.2 (0.3)	85.1 (0.9)	79.7 (1.4)
+ MEMIT	99.9 (0.2)	96.2 (0.9)	77.9 (1.7)	95.7 (1.0)	60.6 (1.3)	97.7 (0.9)	95.5 (1.3)	61.6 (3.0)	36.4 (3.0)	11.9 (2.0)	84.2 (0.9)	54.0 (1.1)
+ MEMIT-XF	99.9 (0.2)	97.6 (0.7)	78.0 (1.7)	97.2 (0.8)	99.7 (0.2)	99.8 (0.3)	100.0 (0.0)	99.2 (0.6)	99.9 (0.2)	99.6 (0.4)	82.4 (1.0)	99.6 (0.2)
OLMo2-7B-Instruct	10.6 (1.9)	14.1 (1.9)	86.2 (1.4)	88.9 (1.5)	10.7 (1.3)	8.6 (1.7)	10.3 (1.9)	12.4 (2.0)	9.7 (1.8)	12.4 (2.0)	87.0 (0.9)	87.7 (1.3)
+ MEMIT	100.0 (0.0)	91.7 (1.4)	85.4 (1.4)	90.9 (1.4)	80.0 (1.6)	97.8 (0.9)	94.1 (1.5)	73.5 (2.7)	64.7 (3.0)	69.9 (2.8)	86.5 (1.0)	77.6 (1.6)
+ MEMIT-XF	100.0 (0.0)	94.9 (1.1)	85.2 (1.4)	94.0 (1.2)	99.9 (0.1)	100.0 (0.0)	99.9 (0.2)	99.9 (0.2)	99.5 (0.4)	100.0 (0.0)	85.8 (1.0)	99.8 (0.2)
OLMo2-13B-Instruct	8.8 (1.8)	11.4 (1.7)	87.7 (1.3)	90.8 (1.4)	11.0 (1.3)	6.0 (1.5)	8.0 (1.7)	8.1 (1.7)	10.1 (1.9)	23.0 (2.6)	91.1 (0.7)	87.0 (1.3)
+ MEMIT	93.8 (1.5)	72.5 (2.4)	87.5 (1.3)	77.5 (2.0)	66.4 (2.1)	84.3 (2.3)	78.4 (2.6)	55.2 (3.1)	54.1 (3.1)	59.9 (3.0)	90.9 (0.7)	71.5 (1.5)
+ MEMIT-XF	89.6 (1.9)	71.4 (2.4)	87.5 (1.3)	77.8 (1.9)	93.7 (1.1)	92.9 (1.6)	91.5 (1.7)	93.0 (1.6)	94.3 (1.4)	96.7 (1.1)	90.7 (0.7)	92.9 (1.0)
Qwen3-0.6B	26.4 (2.7)	28.6 (2.5)	71.9 (2.1)	84.0 (1.8)	34.1 (1.3)	27.5 (2.8)	27.1 (2.8)	25.5 (2.7)	89.9 (1.9)	0.4 (0.4)	59.6 (0.9)	60.2 (0.9)
+ MEMIT	99.9 (0.2)	87.9 (1.7)	70.7 (2.1)	87.3 (1.6)	68.2 (1.2)	79.6 (2.5)	86.4 (2.1)	70.2 (2.8)	96.1 (1.2)	8.6 (1.7)	59.0 (1.0)	59.5 (0.9)
+ MEMIT-XF	97.8 (0.9)	91.2 (1.4)	70.8 (2.1)	90.2 (1.5)	96.8 (0.6)	97.3 (1.0)	97.8 (0.9)	96.5 (1.1)	99.5 (0.4)	92.8 (1.6)	59.6 (1.0)	94.8 (0.8)
Qwen3-1.7B	19.4 (2.5)	21.6 (2.3)	78.1 (1.9)	87.9 (1.6)	28.0 (1.6)	16.3 (2.3)	20.8 (2.5)	16.9 (2.3)	20.5 (2.5)	65.3 (3.0)	66.7 (1.2)	68.7 (1.3)
+ MEMIT	99.9 (0.2)	93.3 (1.2)	76.8 (1.9)	92.5 (1.3)	86.5 (1.3)	91.1 (1.8)	93.2 (1.6)	80.4 (2.5)	75.6 (2.7)	92.4 (1.6)	65.5 (1.2)	83.1 (1.4)
+ MEMIT-XF	99.7 (0.3)	96.7 (0.9)	76.4 (1.9)	96.3 (1.0)	99.6 (0.2)	99.9 (0.2)	99.4 (0.5)	99.2 (0.6)	100.0 (0.0)	99.7 (0.3)	63.9 (1.2)	99.4 (0.3)
Qwen3-4B	17.1 (2.3)	19.7 (2.2)	80.9 (1.8)	89.5 (1.5)	11.5 (1.4)	12.6 (2.1)	16.0 (2.3)	13.4 (2.1)	9.4 (1.8)	6.2 (1.5)	81.9 (1.2)	86.4 (1.4)
+ MEMIT	99.9 (0.2)	92.1 (1.3)	80.0 (1.8)	91.2 (1.4)	75.6 (1.6)	92.9 (1.6)	94.9 (1.4)	73.6 (2.7)	60.1 (3.0)	56.5 (3.1)	81.1 (1.2)	72.5 (1.6)
+ MEMIT-XF	99.9 (0.2)	94.8 (1.1)	79.7 (1.8)	94.2 (1.2)	99.7 (0.2)	99.7 (0.3)	99.8 (0.3)	99.3 (0.5)	100.0 (0.0)	99.6 (0.4)	80.2 (1.2)	99.6 (0.3)
Qwen3-8B	13.5 (2.1)	15.7 (2.0)	84.1 (1.6)	90.6 (1.4)	9.1 (1.2)	10.9 (1.9)	12.9 (2.1)	10.6 (1.9)	3.6 (1.2)	7.5 (1.6)	87.6 (0.9)	89.1 (1.3)
+ MEMIT	99.9 (0.2)	95.7 (1.0)	83.1 (1.6)	95.1 (1.1)	80.4 (1.5)	94.2 (1.4)	98.0 (0.9)	78.6 (2.5)	62.5 (3.0)	68.8 (2.9)	86.7 (1.0)	77.2 (1.6)
+ MEMIT-XF	99.9 (0.2)	97.7 (0.8)	83.1 (1.6)	97.5 (0.8)	99.7 (0.2)	99.7 (0.3)	99.7 (0.3)	99.7 (0.3)	99.8 (0.3)	99.7 (0.3)	85.8 (1.0)	99.7 (0.2)
Qwen3-14B	11.0 (1.9)	14.1 (1.9)	85.2 (1.5)	90.9 (1.4)	9.5 (1.2)	8.5 (1.7)	10.6 (1.9)	8.3 (1.7)	3.9 (1.2)	16.2 (2.3)	87.0 (0.9)	88.4 (1.3)
+ MEMIT	100.0 (0.0)	95.0 (1.1)	84.3 (1.5)	94.6 (1.1)	86.0 (1.3)	98.2 (0.8)	97.7 (0.9)	84.4 (2.3)	67.8 (2.9)	81.9 (2.4)	86.2 (0.9)	82.7 (1.4)
+ MEMIT-XF	99.9 (0.2)	97.5 (0.8)	84.1 (1.5)	97.3 (0.8)	99.7 (0.2)	99.9 (0.2)	100.0 (0.0)	99.6 (0.4)	99.6 (0.4)	99.6 (0.4)	85.3 (1.0)	99.7 (0.3)

Table 8: Model generalization results (MEMIT, CounterFact-1k).

Dataset	Eff. ↑	Paraphrase			Cross-format							
		Gen. ↑	Spe. ↑	Con. ↑	Gen. ↑	ODQA	Triplet	MC	TF	YN	Spe. ↑	Con. ↑
CounterFact-1k	12.5 (2.1)	15.7 (2.0)	84.4 (1.5)	90.5 (1.4)	10.9 (1.2)	11.8 (2.0)	15.5 (2.2)	14.5 (2.2)	5.8 (1.4)	6.8 (1.6)	86.7 (0.9)	86.3 (1.3)
+ MEMIT	100.0 (0.0)	94.0 (1.2)	83.4 (1.6)	93.4 (1.2)	74.5 (1.5)	96.8 (1.1)	93.8 (1.5)	71.1 (2.8)	52.5 (3.1)	58.1 (3.1)	85.9 (0.9)	69.9 (1.5)
+ MEMIT-Para	99.9 (0.2)	99.9 (0.1)	83.2 (1.6)	99.8 (0.2)	74.4 (1.5)	97.2 (1.0)	96.2 (1.2)	72.5 (2.8)	50.7 (3.1)	55.3 (3.1)	85.1 (0.9)	69.1 (1.5)
+ MEMIT-XF	100.0 (0.0)	96.0 (1.0)	83.5 (1.6)	95.3 (1.1)	99.8 (0.1)	99.9 (0.2)	99.7 (0.3)	99.5 (0.4)	100.0 (0.0)	100.0 (0.0)	85.0 (0.9)	99.7 (0.2)
KnowEdit-Wiki-Counterfact	21.7 (2.8)	25.3 (2.9)	-	89.7 (2.1)	10.7 (1.2)	22.1 (2.8)	-	15.3 (2.4)	1.7 (0.9)	3.7 (1.3)	-	81.3 (1.7)
+ MEMIT	99.5 (0.5)	86.3 (2.3)	-	86.8 (2.3)	52.4 (1.9)	94.0 (1.6)	-	56.3 (3.4)	30.3 (3.1)	29.1 (3.1)	-	53.7 (1.5)
+ MEMIT-XF	99.5 (0.5)	88.1 (2.2)	-	88.6 (2.2)	99.3 (0.3)	99.8 (0.3)	-	98.2 (0.9)	99.6 (0.4)	99.6 (0.4)	-	98.9 (0.5)
KnowEdit-Wiki-Recent	5.6 (1.3)	11.1 (1.7)	-	88.6 (1.7)	26.2 (1.8)	30.1 (2.5)	-	-	20.0 (2.2)	28.4 (2.5)	-	73.8 (1.5)
+ MEMIT	99.9 (0.2)	68.9 (2.6)	-	69.0 (2.5)	70.2 (1.7)	92.5 (1.5)	-	-	60.9 (2.7)	57.1 (2.7)	-	68.4 (1.7)
+ MEMIT-XF	99.5 (0.4)	69.4 (2.5)	-	69.7 (2.5)	99.8 (0.1)	99.7 (0.3)	-	-	99.9 (0.2)	99.9 (0.2)	-	99.6 (0.2)
KnowEdit-ZsRE	33.0 (2.6)	33.3 (2.6)	-	91.1 (1.5)	28.4 (1.5)	30.7 (2.5)	-	33.5 (2.6)	25.7 (2.4)	23.7 (2.3)	-	65.3 (1.4)
+ MEMIT	96.3 (1.0)	90.6 (1.6)	-	94.3 (1.3)	68.8 (1.4)	92.8 (1.4)	-	65.3 (2.6)	56.1 (2.7)	60.9 (2.7)	-	64.6 (1.4)
+ MEMIT-XF	96.4 (1.0)	90.0 (1.6)	-	93.5 (1.3)	98.9 (0.3)	96.3 (1.0)	-	99.5 (0.4)	99.9 (0.2)	99.9 (0.2)	-	97.6 (0.6)

Table 9: Dataset generalization results (MEMIT, Llama-3.2-3B-Instruct).

Variant	Eff. ↑	Paraphrase			Cross-format							
		Gen. ↑	Spe. ↑	Con. ↑	Gen. ↑	ODQA	Triplet	MC	TF	YN	Spe. ↑	Con. ↑
MEMIT-XF	100.0 (0.0)	96.0 (1.0)	<u>83.5 (1.6)</u>	95.3 (1.1)	<u>99.8 (0.1)</u>	<u>99.9 (0.2)</u>	<u>99.7 (0.3)</u>	<u>99.5 (0.4)</u>	100.0 (0.0)	100.0 (0.0)	85.0 (0.9)	<u>99.7 (0.2)</u>
w/o joint loss	100.0 (0.0)	89.5 (1.5)	82.9 (1.6)	88.3 (1.6)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	82.1 (1.0)	100.0 (0.0)
w/o multi-key	100.0 (0.0)	94.3 (1.1)	83.6 (1.6)	93.1 (1.3)	99.2 (0.3)	99.0 (0.6)	98.2 (0.8)	99.4 (0.5)	99.8 (0.3)	99.7 (0.3)	85.3 (0.9)	98.9 (0.4)
w/o multi-value	100.0 (0.0)	<u>95.5 (1.0)</u>	83.4 (1.6)	<u>94.9 (1.1)</u>	75.1 (1.5)	97.7 (0.9)	97.7 (0.9)	71.1 (2.8)	52.4 (3.1)	56.4 (3.1)	85.7 (0.9)	69.8 (1.6)

Table 10: Ablation study on alignment components (Llama-3.2-3B-Instruct, CounterFact-1k).

Method	Eff. ↑	Paraphrase			Cross-format							
		Gen. ↑	Spe. ↑	Con. ↑	Gen. ↑	ODQA	Triplet	MC	TF	YN	Spe. ↑	Con. ↑
MEMIT	100.0 (0.0)	94.0 (1.2)	83.4 (1.6)	93.4 (1.2)	74.5 (1.5)	96.8 (1.1)	93.8 (1.5)	71.1 (2.8)	52.5 (3.1)	58.1 (3.1)	85.9 (0.9)	69.9 (1.5)
MEMIT-XF (K=2)	99.9 (0.2)	90.7 (1.4)	83.7 (1.6)	89.5 (1.5)	93.6 (0.8)	95.2 (1.3)	88.0 (2.0)	85.4 (2.2)	100.0 (0.0)	99.3 (0.5)	85.3 (0.9)	90.4 (1.1)
MEMIT-XF (K=3)	99.9 (0.2)	90.5 (1.4)	83.7 (1.6)	89.5 (1.5)	96.2 (0.6)	96.4 (1.2)	86.2 (2.1)	99.7 (0.3)	100.0 (0.0)	98.8 (0.7)	84.9 (0.9)	94.1 (0.9)
MEMIT-XF (K=4)	100.0 (0.0)	94.5 (1.1)	83.6 (1.6)	93.6 (1.2)	99.3 (0.2)	98.3 (0.8)	99.5 (0.4)	99.7 (0.3)	99.9 (0.2)	99.2 (0.6)	85.0 (1.0)	98.9 (0.4)
MEMIT-XF (K=5)	100.0 (0.0)	95.8 (1.0)	83.5 (1.6)	94.9 (1.1)	99.6 (0.2)	99.9 (0.2)	99.7 (0.3)	99.5 (0.4)	99.9 (0.2)	99.0 (0.6)	84.4 (1.0)	99.4 (0.3)
MEMIT-XF (K=6)	100.0 (0.0)	96.0 (1.0)	83.5 (1.6)	95.3 (1.1)	99.8 (0.1)	99.9 (0.2)	99.7 (0.3)	99.5 (0.4)	100.0 (0.0)	100.0 (0.0)	85.0 (0.9)	99.7 (0.2)

Table 11: Ablation study on the number of source formats (Llama-3.2-3B-Instruct, CounterFact-1k).

C.3 Distinguishing Consistency from Correctness

The consistency metric measures agreement across six formats, so consistently wrong predictions can still receive high consistency scores. We therefore report *Consistent-Accuracy* on CounterFact-1k, which equals 1 only when all six answers are correct, and 0 otherwise.

Setting	Consistency	Consistent-Accuracy
Pre-edit	86.3	0.4
ROME	69.7	37.4
MEMIT	69.9	37.1
AlphaEdit	66.9	30.2
UnKE	57.2	2.9
MEMIT-XF	99.7	99.1

Table 12: Consistency and Consistent-Accuracy across six task formats. Consistent-Accuracy equals 1 only when all six answers are correct, and 0 otherwise.

The pre-edit result clearly illustrates the distinction: raw consistency is 86.3%, but only 0.4% of samples are correct across all formats because the edited targets have not yet been learned. A similar gap remains after single-format editing: ROME, MEMIT, AlphaEdit and UnKE achieve 57.2–69.9% consistency, but only 2.9–37.4% Consistent-Accuracy. In contrast, MEMIT-XF achieves 99.7% consistency and 99.1% Consistent-Accuracy, showing that its predictions are both consistent and correct.

C.4 Impact of Multiple-Choice Option-Order Bias

We tested whether multiple-choice option-order bias (Pezeshkpour and Hruschka, 2024) contributes to the observed drop in cross-format generalization, so part of the degradation may not be solely caused by knowledge editing itself. To quantify the option-order effect, we created two versions of each MC question in CounterFact-1k with identical options in different orders and classify a sample as order-sensitive when exactly one version is answered correctly.

Setting	Order-Sensitive Samples (%)	Consistency: Order-Sensitive	Consistency: Others
Pre-edit	24.6	65.1	93.5
ROME	32.2	63.7	72.5
MEMIT	31.3	57.7	74.0
AlphaEdit	38.4	58.4	72.1
MEMIT-XF	7.6	97.8	99.7

Table 13: Option-order sensitivity and cross-format consistency on CounterFact-1k. A sample is considered order-sensitive when exactly one of two multiple-choice versions with identical options in different orders is answered correctly.

The pre-edit result confirms that option-order sensitivity is associated with lower cross-format consistency even before editing. However, this bias does not fully explain the consistency degradation observed after single-format editing. Consistency decreases after ROME, MEMIT and AlphaEdit in both groups, including from 93.5% to 72.5%, 74.0% and 72.1%, respectively, among samples without observed order sensitivity. Notably, MEMIT-XF reduces the proportion of order-sensitive samples to 7.6% and achieves near-perfect consistency in both groups.

C.5 Impact of Regularization

We tested whether regularization overconstrains the update and limits cross-format generalization. MEMIT applies two regularizers during value-vector optimization. KL regularization minimizes divergence between the edited and pre-edit predictive distributions for a generic prompt, “{subject} is a,” thereby

preserving existing predictions about the subject; setting its coefficient from 0.0625 to 0 removes this constraint. Value weight decay constrains the optimized value vector to remain close to its initialization. Setting its coefficient from 0.5 to 0 removes this constraint. In MEMIT’s closed-form update, the covariance regularization weight balances new and old associations. Reducing it from 1 to 0.01 substantially weakens the contribution of old associations and gives the new associations greater influence. We use a small positive value rather than zero for numerical stability.

We conducted a $2 \times 2 \times 2$ factorial ablation over these values on CounterFact-1K. The most relaxed configuration achieved the highest cross-format generalization and consistency (76.52/71.55 vs. the default 73.62/68.86). However, it was nearly tied with the configuration retaining the default covariance weight (76.50/71.53), as the default value of 1 is small enough in single-edit settings. Specificity remained nearly unchanged (83.0 vs. 83.3). Therefore, relaxing the value-vector regularizers modestly improves transfer without a meaningful locality trade-off. Even the best setting recovers only 2.7 consistency points, far less than our cross-format alignment. Therefore, overregularization may contribute to the limited transfer of single-format editing but is not the primary cause of the observed inconsistency.

C.6 Consistency and Specificity

Consistency measures agreement across task formats, whereas specificity measures preservation on unrelated inputs. To test whether inconsistency is introduced by preserving locality, we classified samples with lower post-edit than pre-edit completion specificity as *Specificity Damaged* and the others as *Specificity Preserved* on CounterFact-1k. We then measured the *Consistency Drop* (pre-edit minus post-edit consistency) for each group.

Editing Method	Specificity Preserved Samples (%)	Consistency Drop (Preserved)	Specificity Damaged Samples (%)	Consistency Drop (Damaged)	Difference in Consistency Drop (Preserved – Damaged)	<i>p</i> -value
ROME	596 (59.6%)	16.8	404 (40.4%)	16.2	+0.6	0.792
MEMIT	933 (93.3%)	16.4	67 (6.7%)	17.0	−0.6	0.892
AlphaEdit	870 (87.0%)	20.7	130 (13.0%)	10.4	+10.4	0.004
UnKE	59 (5.9%)	18.9	941 (94.1%)	29.7	−10.9	0.003

Table 14: Consistency drops for samples with preserved versus damaged specificity. *p*-values are from two-sided Welch’s *t*-tests on per-sample consistency drops.

MEMIT showed indistinguishable consistency drops between the two groups (16.4 vs. 17.0; $p = 0.892$), as did ROME (16.8 vs. 16.2; $p = 0.792$). We therefore find no evidence that their consistency degradation is introduced specifically by preserving locality. The interaction is method-dependent: AlphaEdit showed a larger drop when specificity was preserved, whereas UnKE showed the opposite pattern.