

P-Space: Puncturing the Null-Space to Resolve Cognitive Dissonance in Continual Knowledge Editing

Shiqiang Tian¹, Cheng Ding¹, Qin Chen^{1*}, Jie Zhou¹, Xin Li², Liang He¹

¹School of Computer Science and Technology, East China Normal University

²Shanghai Artificial Intelligence Laboratory

{sqtian, cding}@stu.ecnu.edu.cn, {qchen, jzhou, lhe}@cs.ecnu.edu.cn

lixin@pjlab.org.cn

Abstract

Existing continual knowledge editing methods suffer from cognitive dissonance: after an edit, the model answers the edit prompt correctly but reverts to the old fact under complex equivalent queries. This means old and new facts coexist rather than the old being replaced. The root cause of cognitive dissonance is that existing methods fit the new fact on the target prompt without overwriting the old one. In constrained methods, static null-space projectors lock outdated representations into the protected subspace, forcing updates into orthogonal directions. To resolve cognitive dissonance, we propose **P-Space** (*Puncturing the Null-Space to Resolve Cognitive Dissonance in Continual Knowledge Editing*), which at each edit step (1) punctures the protected covariance to unprotect the outdated representation, (2) overwrites the old mapping in closed form along its original direction under the punctured projector, and (3) anchors the new fact into the covariance with adaptive scaling so that future edits preserve it. Experiments on two datasets show that P-Space outperforms existing methods while preserving general capabilities.

1 Introduction

Large Language Models (LLMs) acquire knowledge from large scale pretraining corpora (Roberts et al., 2020; Yao et al., 2023; Zhuang et al., 2025), and much of this knowledge is encoded in their parameters (Brown et al., 2020; Ji et al., 2023; Chen et al., 2026). Since these facts can become outdated as the world changes (Huang et al., 2024), knowledge editing aims to update facts in model weights without full retraining (Meng et al., 2022, 2023; Wang et al., 2024; Hong and Lipani, 2024). Early methods, such as ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023), work well for single edits, but they often fail under sequential editing, where repeated updates shift the model distribution

*Corresponding author.

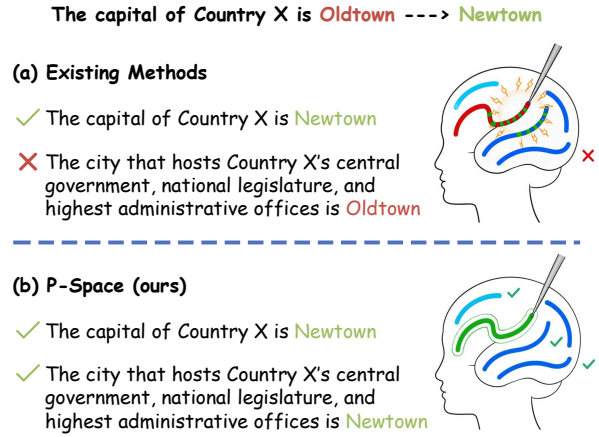


Figure 1: Cognitive dissonance in continual knowledge editing. After editing “the capital of Country X” to Newtown, existing methods still output Oldtown for a complex equivalent prompt. P-Space overwrites the old fact, so both prompts output Newtown.

and cause catastrophic forgetting (Hu et al., 2024; Gu et al., 2024; Cao et al., 2026). To reduce this interference, constrained methods limit how much the weights can move: RECT (Gu et al., 2024) regularizes updates, PRUNE (Ma et al., 2025) imposes condition number restraints on edited matrices, and AlphaEdit (Fang et al., 2025) projects updates onto the null-space of preserved knowledge.

These methods improve stability, but they still share a failure mode that we call cognitive dissonance. Figure 1 gives an example. Suppose an edit changes the capital of Country X from Oldtown to Newtown. After editing, the model answers the direct prompt correctly: “The capital of Country X is Newtown.” However, when asked a longer but equivalent query, such as “the city that hosts Country X’s central government, national legislature, and highest administrative offices,” it still outputs Oldtown. The edit changes one access route, while another route still leads to the old answer, so the

model holds two associations for the same fact at the same time. This shows that the edit has only fitted the new prompt but has not overwritten the outdated knowledge under complex queries.

The root cause is that existing methods optimize edit success on the target prompt, but they do not explicitly overwrite outdated knowledge (Fang et al., 2025). Unconstrained methods may damage other knowledge after many edits. Constrained methods can cause cognitive dissonance through the over protection paradox: fixed constraints protect outdated knowledge together with knowledge that should be preserved. For example, a null-space projector built from a global covariance matrix may include the outdated activation k_{old} and suppress updates along the direction that should be changed. As a result, the new fact is written into other directions instead of replacing the old one. This is orthogonal patching: old and new facts stay in different directions, so the model may give different answers to complex prompts.

To address this problem, we propose **P-Space** (*Puncturing the Null-Space to Resolve Cognitive Dissonance in Continual Knowledge Editing*), a three-stage process that punctures, overwrites, and consolidates knowledge in place. First, Targeted Subspace Puncturing (TSP) removes protection from the outdated knowledge representation by bandpass spectral filtering, while keeping the main covariance structure. Next, Directed Semantic Overwriting (DSO) computes a closed-form update under the punctured projector and writes the new fact along the original knowledge direction, so the old route is overwritten rather than bypassed. Finally, Adaptive Covariance Anchoring (ACA) adds the new knowledge back into the covariance, so future edits treat it as protected. As shown in Figure 1, P-Space builds an edit-specific projector that overwrites the old fact in place, so both the direct prompt and complex equivalent prompts output the new fact while unrelated knowledge is preserved.

We evaluate P-Space on LLaMA-3-8B¹ and Mistral-7B-v0.3 (Jiang et al., 2023) with ZsRE (Levy et al., 2017) and FEVER (Thorne et al., 2018), following the standard continual editing protocols of prior work (Fang et al., 2025; Li et al., 2025). P-Space consistently achieves the best average scores among all baselines, reaching 85.16 on FEVER and 65.09 on ZsRE. The main contribu-

tions of this work are:

- We identify cognitive dissonance in continual knowledge editing, due to the lack of an explicit mechanism to overwrite an outdated fact while strictly preserving unrelated knowledge.
- We propose P-Space, a framework that punctures, overwrites, and consolidates knowledge in place while preserving unrelated knowledge.
- Experiments on two LLMs and two benchmarks show that P-Space achieves the best average scores on FEVER and ZsRE while preserving general capabilities over long edit sequences.

2 Related Work

Knowledge editing updates stored factual knowledge while preserving unrelated behavior. Existing methods are commonly grouped into parameter-modifying and parameter-preserving approaches (Yao et al., 2023).

2.1 Parameter-Modifying Methods

Parameter-modifying methods alter model weights to encode the edited fact. Fine-tuning (Zhu et al., 2020) applies an autoregressive objective to edit examples, but its updates are difficult to localize and can perturb unrelated knowledge. Locate-and-edit methods identify factual storage locations before changing parameters. ROME (Meng et al., 2022) edits factual associations through selected feedforward weights, and MEMIT (Meng et al., 2023) extends ROME’s modeling formulation with a least squares approximation that directly updates parameters at selected layers, enabling multilayer editing and simultaneous updates to hundreds or thousands of facts. When applied repeatedly, these direct updates can accumulate interference and cause forgetting (Gu et al., 2024). Constrained methods reduce this risk by restricting the update space: RECT (Gu et al., 2024) regularizes weight updates, PRUNE (Ma et al., 2025) imposes conditional restraints on edited matrices, and AlphaEdit (Fang et al., 2025) projects updates onto a null-space constructed from preserved knowledge keys. However, a fixed constraint can also protect outdated knowledge together with unrelated knowledge, which can lead to cognitive dissonance.

Another line trains editors that emit parameter updates at test time. KE (De Cao et al., 2021) predicts the weight shift directly from each edit example. MEND (Mitchell et al., 2022a) factorizes

¹<https://www.llama.com/models/llama-3/>

fine-tuning gradients into low-rank components and routes them through a hypernetwork to obtain a localized update. MALMEN (Tan et al., 2024) uses a least squares formulation to aggregate parameter shifts over large edit batches and obtains the optimal parameter shifts by solving the corresponding normal equations. DAFNet (Zhang et al., 2024) introduces a Dynamic Auxiliary Fusion Network that mixes information among knowledge triples within a sequential edit stream to enable continuous mistake rectification. RLEdit (Li et al., 2025) casts continual editing as an RL task and trains an editor that balances edit success at each step with the durability of past edits across sequences. TamEdit (Tian et al., 2026) uses trajectory-aware meta-learning to preserve specificity across continual edits. These editors improve edit success or specificity, but none provides an explicit mechanism to overwrite an outdated fact while strictly preserving unrelated knowledge.

2.2 Parameter-Preserving Methods

Parameter-preserving methods leave the base weights untouched and route edited knowledge through external modules, memories, or prompts. SERAC (Mitchell et al., 2022b) pairs a counterfactual model with a scope classifier that decides, per query, whether the edited model or the original model should respond. T-Patcher (Huang et al., 2023) attaches a small set of patch neurons to redirect specific outputs. GRACE (Hartvigsen et al., 2023) maintains a key-value codebook in latent space that is queried at inference time, and MELO (Yu et al., 2024) routes computation through dynamically activated LoRA blocks indexed by retrieved edits. MemPrompt (Madaan et al., 2022) and IKE (Zheng et al., 2023) realize editing in context by prepending retrieved demonstrations of the edited fact to the prompt. The base parameters stay intact, but the external store or retrieval index grows with the number of edits, so cost scales roughly linearly in the edit set size.

In contrast, our method changes the protected subspace for each edit rather than keeping it fixed. This allows the model to overwrite outdated knowledge in its original direction while still preserving unrelated knowledge.

3 Background and Problem Formulation

3.1 Target Modules

We edit transformer FFN modules, where prior work locates factual associations. At layer l , the FFN output projection W_{out}^l maps a key to an output value m^l . With hidden state h^{l-1} , attention output a^l , layer normalization $\gamma(\cdot)$, input projection W_{in}^l , and activation $\sigma(\cdot)$, this mapping is

$$m^l = W_{\text{out}}^l \sigma(W_{\text{in}}^l \gamma(h^{l-1} + a^l)). \quad (1)$$

Following the key-value view of MLP editing (Meng et al., 2022, 2023), we treat the activation before W_{out}^l as the key and the FFN output as the value. We denote the target output projection by W , the key from query x by k_x , and the weight update by ΔW .

Editing an outdated fact replaces $k_{\text{old}} \mapsto v_{\text{old}}$ with $(k_{\text{new}}, v_{\text{new}})$, where k_{old} and k_{new} come from the old and new queries. We use this notation to define the editing goals below.

3.2 Cognitive Dissonance

Let an edit replace an outdated fact $k_{\text{old}} \mapsto v_{\text{old}}$ with a new target $(k_{\text{new}}, v_{\text{new}})$. Let $\mathcal{X}_{\text{old}}^{\text{eq}}$ denote paraphrases, entailment variants, or complex equivalent prompts that should access the same outdated knowledge. We define cognitive dissonance as the case where the edited model satisfies edit success,

$$(W + \Delta W)k_{\text{new}} \approx v_{\text{new}} \quad (2)$$

for the exact edit prompt, yet still produces the outdated value on some complex equivalent prompts:

$$(W + \Delta W)k_x \approx v_{\text{old}}, \quad \exists x \in \mathcal{X}_{\text{old}}^{\text{eq}}. \quad (3)$$

Thus, the target edit succeeds, but complex equivalent queries fail. The old and new mappings co-exist inside the model. Standard paraphrase failure shows that the new target does not transfer to another phrasing; cognitive dissonance is stricter: the direct edit succeeds, yet a semantically equivalent query recovers the pre-edit answer. It differs from consistency propagation, ripple effects, and related-fact updating, which test other logically dependent facts, and from multi-hop editing, which tests compositional use of the edited fact.

Across method families, the common issue is the lack of an explicit mechanism to overwrite an outdated fact while strictly preserving unrelated

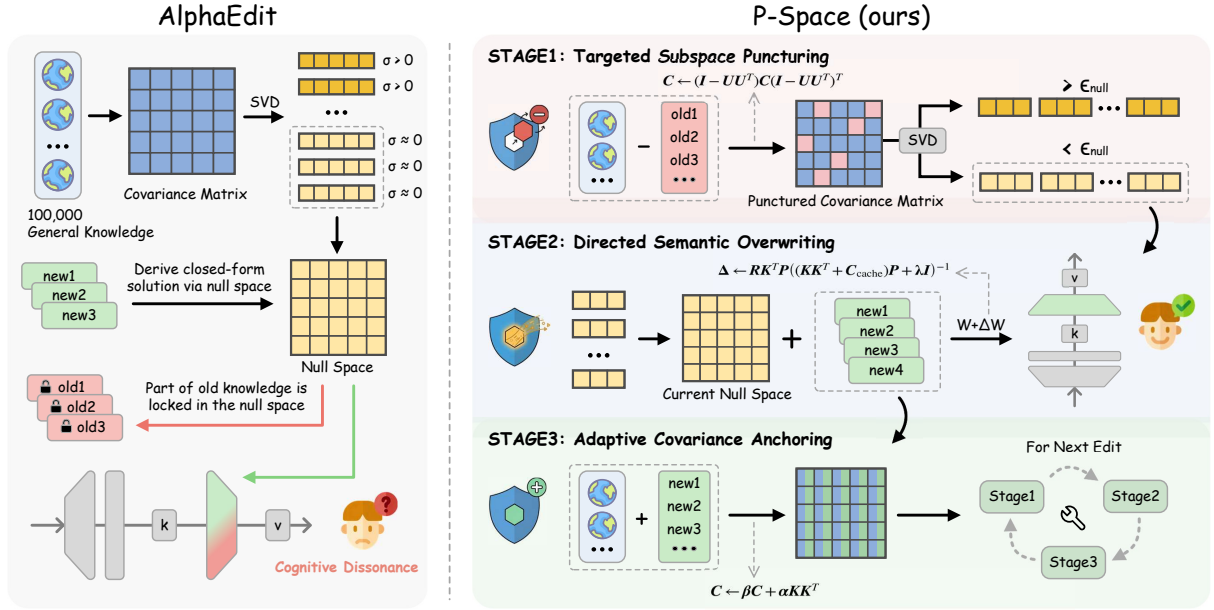


Figure 2: Overall framework of P-Space. Left: a static null-space projector locks outdated knowledge into the protected subspace, forcing updates into orthogonal directions and causing cognitive dissonance. Right: P-Space resolves this through three stages. TSP removes protection only from the outdated fact specific band. DSO writes the new fact along the original knowledge route through the punctured projector. ACA adds the new fact back into the covariance so future edits preserve it.

knowledge. This issue is clear in static constrained methods: the preserved key space may also contain semantic directions of outdated facts. We call this the over protection paradox. Following AlphaEdit (Fang et al., 2025), let C_{total} be the covariance matrix estimated from a large set of preserved knowledge keys. We apply SVD to this covariance, $\{U, \Lambda, U^T\} = \text{SVD}(C_{\text{total}})$, where U contains eigenvectors and Λ contains eigenvalues. Let $\hat{U}$ be the submatrix of U formed by the eigenvectors whose eigenvalues are zero or close to zero, yielding the static update projector $\mathcal{P}_0 = \hat{U}\hat{U}^T$. AlphaEdit shows that this projector removes preserved key directions, so for preserved or general knowledge keys k_g we have $\mathcal{P}_0 k_g \approx 0$. Estimated from broad knowledge, this space can also include directions of outdated knowledge. These directions are protected with general knowledge, making $\mathcal{P}_0 k_{\text{old}}$ close to zero. Thus, the projected update leaves the old route nearly unchanged:

$$(W + \Delta W \mathcal{P}_0) k_{\text{old}} \approx W k_{\text{old}}. \quad (4)$$

The projected update cannot directly change the old knowledge representation, so it writes the new fact in other directions. The target prompt may be corrected, while queries routed through k_{old} can still activate outdated knowledge. Cognitive dissonance is the mismatch between target edit success

and overwrite along the original knowledge route.

3.3 Problem Formulation

Continual editing requires three goals. First, the update ΔW should achieve edit success as defined in Eq. (2). Second, it should overwrite in place by mapping complex equivalent prompts of the outdated knowledge to the new value:

$$(W + \Delta W) k_x \approx v_{\text{new}}, \quad \forall x \in \mathcal{X}_{\text{old}}^{\text{eq}}. \quad (5)$$

This separates overwriting from orthogonal patching: the new knowledge should replace the old mapping along the original knowledge representation, not only fit the exact edit prompt. Third, it should preserve unrelated knowledge. For queries outside the edited fact and its complex equivalent prompts, behavior should remain stable:

$$(W + \Delta W) k_x \approx W k_x, \quad \forall x \in \mathcal{X}_{\text{preserve}}. \quad (6)$$

Existing methods fail on all three goals: unconstrained methods damage unrelated knowledge, static constrained methods suppress k_{old} , and learned editors neither explicitly overwrite the old route nor guarantee global preservation. This motivates our central idea: the protected subspace should depend on the edit, not remain fixed.

4 P-Space

To overwrite an outdated fact without damaging unrelated knowledge, P-Space proceeds in three stages: Targeted Subspace Puncturing makes the outdated route editable, Directed Semantic Overwriting replaces the old mapping in place, and Adaptive Covariance Anchoring protects the new fact from being overwritten by future edits, as illustrated in Figure 2.

4.1 Targeted Subspace Puncturing (TSP)

The covariance used for preservation also contains the fact that now needs to change. If we build a projector from this covariance directly, the old fact is protected together with unrelated knowledge, and the update cannot move along the old route. TSP addresses this by removing protection only from a selected part of the old fact subspace.

For layer l , let M_l be the second moment accumulator and N_l its normalizer, so the protected covariance is $\bar{C}_l = M_l/N_l$. Let K_{old} stack the keys of the old fact and its equivalent queries. Since datasets do not provide these queries, we synthesize them from each edit item’s source prompt, subject, and pre-edit target (Appendix A.3).

These keys contain the outdated fact route, but they also contain shared language and task directions. Directly subtracting $K_{\text{old}}K_{\text{old}}^T$ from the covariance is too coarse: it may open the old route but also weaken useful protection. Instead, TSP selects a fact-related band from the spectrum of K_{old} . We decompose K_{old} by thin SVD:

$$\begin{aligned} K_{\text{old}} &= USV^T, \\ S &= \text{diag}(s_1, \dots, s_r). \end{aligned} \quad (7)$$

We select a middle band $U_B = U_{:,k_h:k_t}$ by cumulative singular value energy.

We then puncture the covariance by projecting away this band:

$$\begin{aligned} \Pi_{\perp} &= I - U_B U_B^T, \\ C_{\text{edit}} &= \Pi_{\perp} \bar{C}_l \Pi_{\perp}. \end{aligned} \quad (8)$$

Directions outside U_B retain the original covariance, so unrelated knowledge remains protected. Directions inside U_B become editable for this step. From the punctured covariance, we compute the current projector:

$$\mathcal{P}_{\text{curr}} = \sum_{i: \sigma_i < \epsilon_{\text{null}}} u_i u_i^T, \quad (9)$$

where $C_{\text{edit}} = \sum_i \sigma_i u_i u_i^T$ and ϵ_{null} defines the null-space threshold. This projector is temporary and is used only for the current edit. We set the head-energy ratio to $\tau_h = 0.1$, the band-energy ratio to $\tau_m = 0.7$, and the null-space threshold to $\epsilon_{\text{null}} = 2 \times 10^{-3}$. Sensitivity results are reported in Appendix D.1.

The punctured projector satisfies two properties, proved in Appendix E. First, $\mathcal{P}_{\text{curr}} K_{\text{old}} \neq 0$: since U_B directions have zero eigenvalue in C_{edit} , they enter the column space of $\mathcal{P}_{\text{curr}}$, and K_{old} has nonzero energy in $\text{span}(U_B)$ by construction. Second, $\mathcal{P}_{\text{curr}} K_p \approx 0$: historical keys K_p come from different facts whose directions lie mostly outside $\text{span}(U_B)$, so their covariance energy is unchanged and they remain in the non-null eigenspace of C_{edit} . Each edit thus receives its own projector: the outdated fact becomes editable, previous edits remain protected, and DSO can overwrite the old route instead of writing an orthogonal patch.

4.2 Directed Semantic Overwriting (DSO)

After TSP, we write the new fact through the punctured projector $\mathcal{P}_{\text{curr}}$ using the projection-based editing formulation of AlphaEdit (Fang et al., 2025). Here K_{new} and V_{new} denote the keys and targets for the new fact and its equivalent queries, K_p collects the historical keys from previous edits, and $R = V_{\text{new}} - W K_{\text{new}}$ is the target residual. We optimize an update ΔW , but only its component projected on the right is written to the model. We denote this effective update for the current step by

$$\Delta W_{\text{curr}} = \Delta W \mathcal{P}_{\text{curr}}. \quad (10)$$

Directed Semantic Overwriting (DSO) solves the following least squares problem:

$$\begin{aligned} \Delta W^* &= \arg \min_{\Delta W} \left\| \Delta W_{\text{curr}} K_{\text{new}} - R \right\|_F^2 \\ &\quad + \left\| \Delta W_{\text{curr}} K_p \right\|_F^2 + \lambda \left\| \Delta W_{\text{curr}} \right\|_F^2. \end{aligned} \quad (11)$$

The effective update is $\Delta W_{\text{curr}} = \Delta W^* \mathcal{P}_{\text{curr}}$, whose closed form is

$$\begin{aligned} \Delta W_{\text{curr}} &= R K_{\text{new}}^T \mathcal{P}_{\text{curr}} \\ &\quad (K_{\text{new}} K_{\text{new}}^T \mathcal{P}_{\text{curr}} + K_p K_p^T \mathcal{P}_{\text{curr}} + \lambda I)^{-1}. \end{aligned} \quad (12)$$

λ is used for stable inversion, and we update $W = W + \Delta W_{\text{curr}}$. The key difference from AlphaEdit is the projector: P-Space uses the edit-specific punctured projector $\mathcal{P}_{\text{curr}}$ instead of a static one. As a result, because $\mathcal{P}_{\text{curr}} K_{\text{old}} \neq 0$, the update can now move along the old route and replace the outdated mapping in place.

Table 1: Comparison of P-Space and baseline methods on continual editing tasks. Avg is computed by averaging all six metrics. We use boldface to highlight the best value in each column. [†]Results are taken from RLEdit (Li et al., 2025) under the matched setup. Zero entries are genuine Top-1 results after long continual editing, not missing values.

FEVER							
Methods	LLaMA-3-8B			Mistral-7B-v0.3			Avg [†]
	Eff.	Gen.	Spe.	Eff.	Gen.	Spe.	
FT [†]	1.80 ±0.10	6.39 ±0.17	26.33 ±0.13	17.08 ±0.24	21.57 ±0.30	37.47 ±0.32	18.44
ROME [†]	38.56 ±0.28	44.53 ±0.29	9.29 ±0.17	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	15.40
MEMIT [†]	0.18 ±0.02	0.01 ±0.01	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.03
PRUNE [†]	56.64 ±0.24	43.31 ±0.18	0.85 ±0.05	5.27 ±0.34	3.11 ±0.16	5.89 ±0.23	19.18
RECT [†]	60.95 ±0.27	52.40 ±0.26	1.75 ±0.07	0.55 ±0.04	0.05 ±0.01	0.00 ±0.00	19.28
AlphaEdit [†]	94.22 ±0.25	94.14 ±0.18	25.57 ±0.14	32.74 ±0.42	30.03 ±0.29	8.44 ±0.45	47.52
MEND [†]	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00
MEND* [†]	10.37 ±0.19	9.34 ±0.22	4.88 ±0.24	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	4.10
MALMEN [†]	0.01 ±0.01	0.01 ±0.01	0.14 ±0.03	16.67 ±0.21	16.61 ±0.18	12.16 ±0.16	7.60
MALMEN* [†]	5.74 ±0.20	5.29 ±0.13	1.26 ±0.18	17.73 ±0.24	13.30 ±0.27	13.85 ±0.21	9.53
DAFNet [†]	31.27 ±0.47	28.82 ±0.43	66.55 ±0.41	4.86 ±0.14	4.21 ±0.19	41.71 ±0.48	29.57
RLEdit [†]	95.34 ±0.34	93.58 ±0.38	70.36 ±0.29	88.99 ±0.22	88.25 ±0.25	73.64 ±0.11	85.03
P-Space	96.14 ±0.21	94.95 ±0.38	69.46 ±0.57	90.21 ±0.13	88.92 ±0.35	71.29 ±0.39	85.16

ZsRE							
Methods	LLaMA-3-8B			Mistral-7B-v0.3			Avg [†]
	Eff.	Gen.	Spe.	Eff.	Gen.	Spe.	
FT [†]	17.10 ±0.22	16.73 ±0.22	8.27 ±0.13	32.84 ±0.30	33.78 ±0.30	42.19 ±0.31	25.15
ROME [†]	0.54 ±0.04	0.57 ±0.04	0.40 ±0.02	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.25
MEMIT [†]	0.00 ±0.00	0.00 ±0.00	0.13 ±0.02	0.00 ±0.00	0.00 ±0.00	0.13 ±0.02	0.04
PRUNE [†]	12.27 ±0.43	12.01 ±0.23	9.88 ±0.29	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	5.69
RECT [†]	11.05 ±0.41	8.12 ±0.15	28.12 ±0.13	8.18 ±0.33	8.04 ±0.46	11.32 ±0.49	12.47
AlphaEdit [†]	86.83 ±0.23	81.48 ±0.28	29.09 ±0.22	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	32.90
MEND [†]	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00
MEND* [†]	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	0.00
MALMEN [†]	9.87 ±0.12	9.00 ±0.09	2.11 ±0.15	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00	3.50
MALMEN* [†]	12.23 ±0.11	11.08 ±0.22	2.43 ±0.09	0.01 ±0.01	0.02 ±0.01	1.25 ±0.04	4.50
DAFNet [†]	21.99 ±0.47	11.17 ±0.43	32.21 ±0.39	1.25 ±0.08	2.12 ±0.12	25.27 ±0.54	15.67
RLEdit [†]	89.42 ±0.34	87.32 ±0.23	44.78 ±0.50	71.12 ±0.31	67.42 ±0.27	27.43 ±0.44	64.58
P-Space	89.81 ±0.42	86.39 ±0.41	46.61 ±0.49	70.29 ±0.38	67.24 ±0.46	30.18 ±0.51	65.09

4.3 Adaptive Covariance Anchoring (ACA)

After DSO, to further protect the new fact in future edits, Adaptive Covariance Anchoring (ACA) adds it back into the protected covariance $\bar{C}_l$. Otherwise, later edits may overwrite it again. Since $\bar{C}_l$ is stored through the pair (M_l, N_l) , ACA updates this accumulator and its normalizer.

A direct update such as $M_l = M_l + K_{\text{new}} K_{\text{new}}^T$ can be too weak because M_l is initialized from many prior knowledge keys and already carries historically large covariance mass. As a result, the new fact may receive little protection compared with the existing covariance. ACA therefore up-

dates M_l with a controlled scale:

$$\begin{aligned} M_l &= \beta M_l + \alpha K_{\text{new}} K_{\text{new}}^T, \\ N_l &= \beta N_l + \alpha, \end{aligned} \quad (13)$$

$\beta \leq 1$ slows long-term drift in the accumulator. The scalar α controls the protection added for the new fact. After the update, the protected covariance for later edits is still computed as $\bar{C}_l = M_l / N_l$. ACA chooses α by checking how much covariance energy the new key matrix K_{new} already has in M_l .

Let $\lambda_K = \|K_{\text{new}} K_{\text{new}}^T\|_2$ measure the scale of the new key covariance. Let λ_* be the desired target protection level, and let $E_{\text{raw}} = \text{Tr}(M_l K_{\text{new}} K_{\text{new}}^T)$ be the current raw covariance energy along the new

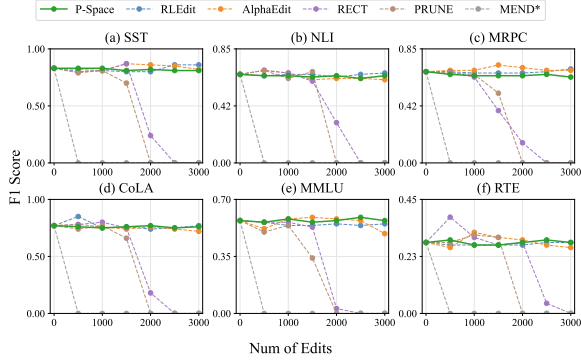


Figure 3: General capability retention across continual editing steps. P-Space maintains stable performance across SST, NLI, MRPC, CoLA, MMLU, and RTE, whereas MEND*, RECT, and PRUNE show sharp degradation under repeated edits.

key directions. We set

$$\alpha = \max \left(0, \frac{\beta (\lambda_{\star} N_l n \lambda_K - E_{\text{raw}})}{n (\lambda_K^2 - \lambda_{\star} \lambda_K + \epsilon)} \right), \quad (14)$$

n is the number of keys in K_{new} and ϵ is a small constant for stable division. The closed-form derivation of Eq. (14) is given in Appendix E. If the new key directions already have enough covariance energy, ACA adds little extra protection. If they are underrepresented, ACA increases α and raises them toward the target level.

ACA completes the editing cycle. TSP opens the old route, DSO overwrites it, and ACA protects the new route for later edits.

5 Experiments

5.1 Experimental Setup

We follow standard continual editing protocols and evaluation metrics used in prior model editing benchmarks (Fang et al., 2025; Li et al., 2025). We report main continual editing results on FEVER (Thorne et al., 2018) and ZsRE (Levy et al., 2017), and use both ZsRE and CounterFact (Meng et al., 2022) for ablation analysis and editing scale studies. We evaluate on LLaMA-3-8B and Mistral-7B-v0.3 with continual streams (e.g., 20×100 , 50×100 , 100×100 , and beyond). Dataset details and metric definitions are provided in Appendix A.2 and A.5, respectively.

Baselines. We compare against representative methods (details in Appendix A.1): FT (Zhu et al., 2020), ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), PRUNE (Ma et al., 2025), RECT

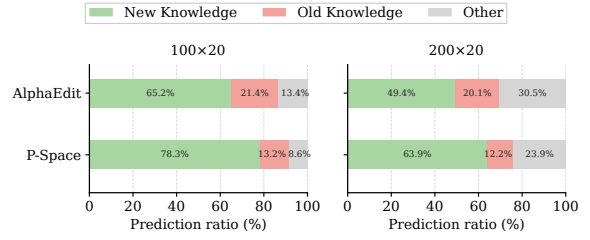


Figure 4: Prediction ratios on ZsRE complex equivalent prompts under continual editing. P-Space yields more New Knowledge and fewer Old Knowledge outputs than AlphaEdit in both settings.

(Gu et al., 2024), AlphaEdit (Fang et al., 2025), MEND (Mitchell et al., 2022a), MALMEN (Tan et al., 2024), DAFNet (Zhang et al., 2024), and RLEdit (Li et al., 2025). Following Li et al. (2025), MEND* and MALMEN* denote variants that re-train their hypernetworks during continual editing to reduce the mismatch between the initial editor and edited model states.

Metrics. We report continual editing metrics Efficacy (Eff.), Generalization (Gen.), and Specificity (Spe.), following prior work (Meng et al., 2022; Mitchell et al., 2022a; Fang et al., 2025). For general capability evaluation, we report General Capability (GC), defined as average F1 over the six GLUE tasks listed in Appendix A.6.

5.2 Main Results

Table 1 reports continual editing results on FEVER and ZsRE. P-Space attains the highest average score on both datasets, reaching 85.16 on FEVER and 65.09 on ZsRE, with gains of 0.13 and 0.51 points over the strongest baseline RLEdit.

The improvements come from different axes on the two datasets. On FEVER, P-Space attains the best Eff. and Gen. on both LLaMA-3-8B and Mistral-7B-v0.3, outperforming RLEdit in Gen. The higher Gen. indicates that equivalent prompts of the edited fact are also mapped to the new value, rather than the edit being effective only on the exact prompt. On ZsRE, P-Space attains the best Spe. on LLaMA-3-8B and competitive Gen. on both models, showing that the overwrite does not come at the cost of unrelated knowledge.

Several baselines degrade sharply under continual editing, either failing on one model or retaining edit success at the cost of specificity. AlphaEdit performs strongly on LLaMA-3-8B for FEVER but its specificity is markedly lower than P-Space, and it also drops on Mistral-7B-v0.3. P-Space, in

Table 2: Ablation study on LLaMA-3-8B with CounterFact under the 20×100 setting, evaluating the contributions of Targeted Subspace Puncturing (TSP) and Adaptive Covariance Anchoring (ACA).

Edit Steps	Variant	Eff. ($\uparrow$)	Gen. ($\uparrow$)	Spe. ($\uparrow$)
20×100	Full P-Space	98.71	95.22	77.82
	w/o TSP	98.90 $\uparrow 0.19$	90.40 $\downarrow 4.82$	79.46 $\uparrow 1.64$
	w/o ACA	99.36 $\uparrow 0.65$	95.15 $\downarrow 0.07$	68.72 $\downarrow 9.10$
	w/o TSP & ACA	98.95 $\uparrow 0.24$	94.78 $\downarrow 0.44$	68.56 $\downarrow 9.26$

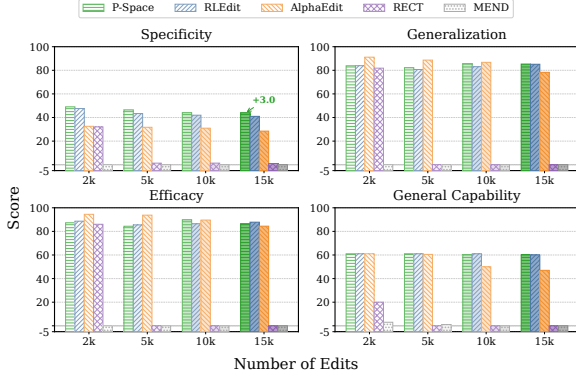


Figure 5: Performance comparison across editing scales. Darker colors indicate more edits. P-Space achieves the strongest specificity while maintaining competitive efficacy, generalization, and general capability.

contrast, remains stable across models and datasets.

5.3 Ablation Study

We conduct ablations on LLaMA-3-8B with CounterFact under the 20×100 editing setting to isolate the contribution of each stage. Table 2 reports the results. We test three P-Space variants: removing TSP alone, removing ACA alone, and removing both stages together. Removing TSP lowers Gen. from 95.22 to 90.40, while Eff. and Spe. stay stable. This indicates that TSP is responsible for helping the edit generalize to equivalent prompts. Removing ACA, on the other hand, lowers Spe. from 77.82 to 68.72, whereas Gen. remains close to the full model. When both stages are removed at the same time, Spe. stays close to the w/o ACA setting, while Gen. stays near the full model. These results confirm that TSP and ACA serve complementary roles. TSP mainly drives edit generalization, whereas ACA mainly guards long-term specificity.

5.4 General Capability Retention

Figure 3 shows that P-Space maintains stable core capabilities across continual edits on SST, NLI, MRPC, CoLA, MMLU, and RTE. Across 3000 continual edits, all six scores stay close to their initial values, showing limited task-level drift. In

contrast, MEND*, RECT, and PRUNE collapse to near zero on several tasks at later edit counts. This gap therefore shows that high edit success alone is not enough, because long edit streams also test retained general behavior. A useful editing method must also keep the model’s general behavior stable after many updates. P-Space achieves this because each new fact is written through the punctured route and then anchored back into the covariance, rather than repeatedly shifting broad model behavior. These results indicate that P-Space preserves general capability while absorbing continual knowledge updates.

5.5 Robustness against Complex Contexts

Simple paraphrases can miss failures on complex equivalent prompts, where edited models may revert to obsolete knowledge. This setting is closer to cognitive dissonance than standard paraphrase tests, because the query can still follow the old knowledge route. We evaluate ZsRE under 100×20 and 200×20 continual edits. Figure 4 shows that P-Space raises New Knowledge and lowers Old Knowledge at both edit scales. Taken together, these results show that P-Space not only fits the direct edit prompt but also reduces outdated outputs under harder complex equivalent contexts. The detailed results are provided in Appendix D.

5.6 Editing Scale

We evaluate robustness as continual edits increase from 2k to 15k. This setting tests whether editing quality degrades as updates accumulate. Figure 5 shows that P-Space achieves the best Specificity at both 2k and 15k, with scores of 49.01 and 43.93, respectively, while General Capability stays near 61.0. This trend indicates that puncturing only the edit-related route limits interference with unrelated knowledge. Meanwhile, Efficacy and Generalization remain close to the strongest baselines, indicating that higher stability does not reduce edit strength. Thus, P-Space scales better to longer continual edit streams.

6 Conclusion

We introduced P-Space, which makes the outdated fact route editable through TSP, overwrites it with DSO, and protects the new association through ACA. Across FEVER and ZsRE on LLaMA-3-8B and Mistral-7B-v0.3, P-Space achieves the best average results while preserving unrelated knowledge. The results further show that stable edit success alone is insufficient: several baselines retain performance early in an edit stream but lose specificity or general capability as edits accumulate. By selectively reopening the outdated route rather than broadly relaxing preservation, P-Space maintains strong specificity and stable general capability up to 15,000 edits. Future work can test whether edit-specific preservation extends to more model families, more complex factual relations, and realistic streams of evolving knowledge.

Limitations

Although P-Space shows strong performance in continual knowledge editing, several broader challenges remain for the field. Current benchmarks still provide only partial coverage of real world knowledge dynamics, especially for complex reasoning, temporally evolving facts, and open-ended user interactions. The field also lacks more challenging and more complex benchmarks that can stress-test edited models under realistic use cases. In addition, evaluating long horizon reliability, safety, and efficiency at larger editing scales remains an important direction for future work. In deployment, incorrect or malicious edits could introduce false information, while overly broad updates could erase useful related knowledge. Edited facts should be audited over time using edit logs and periodic re-evaluation as basic safeguards. These risks require more comprehensive benchmarks and deployment-oriented evaluation protocols for LLM knowledge editing.

Acknowledgments

This research is funded by the National Nature Science Foundation of China (No.62477010, No.62577022 and No.62307028), Shanghai General AI Foundation Models Program (No.2025SHZDZX026D24) and Shanghai Science and Technology Innovation Action Plan (No.24YF2710100).

References

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Ding Cao, Yuchen Cai, Yuqing Huang, Xuesong He, Rongxi Guo, Guiquan Liu, and Guangzhong Sun. 2026. On the superimposed noise accumulation problem in sequential knowledge editing of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30139–30147.
- Kedi Chen, Dezhao Ruan, Yuhao Dan, Yaoting Wang, Siyu Yan, Xuecheng Wu, Yinqi Zhang, Qin Chen, Jie Zhou, Liang He, Biqing Qi, Linyang Li, Qipeng Guo, Xiaoming Shi, and Wei Zhang. 2026. *A survey of inductive reasoning for large language models*. Preprint, arXiv:2510.10182.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. *Editing factual knowledge in language models*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- 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 *International Conference on Learning Representations*, volume 2025, pages 16366–16396.
- Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. 2024. *Model editing harms general abilities of large language models: Regularization to the rescue*. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16801–16819, Miami, Florida, USA. Association for Computational Linguistics.
- Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959.
- Yihuai Hong and Aldo Lipani. 2024. Interpretability-based tailored knowledge editing in transformers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3847–3858.
- Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. *WiKE: Wise-layer knowledge editor for lifelong knowledge editing*. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3476–3503, Bangkok, Thailand. Association for Computational Linguistics.

- Xiusheng Huang, Yequan Wang, Jun Zhao, and Kang Liu. 2024. Commonsense knowledge editing based on free-text in llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14870–14880.
- Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations*.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e 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.
- Zherui Li, Houcheng Jiang, Hao Chen, Baolong Bi, Zhenhong Zhou, Fei Sun, Junfeng Fang, and Xiang Wang. 2025. *Reinforced lifelong editing for language models*. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 34920–34942. PMLR.
- Jun-Yu Ma, Hong Wang, Hao-Xiang Xu, Zhen-Hua Ling, and Jia-Chen Gu. 2025. *Perturbation-restrained sequential model editing*. In *International Conference on Learning Representations*, volume 2025, pages 17302–17329.
- Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.
- Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. *Mass-editing memory in a transformer*. In *The Eleventh International Conference on Learning Representations*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. *Fast model editing at scale*. In *International Conference on Learning Representations*.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR.
- Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. *How much knowledge can you pack into the parameters of a language model?* In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.
- Chenmian Tan, Ge Zhang, and Jie Fu. 2024. *Massive editing for large language models via meta learning*. In *The Twelfth International Conference on Learning Representations*.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. *FEVER: a large-scale dataset for fact extraction and VERification*. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.
- Shiqiang Tian, Cheng Ding, Qin Chen, Jie Zhou, and Liang He. 2026. *TamEdit: Trajectory-aware meta-learning for specificity-preserving continual knowledge editing*. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21417–21439, San Diego, California, United States. Association for Computational Linguistics.
- Xiaohan Wang, Shengyu Mao, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, Huajun Chen, and Ningyu Zhang. 2024. Editing conceptual knowledge for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 706–724.
- 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 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.
- Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19449–19457.
- Taolin Zhang, Qizhou Chen, Dongyang Li, Chengyu Wang, Xiaofeng He, Longtao Huang, Hui Xue, and

- Jun Huang. 2024. [DAFNet: Dynamic auxiliary fusion for sequential model editing in large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1588–1602, Bangkok, Thailand. 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 *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.
- 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*.
- Xinlin Zhuang, Jiahui Peng, Ren Ma, Yinfan Wang, Tianyi Bai, Xingjian Wei, Qiu Jiantao, Chi Zhang, Ying Qian, and Conghui He. 2025. [Meta-rater: A multi-dimensional data selection method for pre-training language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10856–10896, Vienna, Austria. Association for Computational Linguistics.

A Detailed Experimental Settings

A.1 Baseline Methods

The following baseline methods are adopted in this paper:

- **FT** (Zhu et al., 2020) updates a chosen subset of LLM layers with a standard autoregressive next-token loss. The layer choice and learning rate follow the original setup.
- **MEND** (Mitchell et al., 2022a) is a hypernetwork-based editor that decomposes the fine-tuning gradient of an edit example into low-rank factors and feeds them through a small auxiliary network to obtain a localized weight shift. The auxiliary network is trained on the full training split with the released hyperparameters. **MEND*** is the continual-editing variant: to alleviate the drift between the original hypernetwork and the post-edited model, the hypernetwork is re-trained periodically on recent post-edit checkpoints as in the reference implementation.
- **ROME** (Meng et al., 2022) performs single-fact updates inside the LLM. It first uses perturbation-based causal tracing to locate the MLP neuron activations responsible for a target factual association, and then rewrites the corresponding MLP output weights through a Lagrange-style closed-form correction. As ROME is a single-edit method, its original protocol is followed by applying it as a sequence of one-fact edits.
- **MEMIT** (Meng et al., 2023) generalizes ROME to many facts and many layers. It formulates the joint update across the chosen layers as a least-squares problem and obtains a closed-form solution, allowing thousands of associations to be inserted in one batch. Continual-editing performance is evaluated with the original hyperparameters.
- **MALMEN** (Tan et al., 2024) is also a hypernetwork-based editor, but it is targeted at massive editing. The hypernetwork emits a parameter shift for each individual fact, and the global update for a large batch is recovered by solving a normal equation that aggregates these per-fact shifts. It is trained on the full training split with the original hyperparameters. Following the same convention as MEND, **MALMEN*** denotes the continual-editing variant whose hypernetwork is re-trained periodically on post-edited model states.
- **DAFNet** (Zhang et al., 2024) is designed for sequential editing. It augments the hypernetwork with a Dynamic Auxiliary Fusion Network that lets information flow among knowledge triples within an edit stream, so that later edits can correct earlier mistakes through reinforced semantic interactions. It is trained on ZsRE and CounterFact with the released configuration.
- **PRUNE** (Ma et al., 2025) also targets sequential editing. It adds a condition-number restraint on the edited weight matrix, which bounds the perturbation that the latest edit can inject into previously stored knowledge and thus slows down the typical performance decay over long edit streams. Hyperparameters follow the original paper.
- **RECT** (Gu et al., 2024) studies regularization in continual editing. It penalizes the magnitude of weight updates during editing to suppress overfitting to the current edit and to keep the model’s general capabilities intact. The original hyperparameter setting is used.
- **AlphaEdit** (Fang et al., 2025) reduces knowledge disruption in continual editing by projecting each parameter update onto the null space of a preserved-knowledge key matrix before applying it, so that the projected update is orthogonal to the preserved-knowledge directions. Its original hyperparameter configuration is followed.
- **RLEdit** (Li et al., 2025) casts continual editing as a reinforcement-learning problem. The editor is trained over long edit sequences to trade off success on the current edit against the durability of past edits, which yields stronger specificity at comparable efficacy and generalization. The original hyperparameter configuration is used.

P-Space uses the same LLaMA-3-8B and Mistral-7B-v0.3 checkpoints, dataset splits and case IDs, sample order, batch partitioning, editing protocol, evaluation metrics, and evaluation schedule.

A.2 Datasets

The datasets employed in this paper are described below.

- **ZsRE** (Zero-shot Relation Extraction) (Levy et al., 2017) is a relation-style question-answering benchmark widely adopted by the model-editing community. Each instance bundles an edit query with its target answer, a set of back-translated paraphrases that act as equivalent neighbors for measuring generalization, and a natural unrelated question used to probe locality (specificity). With this layout, an edit can be assessed along three axes at once: whether it lands on the source query, whether it transfers to surface-form variants, and whether it leaves unrelated queries untouched.
- **CounterFact** (Meng et al., 2022) pits counterfactual edit targets against the originally factual statement, which makes it a stress test for overwriting strongly memorized associations. Its out-of-scope split is generated by swapping the subject for a different but semantically related entity that shares the same predicate, yielding neighborhood prompts for specificity. Each item additionally provides paraphrase prompts for generalization and generation prompts that, when relevant, are used to score fluency and consistency. CounterFact is used in this paper for the ablation study and the editing-scale analysis.
- **FEVER** (Fact Extraction and VERification) (Thorne et al., 2018) is a Wikipedia-grounded fact-verification benchmark. Its native label space is three-way (SUPPORTS, REFUTES, NOT ENOUGH INFO). Each editing item exposes a target claim, equivalent prompts for generalization, and an unrelated claim for specificity. Together with ZsRE, FEVER serves as one of the two main benchmarks behind Table 1.

A.3 Construction of Old Fact Equivalent Queries

The old-fact equivalent queries used by TSP are not assumed to be native fields of ZsRE, FEVER, or CounterFact. For each edit item, we instantiate the old fact from the source prompt, subject, and pre-edit target y_{old} (target_true), then ask

an instruction-tuned language model to produce semantically equivalent English statements. We use Qwen2.5-7B-Instruct with temperature 0.8, top-p 0.9, and 512 maximum tokens. For old-fact equivalents, the generator preserves the pre-edit subject and target and expresses only the same fact. It is explicitly instructed not to add citations, background, speculation, or additional facts.

Old-fact template.

Generate {num} English statements expressing the following old fact:

Old fact:

{knowledge_sentence}

Requirements:

1. Each statement must preserve exactly the same meaning and factual content.
2. Each statement must explicitly contain the subject "{subject}" and the target "{target_true}".
3. Do not add source attributions, background information, explanations, comments, speculation, modifiers, or additional facts.
4. Use concise paraphrases and express only this fact.
5. Output one complete English statement per line, ending with a period.
6. Do not use numbering, quotation marks, or any additional explanation.

Output {num} English statements, one per line.

For TSP, we generate 20 such statements per edit item, convert them into editing style queries, and stack their layer keys into K_{old} .

For the complex prompt robustness analysis in Section 5.5 and Appendix D, we use the same source fields but generate 100 harder equivalent queries per evaluated item. For complex prompts, the generator receives only source-side fields (source prompt, subject, and pre-edit target), never the post-edit target. It preserves the subject, uses diverse indirect or multi-hop wording, and does not reveal the pre-edit or post-edit target. We reject any complex query containing either target, remove empty lines, numbering, quotation marks, and duplicates, and regenerate until the required number of valid prompts is reached. We manually rated a random 10% sample for semantic equivalence, format validity, correct pre-edit-target retention in old-fact equivalents, and absence of both targets in complex prompts. The overall pass

rate was 95%, and failing prompts were regenerated. Old-fact equivalents construct K_{old} , whereas complex prompts are used only for the robustness analysis.

Complex-equivalent template.

Generate {num} diverse and complex English queries equivalent to the following source query:

Source query:

{original_prompt}

The query refers to the following pre-edit target. Do not include this target in any output:

{target_true}

Requirements:

1. Preserve exactly the same semantic intent as the source query.
2. Preserve the subject "{subject}" in every query.
3. Use diverse indirect or multi-hop wording while keeping the query semantically equivalent.
4. If the source is a question, output an equivalent complex question ending with a question mark.
5. If the source is an incomplete statement or prefix, output an equivalent complex prefix and stop where the target should naturally follow. Do not add a final period.
6. Do not reveal either the pre-edit target or the post-edit target.
7. Output exactly {num} independent English queries, one per line.
8. Do not use numbering, quotation marks, bullet points, or additional explanation.

Output {num} English queries, one per line.

A.4 Hyperparameter Settings

P-Space introduces a small set of manually specified hyperparameters. ϵ_{null} is the eigenvalue threshold that defines the null-space boundary when constructing $\mathcal{P}_{\text{curr}}$. β is the decay factor applied to the covariance accumulator in ACA. λ_* is the target protection level that determines how much covariance energy ACA injects for the new fact. ϵ is a small constant added to the denominator of Eq. (14) for numerical stability. The head energy ratio and band energy ratio control the bandpass selection in TSP: the head ratio sets the cumulative energy cutoff for discarding the top principal components, and the band ratio sets the cumulative energy captured by the selected middle band U_B . $|K_{\text{old}}|$ is the

number of synthesized old-fact equivalent queries used to construct K_{old} . All other quantities (α , λ_K , E_{raw} , $\mathcal{P}_{\text{curr}}$, etc.) are computed automatically at each edit step. The specific values are listed in Table 3.

Table 3: Hyperparameters of P-Space.

Symbol	Value
ϵ_{null}	2e-3
β	0.9
λ_*	1e-2
ϵ	1e-9
Head energy ratio	0.1
Band energy ratio	0.7
$ K_{\text{old}} $	20

A.5 Evaluation Metrics

ZsRE Metrics. For ZsRE, the now-standard top-1 accuracy formulation used in this benchmark is followed (Meng et al., 2022; Mitchell et al., 2022a; Meng et al., 2023). Let $f_{\mathcal{W}}$ be the LLM with parameters $\mathcal{W}$, (x, y) the edit query and its post-edit target, x_e a meaning-preserving rewrite of x , and $(x_{\text{loc}}, y_{\text{loc}})$ a locality query and the answer it should retain.

Efficacy. The fraction of edit prompts on which top-1 decoding from $f_{\mathcal{W}}$ matches the new target y :

$$\mathbb{E} \left\{ y = \arg \max_{y'} \mathbb{P}_{f_{\mathcal{W}}}(y' | x) \right\}. \quad (15)$$

Generalization. The same top-1 accuracy criterion, but evaluated on the equivalent prompt x_e , which checks whether the edit transfers to surface-form rewrites of the source query:

$$\mathbb{E} \left\{ y = \arg \max_{y'} \mathbb{P}_{f_{\mathcal{W}}}(y' | x_e) \right\}. \quad (16)$$

Specificity. The fraction of locality queries x_{loc} on which top-1 decoding still produces the original answer y_{loc} , which measures how well unrelated knowledge is preserved:

$$\mathbb{E} \left\{ y_{\text{loc}} = \arg \max_{y'} \mathbb{P}_{f_{\mathcal{W}}}(y' | x_{\text{loc}}) \right\}. \quad (17)$$

Throughout, $\mathbb{P}_{f_{\mathcal{W}}}(y' | x)$ is the probability that $f_{\mathcal{W}}$ assigns to a candidate output y' given input x .

FEVER Metrics. FEVER reuses the same three top-1 metrics, except that here the prediction space is a verification label rather than free-form text. Let (x, y) be the edit claim and its post-edit label, x_e a paraphrase of the claim, and (x_{loc}, y_{loc}) an unrelated claim with its label. The three metrics then read:

Efficacy.

$$\mathbb{E} \left\{ y = \arg \max_{y'} \mathbb{P}_{f_W}(y' | x) \right\}. \quad (18)$$

Generalization.

$$\mathbb{E} \left\{ y = \arg \max_{y'} \mathbb{P}_{f_W}(y' | x_e) \right\}. \quad (19)$$

Specificity.

$$\mathbb{E} \left\{ y_{loc} = \arg \max_{y'} \mathbb{P}_{f_W}(y' | x_{loc}) \right\}. \quad (20)$$

CounterFact Metrics. For CounterFact, following Meng et al. (2022, 2023), every metric reduces to a probability comparison between the post-edit target o_i and the pre-edit object o_i^c on a particular prompt set. Let $p_i = (s_i, r_i)$ be the edit prompt, $N_i = N((s_i, r_i))$ its rephrased prompt set, and $O_i = O((s_i, r_i))$ the neighborhood prompt set, whose prompts mention different but semantically related subjects.

Efficacy (efficacy success). The fraction of items on which o_i receives higher probability than o_i^c under the edit prompt p_i :

$$\mathbb{E}_i [\mathbb{P}_{f_W}(o_i | p_i) > \mathbb{P}_{f_W}(o_i^c | p_i)]. \quad (21)$$

Generalization (paraphrase success). The same comparison evaluated over the rephrased prompt set N_i :

$$\mathbb{E}_i \mathbb{E}_{p \in N_i} [\mathbb{P}_{f_W}(o_i | p) > \mathbb{P}_{f_W}(o_i^c | p)]. \quad (22)$$

Specificity (neighborhood success). The fraction of neighborhood prompts O_i for which the model still prefers the original object o_i^c over the edit target o_i :

$$\mathbb{E}_i \mathbb{E}_{p \in O_i} [\mathbb{P}_{f_W}(o_i^c | p) > \mathbb{P}_{f_W}(o_i | p)]. \quad (23)$$

These three quantities are reported as the Eff., Gen., and Spe. columns of the CounterFact ablation and editing-scale studies.

A.6 GLUE Benchmark

GLUE (General Language Understanding Evaluation) is a standard suite for training, evaluating, and analyzing natural language understanding systems. Six of its tasks are selected to track how well an edited model retains general language ability over a long edit stream.

- **SST** (Stanford Sentiment Treebank) is a single-sentence sentiment task built from movie-review fragments; the model assigns a binary sentiment label to each input sentence.
- **MMLU** (Massive Multi-task Language Understanding) is a broad multi-domain knowledge probe that scores zero-shot and few-shot performance on questions ranging from STEM subjects to the humanities.
- **MRPC** (Microsoft Research Paraphrase Corpus) targets semantic equivalence: given a sentence pair, the model decides whether the two sentences mean the same thing.
- **RTE** (Recognizing Textual Entailment) is a binary entailment task in which the model decides, given a premise and a hypothesis, whether the premise entails the hypothesis.
- **CoLA** (Corpus of Linguistic Acceptability) consists of grammaticality judgments collected from linguistics literature; the model classifies each sentence as acceptable or not.
- **NLI** (Natural Language Inference) is a sentence-pair task that asks the model to label the logical relation between two sentences.

B Dataset Visualization and Editing Examples

B.1 Dataset Item Examples

Figure 6 shows representative item formats from CounterFact, FEVER, and ZsRE, illustrating the edit targets and evaluation prompts used in our experiments.

C Supplementary Experimental Results

This section provides the detailed numbers behind the general capability and editing scale results discussed in the main text.

C.1 Results on General Capabilities

Table 6 reports the full general capability scores across continual editing steps. P-Space keeps these scores stable, while RECT and PRUNE drop sharply in later steps.

C.2 Results on Different Editing Scales

Table 7 reports detailed results under larger editing streams. P-Space keeps strong specificity and stable general capability as the number of edits increases.

D Detailed Results on Complex Prompts

To test how models behave on complex equivalent prompts after editing, AlphaEdit and P-Space are evaluated on ZsRE under two continual editing settings, 100×20 and 200×20 . Table 8 reports three output types: New Knowledge, Old Knowledge, and Other.

D.1 TSP Sensitivity

We evaluate the spectral choices in TSP with a one-factor-at-a-time analysis on CounterFact under the 20×100 setting. The remaining parameters use $\tau_h = 0.1$, $\tau_m = 0.7$, and $\epsilon_{\text{null}} = 2 \times 10^{-3}$. The default values provide the best overall balance, while nearby choices remain competitive.

Table 4: Sensitivity to TSP spectral thresholds on CounterFact.

Parameter	Value	Eff.	Gen.	Spe.
Head ratio τ_h	0.05	97.14	90.36	69.29
Head ratio τ_h	0.10	98.71	95.22	77.82
Head ratio τ_h	0.15	98.35	92.21	77.52
Band ratio τ_m	0.50	97.62	89.60	68.51
Band ratio τ_m	0.70	98.71	95.22	77.82
Band ratio τ_m	0.90	98.72	95.11	76.83

D.2 Computational Overhead

We profile AlphaEdit and P-Space on one NVIDIA A800 80GB GPU. P-Space requires additional time and memory for spectral decomposition and covariance updates.

Table 5: Computational overhead under identical hardware.

Method	Peak VRAM (GB)	Time/edit (s)
AlphaEdit	33.656	6.2307
P-Space	34.626	11.8221

P-Space adds 0.970 GB peak VRAM and 5.5914 seconds per edit relative to AlphaEdit.

E Proofs and Derivations

E.1 Proof of TSP Properties

We prove the two properties stated in Section 4.1.

Property 1: $\mathcal{P}_{\text{curr}} K_{\text{old}} \neq 0$. Let $U_B \in \mathbb{R}^{d \times b}$ be the selected band from the SVD of K_{old} , and define $\Pi_{\perp} = I - U_B U_B^T$.

Step 1. For any $u \in \text{span}(U_B)$, we have $U_B U_B^T u = u$, hence $\Pi_{\perp} u = 0$.

Step 2. It follows that

$$C_{\text{edit}} u = \Pi_{\perp} \bar{C}_l \Pi_{\perp} u = \Pi_{\perp} \bar{C}_l \cdot 0 = 0,$$

so every direction in $\text{span}(U_B)$ is an eigenvector of C_{edit} with eigenvalue zero.

Step 3. Since $\mathcal{P}_{\text{curr}}$ projects onto the eigenspace of C_{edit} with eigenvalues below ϵ_{null} , and $0 < \epsilon_{\text{null}}$, we have

$$\begin{aligned} \text{span}(U_B) &\subseteq \text{col}(\mathcal{P}_{\text{curr}}), \\ \text{i.e., } \mathcal{P}_{\text{curr}} u &= u, \forall u \in \text{span}(U_B). \end{aligned}$$

Step 4. By construction, U_B consists of columns of U from the thin SVD $K_{\text{old}} = U S V^T$. Since U_B captures a band of the spectrum of K_{old} with positive singular values, K_{old} has nonzero projection onto $\text{span}(U_B)$.

Conclusion. Combining Steps 3 and 4: $\text{span}(U_B) \subseteq \text{col}(\mathcal{P}_{\text{curr}})$ and K_{old} has nonzero energy in $\text{span}(U_B)$, so

$$\mathcal{P}_{\text{curr}} K_{\text{old}} \neq 0.$$

Property 2: $\mathcal{P}_{\text{curr}} K_p \approx 0$. Historical edit keys K_p correspond to facts different from the current old fact. Their activation directions are determined by different subjects and relations, and are thus approximately orthogonal to the fact-specific band U_B extracted from K_{old} .

Formally, let k_p be any column of K_p . Since TSP only zeroes the eigenvalues of C_{edit} along $\text{span}(U_B)$, the covariance C_{edit} approximately retains the eigenstructure of $\bar{C}_l$ in $\text{span}(U_B)^{\perp}$ (exactly so when $\text{span}(U_B)$ is an invariant subspace of $\bar{C}_l$). Because k_p comes from natural text inputs drawn from the same distribution used to estimate $\bar{C}_l$, its energy lies predominantly in the

non-null eigenspace of $\bar{C}_l$. This eigenspace is unchanged outside U_B , so k_p remains in the non-null eigenspace of C_{edit} , which is orthogonal to $\mathcal{P}_{\text{curr}}$. Therefore:

$$\mathcal{P}_{\text{curr}} k_p \approx 0.$$

The approximation becomes exact when $k_p \perp \text{span}(U_B)$. The bandpass design of TSP (dropping the top principal components that capture shared linguistic structure) reduces the overlap between U_B and unrelated keys, tightening this bound in practice.

E.2 Derivation of Adaptive Scaling α (Eq. 14)

We derive the closed-form expression for α in ACA.

Step 1. After ACA, the updated covariance is

$$\bar{C}'_l = \frac{\beta M_l + \alpha K_{\text{new}} K_{\text{new}}^T}{\beta N_l + \alpha}.$$

We require the average energy of $\bar{C}'_l$ along the new key directions to reach a target level $\lambda_\star \cdot \lambda_K$, where $\lambda_K = \|K_{\text{new}} K_{\text{new}}^T\|_2$.

Step 2. Let n be the number of columns in K_{new} . We require:

$$\frac{1}{n} \text{Tr}(K_{\text{new}}^T \bar{C}'_l K_{\text{new}}) = \lambda_\star \lambda_K. \quad (24)$$

Step 3. Substituting the updated covariance into the left-hand side:

$$\begin{aligned} & \text{Tr}(K_{\text{new}}^T (\beta M_l + \alpha K_{\text{new}} K_{\text{new}}^T) K_{\text{new}}) \\ &= \beta \text{Tr}(K_{\text{new}}^T M_l K_{\text{new}}) + \alpha \text{Tr}((K_{\text{new}}^T K_{\text{new}})^2) \\ &= \beta E_{\text{raw}} + \alpha \|K_{\text{new}}^T K_{\text{new}}\|_F^2, \end{aligned}$$

where $E_{\text{raw}} \triangleq \text{Tr}(M_l K_{\text{new}} K_{\text{new}}^T)$. For near-orthogonal keys, $\|K_{\text{new}}^T K_{\text{new}}\|_F^2 \approx n \lambda_K^2$.

Step 4. The target condition becomes:

$$\frac{\beta E_{\text{raw}} + \alpha n \lambda_K^2}{n (\beta N_l + \alpha)} = \lambda_\star \lambda_K.$$

Cross-multiplying and collecting α on the left:

$$\beta E_{\text{raw}} + \alpha n \lambda_K^2 = \lambda_\star \lambda_K n (\beta N_l + \alpha),$$

$$\alpha n (\lambda_K^2 - \lambda_\star \lambda_K) = \lambda_\star \beta N_l n \lambda_K - \beta E_{\text{raw}}.$$

Therefore:

$$\alpha = \frac{\beta (\lambda_\star N_l n \lambda_K - E_{\text{raw}})}{n (\lambda_K^2 - \lambda_\star \lambda_K + \epsilon)}, \quad (25)$$

where ϵ is added for numerical stability. Taking $\max(0, \cdot)$ ensures $\alpha \geq 0$: if the new directions already have sufficient energy ($E_{\text{raw}} \geq \lambda_\star N_l n \lambda_K$), no additional injection is needed.

CounterFact Dataset Example

```
{
  "case_id": 13032,
  "requested_rewrite": {
    "prompt": "{} was born in",
    "subject": "Kurupt",
    "target_new": {"str": "Tokyo"},
    "target_true": {"str": "Philadelphia"}
  },
  "paraphrase_prompts": [
    "Kurupt is native to",
    "Kurupt originated from"
  ],
  "neighborhood_prompts": [
    "Derek George was born in",
    "Dick Molpus originated from"
  ],
  "attribute_prompts": [
    "Shigeru Ban was native to",
    "Matsunaga Teitoku was born in"
  ]
}
```

FEVER Dataset Example

```
{
  "prompt": "Reign Over Me was released in 2017.",
  "equiv_prompt": [
    "Reign Over Me was published in 2017.",
    "Reign Over Me was launched in 2017."
  ],
  "ans": "REFUTES",
  "alt": "SUPPORTS",
  "unrel_prompt": "Right atrioventricular valve lies between the left atrium and the left ventricle.",
  "subject": "Reign Over Me"
}
```

ZsRE Dataset Example

```
{
  "subject": "Marlin Schneider",
  "src": "Which was the position that Marlin Schneider held?",
  "pred": "member of the Wisconsin State Assembly",
  "rephrase": "What's the position of Marlin Schneider?",
  "alt": "member of the Minnesota House of Representatives",
  "loc": "nq question: what's the dwarf's name in game of thrones",
  "loc_ans": "Tyrion Lannister"
}
```

Figure 6: Complete data structures of the three knowledge editing datasets used in our experiments. The figure displays the exact JSON format for each dataset: **CounterFact** (top) contains editing requests specific to each case with comprehensive evaluation prompts; **FEVER** (middle) provides fact verification instances with three class classification labels (SUPPORTS, REFUTES, or NOT ENOUGH INFO); **ZsRE** (bottom) presents question answering format edits with rephrasing and specificity components.

Table 6: Detailed performance of general capabilities (F1 Score) under a continual editing regime. [†]Results of baseline methods are taken from RLEdit (Li et al., 2025).

Method	SST	MRPC	CoLA	RTE	MMLU	NLI
<i>RLEdit</i> [†]						
Step 0	0.83	0.68	0.77	0.28	0.57	0.66
Step 1	0.80	0.68	0.85	0.27	0.56	0.65
Step 2	0.81	0.67	0.75	0.27	0.54	0.66
Step 3	0.80	0.67	0.76	0.27	0.54	0.66
Step 4	0.80	0.67	0.74	0.27	0.55	0.64
Step 5	0.86	0.68	0.75	0.28	0.54	0.66
Step 6	0.86	0.70	0.77	0.28	0.55	0.67
<i>AlphaEdit</i> [†]						
Step 0	0.83	0.68	0.77	0.28	0.57	0.66
Step 1	0.82	0.69	0.78	0.26	0.52	0.69
Step 2	0.81	0.69	0.76	0.32	0.58	0.67
Step 3	0.87	0.73	0.74	0.30	0.59	0.62
Step 4	0.86	0.71	0.76	0.29	0.58	0.63
Step 5	0.85	0.69	0.74	0.27	0.57	0.63
Step 6	0.82	0.69	0.72	0.26	0.49	0.62
<i>RECT</i> [†]						
Step 0	0.83	0.68	0.77	0.28	0.57	0.66
Step 1	0.80	0.68	0.78	0.38	0.56	0.69
Step 2	0.81	0.64	0.80	0.30	0.56	0.67
Step 3	0.87	0.39	0.75	0.27	0.53	0.61
Step 4	0.24	0.15	0.18	0.28	0.03	0.30
Step 5	0.00	0.00	0.00	0.04	0.00	0.00
Step 6	0.00	0.00	0.00	0.00	0.00	0.00
<i>PRUNE</i> [†]						
Step 0	0.83	0.68	0.77	0.28	0.57	0.66
Step 1	0.79	0.67	0.74	0.28	0.50	0.69
Step 2	0.81	0.66	0.77	0.31	0.54	0.63
Step 3	0.70	0.52	0.66	0.30	0.34	0.68
Step 4	0.00	0.00	0.00	0.00	0.00	0.00
Step 5	0.00	0.00	0.00	0.00	0.00	0.00
Step 6	0.00	0.00	0.00	0.00	0.00	0.00
<i>MEND</i> ^{*†}						
Step 0	0.83	0.68	0.77	0.28	0.57	0.66
Step 1	0.00	0.00	0.00	0.00	0.00	0.00
Step 2	0.00	0.00	0.00	0.00	0.00	0.00
Step 3	0.00	0.00	0.00	0.00	0.00	0.00
Step 4	0.00	0.00	0.00	0.00	0.00	0.00
Step 5	0.00	0.00	0.00	0.00	0.00	0.00
Step 6	0.00	0.00	0.00	0.00	0.00	0.00
<i>P-Space</i>						
Step 0	0.83	0.68	0.77	0.28	0.57	0.66
Step 1	0.83	0.66	0.76	0.29	0.56	0.65
Step 2	0.83	0.65	0.75	0.27	0.58	0.65
Step 3	0.81	0.65	0.76	0.27	0.56	0.64
Step 4	0.82	0.65	0.77	0.28	0.57	0.65
Step 5	0.81	0.66	0.75	0.29	0.59	0.63
Step 6	0.81	0.64	0.76	0.28	0.57	0.65

Table 7: Detailed performance comparison across different editing scales (20×100 to 150×100). [†]Results of baseline methods are taken from RLEdit (Li et al., 2025).

Scale	Method	Eff.	Gen.	Spe.	GC
20×100 (2k)	MEND [†]	0.00	0.00	0.00	3.0
	AlphaEdit [†]	94.47	91.13	32.55	61.0
	RLEdit [†]	88.65	83.91	47.61	61.0
	RECT [†]	86.02	81.81	32.04	20.0
	P-Space	87.25	83.83	49.01	61.0
50×100 (5k)	MEND [†]	0.00	0.00	0.00	1.0
	AlphaEdit [†]	93.76	88.65	31.71	60.5
	RLEdit [†]	85.46	80.68	43.35	61.0
	RECT [†]	0.07	0.07	1.34	0.0
	P-Space	84.36	82.14	46.41	61.0
100×100 (10k)	MEND [†]	0.00	0.00	0.00	0.0
	AlphaEdit [†]	89.60	86.75	30.96	50.0
	RLEdit [†]	86.45	83.07	41.96	61.0
	RECT [†]	0.08	0.08	1.36	0.0
	P-Space	89.82	85.50	44.10	60.0
150×100 (15k)	MEND [†]	0.00	0.00	0.00	0.0
	AlphaEdit [†]	84.43	78.28	28.48	47.0
	RLEdit [†]	87.81	85.13	40.98	60.0
	RECT [†]	0.02	0.03	1.02	0.0
	P-Space	86.37	85.29	43.93	60.0

Table 8: Evaluation of cognitive dissonance on ZsRE under complex equivalent prompts. The frequency (%) of generated targets matching New Knowledge, Old Knowledge, or Other is reported.

Setting	Method	New Knowledge ($\uparrow$)	Old Knowledge ($\downarrow$)	Other ($\downarrow$)
100×20	AlphaEdit	65.21	21.36	13.43
	P-Space (Ours)	78.27	13.15	8.58
200×20	AlphaEdit	49.37	20.13	30.50
	P-Space (Ours)	63.92	12.15	23.93