CFPP: Conflict-Free Preservation Projection for Locate-then-Edit Model Editing

Anonymous ACL submission

Abstract

Locate-then-edit methods update factual knowledge in large language models by first localizing a set of editable neurons or layers and then computing the necessary weight updates. However, these methods face a fundamental tension between enforcing target edits and preserving unrelated knowledge, where the very regularization designed for preservation can suppress the desired edit. This conflict is most acute when the preservation subspace exhibits significant overlap with the edit subspace. Through a geometric and optimization analysis, we demonstrate that such subspace overlap inevitably pits editing efficacy against knowledge preservation. To resolve this intrinsic contradiction, we introduce Conflict-Free Preservation Projection (CFPP), a plug-and-play framework that projects the preservation subspace away from the edit subspace. By incorporating this projected term into the edit objective, CFPP enables a conflict-free integration of the preservation constraint into the editing process. We instantiate CFPP on several representative locate-then-edit methods and evaluate it on CounterFact, ZsRE, and Wiki-recent with GPT2-XL, GPT-J, and Llama-3 models. Results show that CFPP consistently and significantly enhances their editing performance. Code is available at https://anonymous.4open.science/r/CFPP-5206.

1 Introduction

Model editing aims to update a deployed language model (Cao et al., 2021; Yao et al., 2023; Zhang et al., 2024) so that a small set of facts changes while the rest of the model behavior remains stable. This problem is especially important in post-deployment scenarios such as factual correction, knowledge refresh, and safety intervention, where retraining the full model incurs prohibitive computational and time costs. In recent years, a strong family of *locate-then-edit* methods has emerged, including ROME (Meng et al., 2022), MEMIT (Meng et al., 2023), PMET (Li et al., 2024), AlphaEdit (Fang et al., 2025), and related variants. These methods follow a common paradigm: they first localize an internal representation associated with a target fact, and then compute a structured local update, often low-rank or closed-form, to a small subset of model parameters.

Despite their differences in parameterization and update strategy, these methods share a common structural objective: they must enforce the target factual rewrite while preserving unrelated knowledge. In practice, this preservation requirement is typically enforced by a second-moment regularizer computed from background activations, which penalizes updates that interfere with preserved knowledge. While such a design is intuitively reasonable, it also introduces an inherent tension between preservation and editing. The regularization that protects preserved knowledge may suppress the update directions needed for the target edit. When their underlying subspaces overlap, the local optimization is forced into a trade-off, where stronger preservation may lead to weaker editing.

This issue becomes particularly severe in batched or large-scale editing settings (Meng et al., 2023; Gupta et al., 2024). As the number of simultaneous edits increases, the edit keys span a larger and more heterogeneous subspace, making overlap with the dominant preservation subspace increasingly likely. In this regime, the preservation term no longer acts merely as a stabilizer; it can become an active source of interference by penalizing the update components required for successful editing. Consequently, existing locate-then-edit methods may underperform even when the target representations are correctly localized, because the resulting local optimization objective contains an inherent preservation–edit conflict.

This paper introduces **Conflict-Free Preservation Projection** (CFPP), a plug-and-play model editing framework that resolves this conflict at its

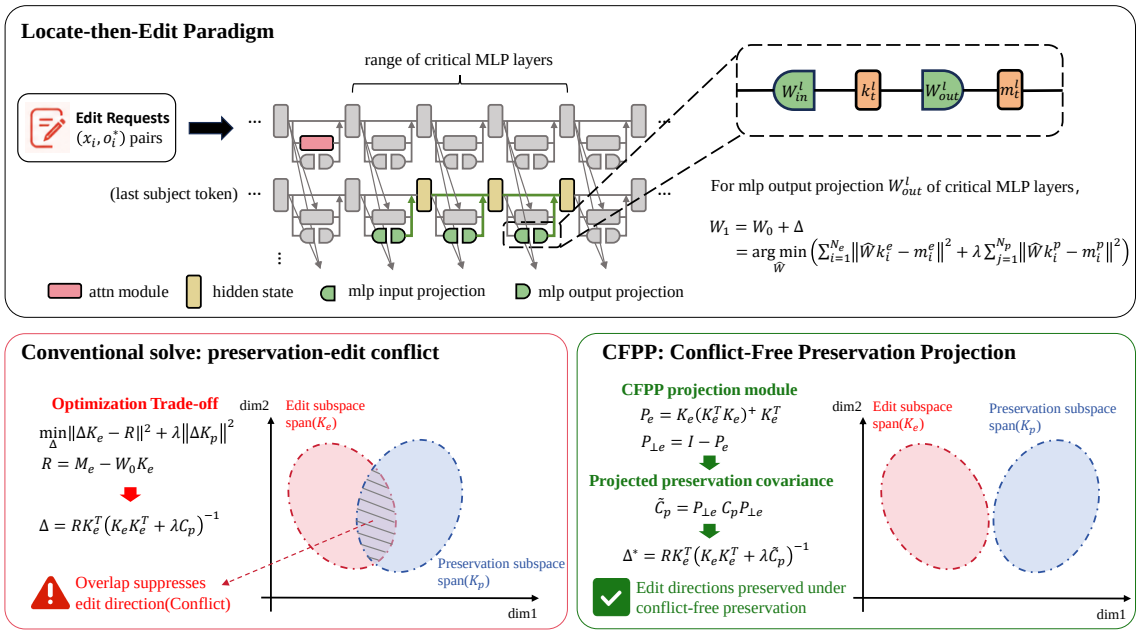

Figure 1: Overview of the locate-then-edit paradigm and the preservation–edit conflict. Top: edit requests are routed to critical MLP layers, where a local closed-form update is computed. Bottom left: in the conventional solve, overlap between the edit subspace and the preservation subspace causes the preservation regularizer to penalize edit-relevant directions, leading to an optimization trade-off. Bottom right: CFPP projects the preservation subspace away from the edit subspace, enabling conflict-free preservation while keeping edit-relevant directions unconstrained.

source. The key idea is simple yet principled: before solving the local update, CFPP projects the preservation subspace away from the subspace spanned by the active edit keys. This ensures that the preservation subspace constrains only directions orthogonal to the edit subspace, while leaving edit-relevant directions free to realize the target factual update. In this way, CFPP does not weaken preservation indiscriminately; instead, it eliminates only the conflicting part of the preservation constraint that interferes with the current edit objective.

From an optimization perspective, CFPP can be viewed as a minimal intervention in the local closed-form solution underlying many locate-then-edit methods. It does not alter the target-vector optimization stage, introduce additional trainable parameters, or require retraining or auxiliary supervision. Rather, CFPP simply replaces the raw preservation covariance with a conflict-free projected counterpart before solving for the weight update. This makes CFPP especially practical: it preserves the overall workflow of existing methods while improving editing efficacy through a conflict-free local solve.

Our main contributions are summarized as follows:

- **Theoretical Insight:** We identify preservation–edit subspace overlap as a critical failure mode in *locate-then-edit* methods and formalize it as a geometric conflict in the local optimization objective.
- **Method:** We propose CFPP, a projection mechanism that resolves this conflict by removing edit-subspace components from the preservation subspace. CFPP requires only a single matrix modification and seamlessly integrates into existing locate-then-edit methods.
- **Evaluation:** We conduct comprehensive experiments across multiple models and datasets, demonstrating that CFPP improves editing efficacy, generalization, and locality in large-scale editing scenarios through rigorous benchmarks and geometric diagnostics.

2 Related Work

Knowledge Editing. Existing knowledge editing methods fall into two main categories: auxiliary-based methods (Mitchell et al., 2022a,b; Hartvigsen et al., 2023; Qi et al., 2025) that preserve original parameters and introduce additional information or components, and location-based (parameter-modifying) methods that directly adjust weights for

efficient factual updates without costly retraining. Building on the insight that factual associations reside in LLM feed-forward networks (Geva et al., 2021, 2022; Dai et al., 2022), ROME uses causal tracing to locate and modify critical MLP layers, while MEMIT extends this to enable massive concurrent editing. Subsequently, many other location-based methods have emerged: PMET employs attention layers to assist optimization, AnyEdit (Jiang et al., 2025a) uses autoregressive chunked editing to handle knowledge of arbitrary length and diverse formats, while EMMET (Dai et al., 2026) introduces a residual embedding alignment mechanism, and so on (Li and Chu, 2025; Zhang et al., 2025; Jiang et al., 2025b; Qiao et al., 2025).

Interference Elimination. Theoretical analyses of knowledge superposition (Hu et al., 2025) suggest that interference can arise when factual representations occupy non-orthogonal directions in the representation space. To mitigate such interference, recent works have explored projection-based or subspace-based editing constraints. AlphaEdit (Fang et al., 2025) projects parameter perturbations onto the null space of preserved knowledge to reduce disruption to unrelated facts, while SUIT (Park et al., 2026) confines edits to edit-relevant subspaces to avoid unnecessary perturbations. Related studies on sequential or multilingual editing further constrain new updates to be orthogonal to previous update subspaces, thereby reducing cross-edit interference (Sun et al., 2025; Lyu et al., 2025; Cai and Cao, 2024; Xu et al., 2025).

Despite these advances, existing approaches mainly constrain the update itself, rather than the preservation term. As a result, the geometric conflict caused by overlap between the edit subspace and the dominant preservation subspace remains insufficiently addressed. In contrast, CFPP resolves this conflict at its source by projecting the preservation subspace away from the active edit subspace before solving the local update. This enables conflict-free preservation without penalizing the directions required for the current edit.

3 Preliminaries: Editing Memory in LLMs

Prior work has shown that pretrained language models store substantial factual knowledge in their parameters, which can be probed through cloze-style queries, closed-book question answering, and automatically optimized prompts (Petroni et al., 2019; Roberts et al., 2020; Shin et al., 2020; Jiang et al., 2021). A prominent line of research further suggests that such knowledge is primarily encoded in the feed-forward network (FFN) sublayers, where factual associations can be interpreted as linear key-value memories (Geva et al., 2021, 2022; Dai et al., 2022). Building on this view, locate-then-edit methods regard the MLP down-projection matrix W^l_{out} as an associative memory store and update specific facts by directly modifying this matrix (Meng et al., 2022, 2023).

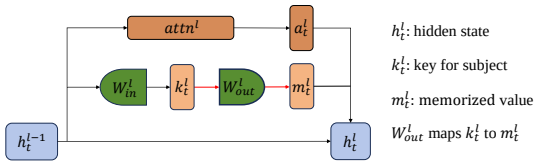

Figure 2: Illustration of the model editing problem.

As illustrated in Figure 2, each FFN layer maps subject representations $k^l_t(x) = \sigma(W^l_{\text{in}}\gamma(h^{l-1}_t(x)))$ to their corresponding memory outputs $m^l_t(x) = W^l_{\text{out}}\sigma(W^l_{\text{in}}\gamma(h^{l-1}_t(x)))$. Following causal tracing analyses (Meng et al., 2022), modifying W^l_{out} alone is often sufficient to alter these stored associations. The edit objective is therefore to find an updated weight matrix $\hat{W}$ that maps edit keys to their target values while preserving the mappings of unrelated background keys. Formally, this objective can be written as:

$$W_1 \triangleq \arg\min_{\hat{W}} \Bigg(\sum_{i=1}^{N_e} \|\hat{W}k^e_i - m^e_i\|^2 + \lambda \sum_{j=1}^{N_p} \|\hat{W}k^p_j - m^p_j\|^2 \Bigg), \tag{1}$$

where $\{k^e_i, m^e_i\}_{i=1}^{N_e}$ denote the key-value pairs associated with the target facts to be edited, and $\{k^p_j, m^p_j\}_{j=1}^{N_p}$ denote background key-value pairs whose mappings should be preserved. The hyperparameter λ controls the trade-off between edit fidelity and knowledge preservation.

By collecting keys and values into matrices $K_e, M_e \in \mathbb{R}^{d\times N_e}$ and $K_p, M_p \in \mathbb{R}^{d\times N_p}$, and decomposing the edited weights as $W_1 = W_0 + \Delta$, where W_0 denotes the original parameters satisfying $W_0K_p = M_p$, the objective admits a closed-form solution through the normal equations (Meng

et al., 2023):

$$(W_0 + \Delta)[K_p \; K_e] = [M_p \; M_e], \tag{2}$$

$$W_0 K_p = M_p. \tag{3}$$

Combining Equation (2) with $[K_p \; K_e]^\top$ and subtracting Equation (3) yields the following update for W^l_{out} (Meng et al., 2023):

$$\Delta = R K_e^\top (\lambda C_p + K_e K_e^\top)^{-1}, \tag{4}$$

where $R = M_e - W_0 K_e$ is the residual that steers edit keys toward their target values, and $C_p = K_p K_p^\top$ is a second-moment matrix constructed from background activations. Since the original pretraining corpus is typically inaccessible, C_p is commonly estimated from random samples drawn from public text corpora (Meng et al., 2023). This closed-form formulation underlies many modern locate-then-edit methods, offering an efficient mechanism for targeted knowledge injection while also exposing a geometric tension between the edit subspace induced by K_e and the preservation structure encoded by C_p.

4 Method

4.1 The Preservation–Edit Subspace Conflict

Locate-then-edit methods such as MEMIT formulate knowledge editing as a constrained least-squares problem that balances target fact insertion with preservation of unrelated knowledge. For a batch of edits, the local objective can be written as:

$$\min_{\Delta} \; \|\Delta K_e - R\|^2 + \lambda \|\Delta K_p\|^2, \tag{5}$$

where $K_e \in \mathbb{R}^{d \times N_e}$ contains the key representations of the target facts, K_p aggregates background keys whose mappings should be preserved.

Although this formulation is mathematically elegant, it entails an intrinsic preservation–edit conflict. When the dominant eigenspace of the preservation covariance C_p overlaps with the edit subspace $\text{span}(K_e)$, the preservation regularizer $\lambda \|\Delta K_p\|^2$ penalizes directions that are also needed to realize the target edits. This conflict becomes more pronounced in massive editing settings, where the edit-key matrix spans a larger and more heterogeneous subspace, thereby increasing its overlap with dominant preservation directions. As a result, the regularizer can suppress edit-relevant update components, leading to systematic under-editing.

Model	C_p	CounterFact				
		Overlap↓	θ_{mean} ↑	Eff.↑	Gen.↑	Spe.↑
Llama3-8B	Raw	0.0661	78.79	95.40	54.20	24.46
	C_{h_1}	0.0957	75.11	95.60	52.50	23.65
	C_{h_2}	0.0718	78.23	95.60	53.10	23.81
	C_{l_1}	0.0556	79.42	96.10	54.65	24.57
	C_{l_2}	0.0488	80.51	96.70	55.95	24.78

Table 1: Impact of preservation–edit subspace overlap on editing performance. Overlap and principal-angle measurements are computed on the final selected editing layer using rank-32 subspaces.

We quantify this conflict through subspace-overlap diagnostics. Let $U_p \in \mathbb{R}^{d \times r}$ denote the top-r eigenvectors of C_p, and let $U_e \in \mathbb{R}^{d \times r}$ denote the top-r left singular vectors of K_e. We measure their geometric alignment in two complementary ways. First, we compute the principal angles:

$$\theta_i = \arccos \sigma_i(U_p^\top U_e), \qquad i = 1, \ldots, r, \tag{6}$$

where $\sigma_i(\cdot)$ denotes the i-th singular value. Smaller principal angles indicate stronger alignment between preservation and editing directions, and therefore greater potential interference. Second, we summarize the overall overlap using the normalized Frobenius score:

$$\text{Overlap}(U_p, U_e) = \frac{\|U_p^\top U_e\|_F^2}{r}, \tag{7}$$

which lies in $[0, 1]$ when the two subspaces have the same rank. A larger value indicates that a greater fraction of the edit subspace is aligned with dominant preservation directions and is therefore more likely to be suppressed by the regularizer. Further justification of these diagnostics is provided in Appendix A.

Table 1 illustrates this effect on Llama3-8B under the MEMIT editing method with 1,000 edits. We compare the raw preservation covariance with four alternative covariances sampled from the same number of Wikipedia samples: two higher-overlap variants, C_{h1} and C_{h2}, and two lower-overlap variants, C_{l1} and C_{l2}, all measured with respect to K_e. The results show a clear trend: higher overlap leads to smaller principal angles and weaker editing performance, especially in generalization and specificity, whereas lower overlap yields larger principal angles and consistently improves all three editing metrics. These observations support our claim that preservation–edit subspace overlap directly degrades edit performance.

4.2 Conflict-Free Preservation Projection

To resolve this conflict, we propose Conflict-Free Preservation Projection (CFPP), which removes the edit-aligned effect of the preservation covariance while retaining its components orthogonal to the edit subspace. The key idea is to eliminate only the overlap between the preservation directions and the active edit subspace, rather than weakening preservation uniformly, as illustrated in Figure 1.

The orthogonal projector onto the edit subspace is defined as

$$P_e = K_e(K_e^\top K_e)^+ K_e^\top, \tag{8}$$

where $(\cdot)^+$ is the Moore-Penrose pseudoinverse. The projector onto the orthogonal complement of the edit subspace is then given by:

$$P_{\perp e} = I - P_e. \tag{9}$$

Instead of using the raw preservation covariance C_p, CFPP replaces it with the projected covariance:

$$\widetilde{C}_p = P_{\perp e} C_p P_{\perp e}. \tag{10}$$

As shown in Appendix A, this projection preserves the directions of C_p that are orthogonal to $\mathrm{span}(K_e)$ while removing its edit-aligned effect. Intuitively, preservation constraints are enforced only in directions that do not interfere with the current edit.

Substituting $\widetilde{C}_p$ into the local solve gives the CFPP update rule:

$$\Delta^* = RK_e^\top \left(K_e K_e^\top + \lambda \widetilde{C}_p\right)^{-1}. \tag{11}$$

Compared with the baseline solver, the only change is the replacement $C_p \mapsto \widetilde{C}_p$. This operation introduces no additional trainable parameters and requires neither retraining nor architectural modification. Conceptually, CFPP does not weaken preservation globally; instead, it prevents the preservation subspace from penalizing directions required for the current edit. The update is therefore encouraged to preserve background behavior only along directions outside the edit subspace, yielding a conflict-free edit objective.

4.3 Soft Projection for Flexible Trade-offs

Hard projection, corresponding to $\beta = 1$, may be overly aggressive when some directions shared by the preservation and edit subspaces still encode useful background information. To provide finer control over the efficacy-locality trade-off, we introduce a soft projection variant:

$$P_\beta = I - \beta P_e, \qquad \beta \in [0, 1], \tag{12}$$

which produces the softened covariance:

$$\widetilde{C}_\beta = P_\beta C_p P_\beta. \tag{13}$$

The hyperparameter β interpolates between the baseline ($\beta = 0$) and full conflict removal ($\beta = 1$), with $\beta = 0.5$ generally providing a favorable balance. Implementation details are given in Appendix B.

5 Experiments

In this section, we empirically evaluate CFPP from three complementary perspectives. We first examine whether CFPP mitigates the geometric conflict that motivates our method. We then evaluate its effectiveness in massive editing scenarios across different LLMs. Finally, we investigate whether CFPP can be effectively integrated into other locate-then-edit methods. Specifically, we address the following research questions (RQs):

- **RQ1.** Can CFPP effectively alleviate the conflict between preserved knowledge and edited knowledge?
- **RQ2.** How does CFPP perform on massive editing tasks compared with strong baselines across different LLMs?
- **RQ3.** Can CFPP be effectively integrated into other locate-then-edit methods?

5.1 Experimental Setup

Models, datasets, and baselines. We conduct experiments on three representative LLMs: GPT2-XL, GPT-J-6B, and Llama3-8B. We evaluate editing performance on three standard knowledge editing benchmarks: CounterFact (Meng et al., 2022), ZsRE (Levy et al., 2017), and Wiki-recent (Zhang et al., 2024). CounterFact measures factual rewriting quality under templated prompts and paraphrases, ZsRE evaluates question-answer-style generalization, and Wiki-recent assesses the ability to update recently changed facts while preserving unrelated behavior. As baselines, we compare against representative model editing methods, including FT, ROME, MEMIT, PMET, and AlphaEdit.

Model	CounterFact				ZsRE			
	Overlap↓		θ_{mean} ↑		Overlap↓		θ_{mean} ↑	
	Before CFPP	After CFPP	Before CFPP	After CFPP	Before CFPP	After CFPP	Before CFPP	After CFPP
GPT2-XL	0.1269	0.0406	74.41	82.62	0.0886	0.0315	77.75	83.95
GPT-J-6B	0.1410	0.0477	72.70	81.65	0.1078	0.0419	75.72	82.59
Llama3-8B	0.0661	0.0176	78.79	84.75	0.0685	0.0194	78.57	84.48

Table 2: Conflict diagnostics on the final selected editing layer of each model, computed from rank-32 subspaces using 1,000 edit requests per dataset. "Overlap" denotes the normalized Frobenius overlap score between the dominant preservation subspace and the edit subspace, and θ_{mean} denotes the mean principal angle in degrees.

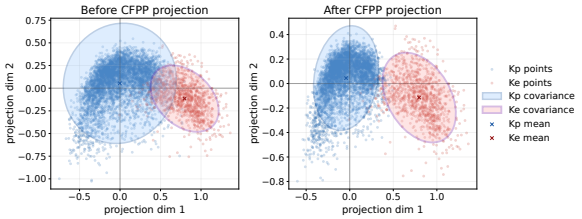

Figure 3: Visualization of the preservation–edit subspace conflict before and after CFPP projection on the final selected editing layer. Blue points and ellipses denote the preservation distribution induced by K_p, while red points and ellipses denote the edit-key distribution induced by K_e.

Metrics. Formally, let o_i denote the model output for the i-th edited query and $o_i^\star$ denote the target entity. Efficacy is defined as

$$\text{Eff.} = \mathbb{E}_i \left[o_i^\star = \arg\max_o \ \mathbb{P}_{f_\theta}(o \mid (s_i, rel_i)) \right], \tag{14}$$

where (s_i, rel_i) denotes the subject-relation query and f_θ is the edited model. For CounterFact and Wiki-recent, efficacy is directly computed on factual completion prompts by checking whether the target entity is assigned the highest probability. For ZsRE, however, efficacy is measured by the average token-level hit rate on the target answer rather than strict whole-answer retrieval. Detailed metric definitions and dataset-specific evaluation protocols are provided in Appendix C. Together, these metrics provide a balanced view of editing quality by measuring whether the target fact is inserted, whether it generalizes across prompt variants, and whether unrelated knowledge remains stable.

5.2 Effectiveness of CFPP in Alleviating Preservation–Edit Conflict (RQ1)

To determine whether CFPP reduces the geometric conflict that motivates our method, we analyze the overlap between the dominant preservation subspace induced by C_p and the active edit subspace induced by K_e. As discussed in Section 4.1, we quantify this conflict using principal angles and overlap scores computed from the top eigenspace of C_p and the top left singular subspace of K_e.

Table 2 provides direct evidence that CFPP reduces geometric interference. Across all three model backbones, the overlap score decreases substantially after projection, while the mean principal angle increases. This indicates that the dominant preservation subspace becomes more separated from the active edit subspace, exactly as intended by CFPP. Figure 3 provides a complementary visualization. After CFPP, the preservation distribution induced by K_p becomes more separated from the edit distribution induced by K_e, resulting in visibly reduced overlap. This is consistent with the quantitative results in Table 2, further confirming that CFPP alleviates the preservation–edit conflict.

5.3 Main Results on Massive Editing (RQ2)

We evaluate the effectiveness of MEMIT-CFPP in massive editing scenarios by comparing it with strong baselines on GPT2-XL, GPT-J-6B, and Llama3-8B. Specifically, we sample 10,000 edit requests from CounterFact and ZsRE, and use all 1,266 edit requests from Wiki-recent to evaluate the model's ability to update recently changed facts.

As shown in Table 3, MEMIT-CFPP consistently improves over MEMIT across most models and benchmarks, with the most notable gains appearing in efficacy and generalization. These improvements suggest that removing the conflicting components of the preservation covariance helps the local update better realize the target edits. The gains are most evident on GPT2-XL and Llama3-8B, where CFPP substantially boosts performance on CounterFact, Wiki-recent, and ZsRE while maintaining comparable specificity or locality. On GPT-J-6B, MEMIT-CFPP also improves generalization while preserving comparable specificity and locality, with

Model	Method	CounterFact			Wiki-recent			ZsRE		
		Eff.↑	Gen.↑	Spe.↑	Eff.↑	Por.↑	Loc.↑	Eff.↑	Gen.↑	Spe.↑
GPT2-XL	Pre-edit	0.28	0.40	10.76	2.61	3.80	3.58	22.19	21.25	24.18
	FT	**99.99**	15.91	9.85	30.73	3.80	3.58	42.45	40.36	2.50
	ROME	0.62	0.46	1.05	24.80	3.79	3.47	21.20	18.70	1.65
	PMET	19.19	9.14	**10.70**	27.80	9.95	13.17	39.42	36.58	24.90
	AlphaEdit	65.34	**32.05**	4.85	**82.86**	**18.82**	**21.97**	64.01	56.91	20.96
	MEMIT	46.32	25.64	9.94	47.31	13.93	18.23	62.58	58.02	**25.90**
	MEMIT$^+$	57.78$_{+11.46}$	31.97$_{+6.33}$	8.65	56.71$_{+9.40}$	14.28$_{+0.35}$	18.79$_{+0.56}$	**68.66**$_{+6.08}$	**62.84**$_{+4.82}$	24.90
GPT-J-6B	Pre-edit	0.36	0.45	14.62	2.45	3.09	6.13	26.46	25.74	27.02
	FT	**99.98**	45.46	6.05	38.31	17.82	18.68	76.33	73.08	24.67
	ROME	0.00	0.00	0.00	27.88	6.35	2.51	22.82	20.69	1.08
	PMET	96.84	**65.50**	10.85	87.91	22.47	22.99	88.62	82.49	26.25
	AlphaEdit	93.10	45.06	6.47	**99.13**	23.65	25.34	83.57	72.33	22.48
	MEMIT	95.60	54.63	11.38	98.10	23.97	27.14	95.85	88.39	26.30
	MEMIT$^+$	95.42	55.31$_{+0.68}$	**11.60**$_{+0.22}$	98.63$_{+0.53}$	**24.01**$_{+0.04}$	**27.22**$_{+0.08}$	**95.94**$_{+0.09}$	**88.87**$_{+0.48}$	**26.43**$_{+0.13}$
Llama3-8B	Pre-edit	0.32	0.38	22.08	6.16	5.76	15.69	36.38	35.70	31.84
	FT	21.68	3.61	0.01	0.00	0.00	0.00	39.12	38.36	20.54
	ROME	4.82	1.97	0.24	9.08	2.07	0.82	2.41	1.83	0.17
	PMET	90.78	**50.60**	15.87	85.86	26.89	30.00	88.66	84.31	31.20
	AlphaEdit	76.26	36.88	**22.12**	79.46	26.22	30.30	85.70	81.77	32.15
	MEMIT	85.61	42.18	21.47	86.41	25.46	30.56	89.19	85.42	**32.43**
	MEMIT$^+$	**92.70**$_{+7.09}$	46.42$_{+4.24}$	20.68	**93.29**$_{+6.88}$	**27.28**$_{+1.82}$	**33.09**$_{+2.53}$	**91.11**$_{+1.92}$	**86.30**$_{+0.88}$	31.80

Table 3: Performance comparison of knowledge editing methods on GPT2-XL, GPT-J-6B, and Llama3-8B across CounterFact, Wiki-recent, and ZsRE. Reported values are averaged over three independent runs. ROME is evaluated under sequential editing due to its methodological constraints. Best results are highlighted in **bold**, and cases where MEMIT$^+$ (MEMIT-CFPP) outperforms MEMIT are underlined.

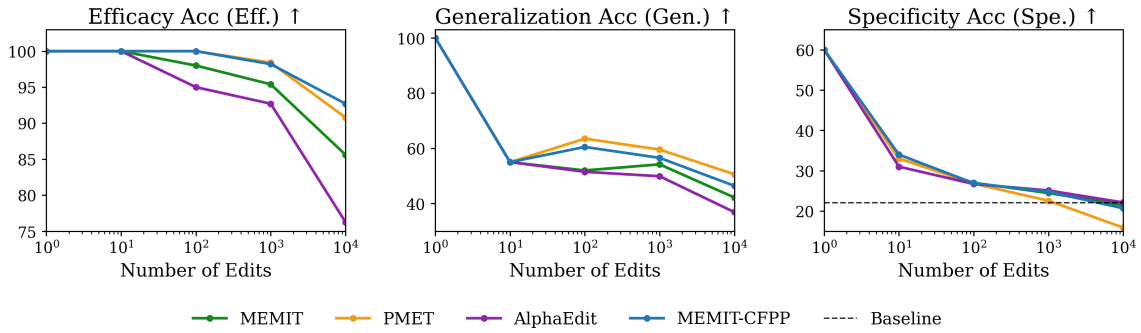

Figure 4: Scaling behavior on Llama3-8B for CounterFact as the number of edits increases from 1 to 10^4. We compare MEMIT, PMET, AlphaEdit, and MEMIT-CFPP in terms of efficacy, generalization, and specificity.

only a slight change in efficacy. Overall, these results show that CFPP strengthens edit insertion and transfer without simply sacrificing preservation.

Figure 4 analyzes the scaling behavior as the number of edits increases from 1 to 10^4. MEMIT-CFPP maintains high efficacy across all edit scales and shows stronger generalization than MEMIT and AlphaEdit at large scales. Although specificity decreases for all methods as the edit scale grows, MEMIT-CFPP remains comparable to existing baselines, indicating that CFPP improves large-scale editing without causing additional preservation degradation. Additional results on more models and datasets are provided in Appendix D.

5.4 Generalization to Other Methods (RQ3)

CFPP can be applied to locate-then-edit methods that compute local closed-form updates regularized by a preservation covariance. We integrate it into PMET and AlphaEdit to evaluate its generality beyond MEMIT.

Table 4 evaluates CFPP when integrated into PMET and AlphaEdit across three models on CounterFact and ZsRE, where 10,000 knowledge facts are edited. CFPP generally improves efficacy and generalization for both methods, with especially

Model	Method	CounterFact				ZsRE			
		Eff.↑	Gen.↑	Spe.↑	Avg.↑	Eff.↑	Gen.↑	Spe.↑	Avg.↑
GPT2-XL	PMET	19.19	9.14	10.70	13.01	39.42	36.58	24.90	33.63
	PMET+CFPP	**25.15**	**12.38**	10.11	**15.88**	**42.20**	**38.53**	24.63	**35.12**
	AlphaEdit	65.34	32.05	4.85	34.08	64.01	56.91	20.96	47.29
	AlphaEdit+CFPP	**66.92**	**32.14**	**5.05**	**34.70**	**64.10**	56.85	**21.00**	**47.32**
GPT-J-6B	PMET	96.84	65.50	10.85	57.73	88.62	82.49	26.25	65.79
	PMET+CFPP	**96.92**	**67.30**	**10.85**	**58.36**	**89.69**	**83.64**	26.13	**66.49**
	AlphaEdit	93.10	45.06	6.47	48.21	83.57	72.33	22.48	59.46
	AlphaEdit+CFPP	**93.46**	**45.28**	**6.54**	**48.43**	**83.63**	**73.14**	22.30	**59.69**
Llama3-8B	PMET	90.78	50.60	15.87	52.42	88.66	84.31	31.20	68.06
	PMET+CFPP	**91.65**	**50.78**	14.98	**52.47**	**89.20**	**84.81**	30.65	**68.22**
	AlphaEdit	76.26	36.88	22.12	45.09	85.70	81.77	32.15	66.54
	AlphaEdit+CFPP	**76.28**	**36.90**	**22.25**	**45.14**	85.65	**82.03**	**32.17**	**66.62**

Table 4: Performance comparison of CFPP when integrated into PMET and AlphaEdit across CounterFact and ZsRE.

consistent gains for PMET, while maintaining comparable locality. The improvements on AlphaEdit are smaller but still visible on most metrics, indicating that CFPP can complement methods that already use stronger structural constraints. These results show that CFPP serves as a general geometric correction within the locate-then-edit paradigm.

6 Discussion

Our experiments suggest that the trade-off between editing and preservation is not always intrinsic to model editing itself. Instead, it can partly arise from a geometric mismatch within the local closed-form objective. This reframes preservation not merely as a regularization strength to tune, but as a geometric constraint that should be made compatible with the edit objective. CFPP therefore complements prior work by projecting the preservation subspace away from the active edit subspace in the local update. As long as future locate-then-edit methods rely on a preservation covariance, similar preservation–edit overlap may reappear and cause interference.

7 Conclusion

We introduced Conflict-Free Preservation Projection (CFPP), a plug-and-play extension for locate-then-edit model editing. CFPP addresses the preservation–edit conflict that arises when the preservation subspace overlaps with the active edit subspace and suppresses directions needed for the target edit. By projecting the preservation subspace away from the edit subspace before computing the local closed-form update, CFPP preserves the original editing workflow, introduces no trainable parameters, and integrates into multiple locate-then-edit methods. Experiments across representative models and benchmarks show that CFPP improves editing efficacy and generalization while maintaining competitive preservation, highlighting the importance of shaping preservation constraints to avoid suppressing edit-relevant directions.

Limitations

CFPP depends on the quality of the estimated preservation covariance C_p, which is typically approximated using public corpora. Inaccurate covariance estimation may therefore reduce the effectiveness of the projection. In addition, the hard projection variant ($\beta = 1$) completely removes preservation pressure along edit-subspace directions, which may be overly aggressive when some shared directions still encode useful background information. Although the soft projection variant alleviates this issue, it introduces an additional hyperparameter β that requires tuning. Finally, CFPP is designed primarily for massive editing; its behavior under sequential editing, where the edit subspace evolves over time, remains to be studied.

Ethics Considerations

This work aims to improve the reliability of knowledge editing in large language models, but such techniques also raise potential ethical concerns. Because model editing can directly alter factual associations inside a model, careless or malicious use may lead to misinformation, biased outputs, or unintended behavioral changes. Our experiments use publicly available code, datasets, and models for research evaluation in accordance with their intended use, licenses, and terms of use. We therefore recommend applying knowledge editing methods with appropriate safeguards, including careful validation of edited facts, evaluation of side effects, and monitoring of model behavior after editing. We encourage the responsible use of these techniques in transparent and well-controlled settings.

References

Yuchen Cai and Ding Cao. 2024. O-edit: Orthogonal subspace editing for language model sequential editing. *CoRR*, abs/2410.11469.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6491–6506. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 8493–8502. Association for Computational Linguistics.

Yanbo Dai, Zhenlan Ji, Zongjie Li, and Shuai Wang. 2026. EAMET: ROBUST MASSIVE MODEL EDITING VIA EMBEDDING ALIGNMENT OPTIMIZATION. In *The Fourteenth International Conference on Learning Representations*.

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 *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 30–45. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pages 15202–15232. 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. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Knowledge in superposition: Unveiling the failures of lifelong knowledge editing for large language models. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 24086–24094. AAAI Press.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Mingyang Wan, Guojun Ma, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025a. Anyedit: Edit any knowledge encoded in language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Houcheng Jiang, Junfeng Fang, Tianyu Zhang, Baolong Bi, An Zhang, Ruipeng Wang, Tao Liang, and Xiang Wang. 2025b. Neuron-level sequential editing for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 16678–16702. Association for Computational Linguistics.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know *When* language models know? on the calibration of language models for question answering. *Trans. Assoc. Comput. Linguistics*, 9:962–977.

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), Vancouver, Canada, August 3-4, 2017*, pages 333–342. Association for Computational Linguistics.

Qi Li and Xiaowen Chu. 2025. Adaedit: Advancing continuous knowledge editing for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 4127–4149. Association for Computational Linguistics.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. 2024. PMET: precise model editing in a transformer. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18564–18572. AAAI Press.

Sicheng Lyu, Yu Gu, Xinyu Wang, Jerry Huang, Sitao Luan, Yufei Cui, Xiao-Wen Chang, and Peng Lu. 2025. Evoedit: Evolving null-space alignment for robust and efficient knowledge editing. *CoRR*, abs/2510.13851.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pages 15817–15831. PMLR.

Haewon Park, Sangwoo Kim, and Yohan Jo. 2026. SUIT: Knowledge editing with subspace-aware key-value mappings. In *The Fourteenth International Conference on Learning Representations*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. 2025. In-context editing: Learning knowledge from self-induced distributions. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Shanbao Qiao, Xuebing Liu, and Seung-Hoon Na. 2025. Wasserstein distance constraint and parameter sparsification for batched and iterative knowledge editing. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 25019–25028. AAAI Press.

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 2020, Online, November 16-20, 2020*, pages 5418–5426. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

Wei Sun, Tingyu Qu, Mingxiao Li, Jesse Davis, and Marie-Francine Moens. 2025. Mitigating negative interference in multilingual knowledge editing through null-space constraints. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 8796–8810. Association for Computational Linguistics.

Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. 2025. Knowledge decoupling via orthogonal projection for lifelong editing of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 13194–13213. Association for Computational Linguistics.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10222–10240. Association for Computational Linguistics.

Mengqi Zhang, Xiaotian Ye, Qiang Liu, Shu Wu, Pengjie Ren, and Zhumin Chen. 2025. Uncovering overfitting in large language model editing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, and 3 others. 2024. A comprehensive study of knowledge editing for large language models. *CoRR*, abs/2401.01286.

A Subspace Diagnostics and Projection Proof

A.1 Why Principal Angles and Overlap Scores Reflect Subspace Conflict

The preservation covariance C_p is a second-moment matrix constructed from background keys. Its top eigenvectors identify the dominant directions along which the preservation regularizer imposes the strongest penalty. In contrast, the left singular vectors of K_e span the principal directions of the active edit-key subspace. Thus, comparing the top eigenspace of C_p with the leading left singular subspace of K_e measures whether the directions most strongly protected by preservation overlap with the directions required by editing.

Principal angles quantify the geometric separation between two subspaces. If the singular values of $U_p^\top U_e$ are large, the corresponding principal angles are small, meaning that the two subspaces contain nearly aligned directions. In our setting, such alignment indicates that the preservation regularizer is likely to penalize directions used by the edit update. Conversely, larger principal angles indicate that the preservation and edit subspaces are more separated, implying weaker geometric conflict.

The normalized Frobenius overlap score aggregates this alignment across all selected directions:

$$\mathrm{Overlap}(U_p, U_e) = \frac{\|U_p^\top U_e\|_F^2}{r} = \frac{1}{r}\sum_{i=1}^{r}\cos^2\theta_i. \tag{15}$$

This value lies in $[0, 1]$ when U_p and U_e have orthonormal columns and the same rank. It equals 0 when the two subspaces are orthogonal and equals 1 when they coincide. Therefore, the overlap score gives a compact scalar summary of how much the dominant preservation subspace induced by C_p intersects with the edit subspace induced by K_e. A larger score indicates stronger preservation–edit conflict, while a smaller score indicates that the preservation regularizer is less likely to suppress edit-relevant directions.

A.2 Why Projection Preserves K_p Components Orthogonal to K_e

Let $P_e = K_e(K_e^\top K_e)^+ K_e^\top$ be the orthogonal projector onto $\mathrm{span}(K_e)$, and let $P_{\perp e} = I - P_e$ be the projector onto its orthogonal complement. For any background key k_p, decompose it as

$$k_p = k_p^{\parallel} + k_p^{\perp}, \tag{16}$$

where $k_p^{\parallel} \in \mathrm{span}(K_e)$ and $k_p^{\perp} \in \mathrm{span}(K_e)^\perp$. By the properties of orthogonal projection,

$$P_{\perp e}k_p = P_{\perp e}k_p^{\parallel} + P_{\perp e}k_p^{\perp} = 0 + k_p^{\perp} = k_p^{\perp}. \tag{17}$$

Thus, the component orthogonal to $\mathrm{span}(K_e)$ is preserved exactly, while the component aligned with the edit subspace is removed. Applying this projection column-wise to K_p gives

$$P_{\perp e}K_p = K_p^{\perp}. \tag{18}$$

Therefore, the projected preservation covariance becomes

$$\begin{aligned} P_{\perp e}K_pK_p^\top P_{\perp e} &= (P_{\perp e}K_p)(P_{\perp e}K_p)^\top \\ &= K_p^{\perp}(K_p^{\perp})^\top. \end{aligned} \tag{19}$$

This shows that CFPP preserves the covariance induced by the components of K_p orthogonal to the edit subspace, while removing the preservation effect aligned with the active edit subspace.

B Implementation Details

B.1 Numerical Treatment of the Pseudoinverse

CFPP relies on the projector

$$P_e = K_e(K_e^\top K_e)^+ K_e^\top, \tag{20}$$

where $(K_e^\top K_e)^+$ denotes the Moore-Penrose pseudoinverse. In practice, the Gram matrix $K_e^\top K_e$

can be rank-deficient or ill-conditioned, especially when edited keys are highly correlated or when the number of edits is large. Therefore, we do not assume ordinary invertibility.

Our implementation computes the pseudoinverse numerically using a singular-value-based routine with a relative tolerance threshold. Concretely, if

$$K_e^\top K_e = U \Sigma U^\top, \tag{21}$$

then

$$(K_e^\top K_e)^+ = U \Sigma^+ U^\top, \tag{22}$$

where Σ^+ is formed by inverting only those singular values larger than a prescribed tolerance and setting the remaining ones to zero. This treatment removes numerically unstable directions and ensures that the projector remains well-defined even when the edit-key subspace is not full rank. In the implementation, this behavior is controlled by the hyperparameter `cfpp_rtol`, which specifies the relative cutoff used in the pseudoinverse computation.

B.2 Rank-Truncated Edit Subspaces

We support restricting the projection to the dominant edit directions. Instead of using the full column space of K_e, we first compute a truncated basis from its leading singular directions and construct the projector on this reduced subspace. This option is controlled by `cfpp_rank`. When `cfpp_rank` is unset, CFPP uses the full edit-key subspace. When `cfpp_rank = r`, only the top-r principal directions of K_e participate in the projection. We also support percentage-style specifications, so that the effective rank can scale with the number of edits. This design makes CFPP more robust as the editing scale increases, because the raw edit-key matrix can contain weak or noisy directions that should not necessarily be projected out of the preservation term.

C Evaluation Metrics and Datasets

Existing model-editing work uses two related families of evaluation metrics. The first is a *probability-based success* criterion, which checks whether the edited target becomes more likely than the original target. The second is an *accuracy-based* criterion, which checks whether the model explicitly predicts the desired answer. Our evaluation uses the accuracy-based metrics.

Probability-based success metrics. Let x_i denote an evaluation prompt for the i-th edit, let $o_i^\star$ denote the edited target answer, and let o_i^{old} denote the original answer before editing. A standard rewrite or paraphrase success metric is

$$\text{Succ}_{\text{edit}} = \mathbb{E}_i \left[\mathbb{P}_{f_\theta}(o_i^\star \mid x_i) > \mathbb{P}_{f_\theta}(o_i^{\text{old}} \mid x_i) \right]. \tag{23}$$

For locality or specificity-style prompts, the desired inequality is reversed:

$$\text{Succ}_{\text{loc}} = \mathbb{E}_i \left[\mathbb{P}_{f_\theta}(o_i^{\text{old}} \mid x_i) > \mathbb{P}_{f_\theta}(o_i^\star \mid x_i) \right]. \tag{24}$$

These probability-comparison metrics are widely used in prior work.

Accuracy-based metrics. In the main paper, however, we report accuracy-style metrics. For CounterFact and Wiki-recent, efficacy is computed using strict answer retrieval:

$$\text{Eff.} = \mathbb{E}_i \left[o_i^\star = \arg\max_o \ \mathbb{P}_{f_\theta}(o \mid x_i) \right]. \tag{25}$$

Generalization, portability, locality, and specificity are defined analogously by replacing x_i with the corresponding paraphrase, portability, locality, or neighborhood prompt set and averaging the resulting correctness indicators.

CounterFact. For CounterFact, we report **efficacy**, **generalization**, and **specificity** in accuracy form in the main tables. Efficacy measures whether the edited model predicts the target new answer on the original rewrite prompt. Generalization measures the same strict answer-level accuracy on paraphrase prompts associated with the edited fact. Specificity measures answer-level accuracy on neighborhood prompts that should still preserve the original ground-truth answer.

ZsRE. For ZsRE, we also report **efficacy**, **generalization**, and **specificity**, but here each metric is computed as an average token-level hit rate. Let the target answer sequence be $(y_{i,1}^\star, \ldots, y_{i,T_i}^\star)$. Then the token-level accuracy for a prompt x_i is

$$\text{Acc}_{\text{tok}}(x_i) = \frac{1}{T_i} \sum_{t=1}^{T_i} \left[\mathbf{1} \left(y_{i,t}^\star = \arg\max_y \ \mathbb{P}_{f_\theta}(y \mid x_i, y_{i,<t}^\star) \right) \right]. \tag{26}$$

The reported ZsRE efficacy, generalization, and specificity are obtained by averaging $\text{Acc}_{\text{tok}}(x_i)$

Model	Rank	Overlap	θ_{mean}	θ_{max}	θ_{min}
GPT2-XL	1	0.4095	50.21	50.21	50.21
	2	0.2829	64.18	86.98	41.37
	4	0.2189	67.25	88.15	34.91
	8	0.1647	71.33	87.76	32.12
	16	0.1400	73.44	89.82	28.67
	32	0.1268	74.42	89.94	25.11
	64	0.1174	74.83	89.99	19.44
	128	0.1206	73.99	89.90	14.45

Table 5: Rank sweep for GPT2-XL.

Model	Rank	Overlap	θ_{mean}	θ_{max}	θ_{min}
GPT-J-6B	1	0.6123	38.51	38.51	38.51
	2	0.3360	61.28	87.50	35.05
	4	0.2001	69.71	88.59	31.46
	8	0.1654	71.63	89.30	26.90
	16	0.1623	71.07	89.36	22.34
	32	0.1410	72.70	89.65	19.80
	64	0.1257	73.86	89.90	17.00
	128	0.1231	73.91	89.89	14.85

Table 6: Rank sweep for GPT-J-6B.

over rewrite, paraphrase, and neighborhood prompts, respectively. Thus, ZsRE remains accuracy-based, but it is measured at the token level rather than by strict whole-answer exact match.

Wiki-recent. For Wiki-recent, we report **efficacy**, **portability**, and **locality**, again using accuracy-style evaluation. Efficacy measures whether the edited model retrieves the updated target fact on the primary rewrite prompt. Portability measures whether the same updated knowledge transfers correctly to reasoning variants or alias-based prompts associated with the edited fact. Locality measures whether unrelated prompts continue to produce their original correct answers after editing.

Dataset documentation. All datasets used in this work are publicly available English-language knowledge editing benchmarks. CounterFact and ZsRE focus on factual rewriting and question-answer-style knowledge updates, while Wiki-recent focuses on recently changed factual knowledge and evaluates whether updated facts transfer across related prompts without disrupting unrelated knowledge. We do not collect new data or introduce new demographic annotations; the demographic coverage and other dataset-specific properties therefore follow the original benchmark releases.

Model	Rank	Overlap	θ_{mean}	θ_{max}	θ_{min}
Llama3-8B	1	0.0481	77.33	77.33	77.33
	2	0.0397	81.74	89.85	73.63
	4	0.0725	79.69	89.88	58.30
	8	0.0659	80.16	89.82	49.91
	16	0.0607	79.60	89.35	47.43
	32	0.0661	78.79	89.99	42.81
	64	0.0792	77.41	89.92	36.94
	128	0.0757	77.89	89.99	33.15

Table 7: Rank sweep for Llama3-8B.

D Additional Experiments

D.1 Rank Selection

In Section 4.1, the overlap and principal-angle computations both depend on the rank r. In what follows, we analyze how this choice affects the stability of the diagnostics. Specifically, we report the overlap score and principal-angle statistics (mean, max, and min) as r varies. The results in Tables 5–7 show a clear saturation trend by $r = 32$: the overlap and θ_{mean} change only mildly beyond this point, and θ_{max} is already near its ceiling. Increasing r further mainly expands the tail of small principal angles (reflected in a gradual decrease of θ_{min}), which adds computational cost but does not materially alter the overlap diagnostics used in our analysis. Therefore, we set $r = 32$ for the overlap and principal-angle diagnostics as a stable and cost-effective choice.

D.2 Additional Scaling Plots

To complement the main results, we provide additional scaling plots generated using the same evaluation protocol as Figure 4. These figures show how efficacy, generalization, and specificity evolve as the number of edits increases from 1 to 10^4 across additional model–dataset combinations, verifying that the trends discussed in the main text are not specific to the Llama3-8B CounterFact setting. Figures 5–9 report the scaling behavior for GPT2-XL, GPT-J-6B, and Llama3-8B on additional benchmarks. Across these supplementary settings, MEMIT-CFPP remains competitive with strong baselines and generally exhibits the same qualitative trade-off pattern observed in the main text.

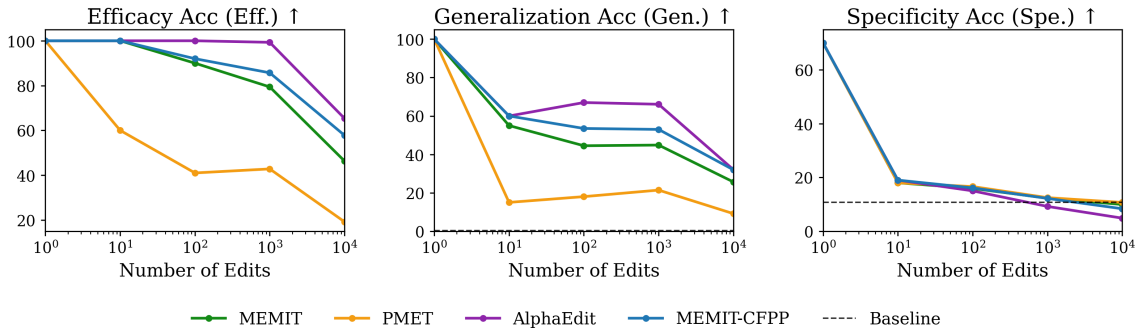

Figure 5: Scaling results for GPT2-XL on CounterFact.

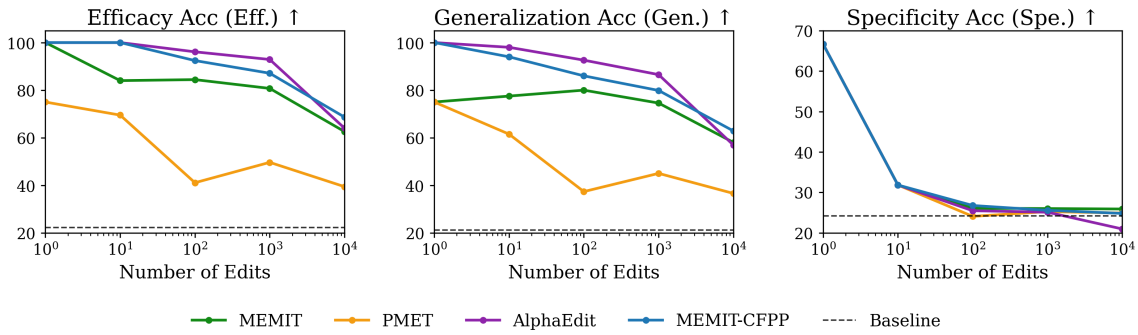

Figure 6: Scaling results for GPT2-XL on ZsRE.

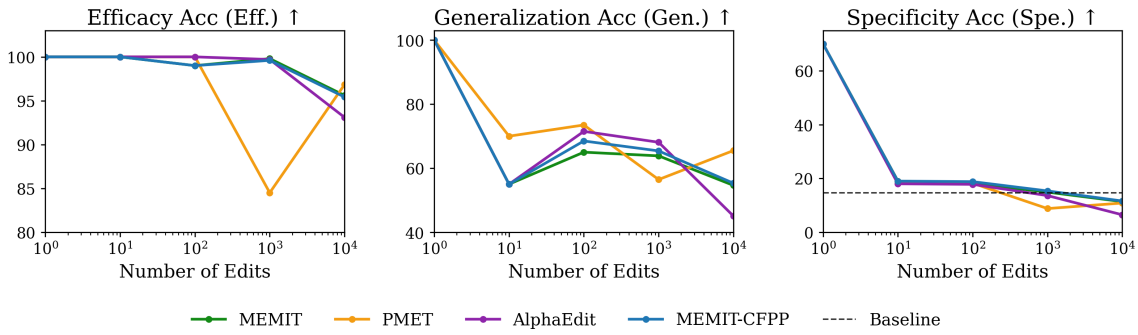

Figure 7: Scaling results for GPT-J-6B on CounterFact.

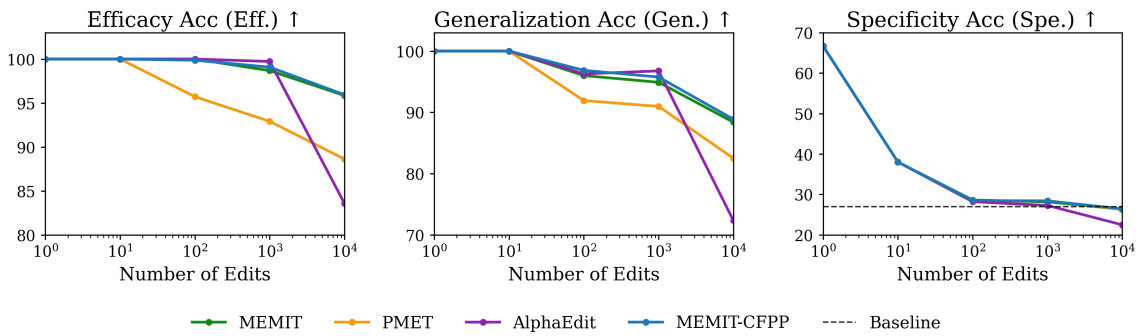

Figure 8: Scaling results for GPT-J-6B on ZsRE.

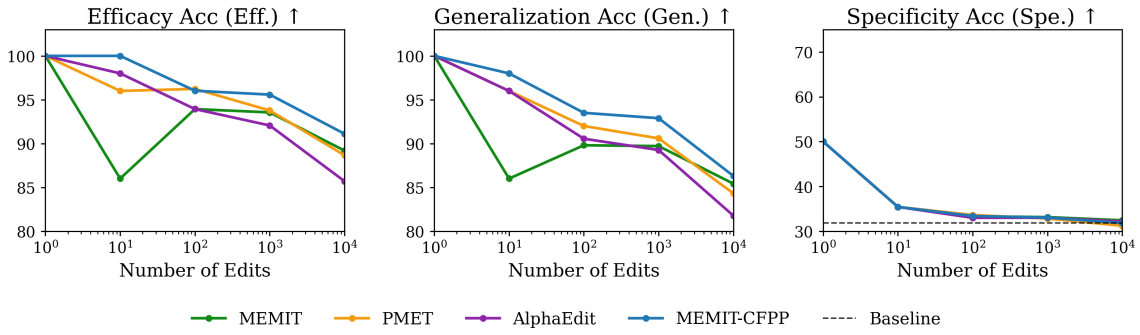

Figure 9: Scaling results for Llama3-8B on ZsRE.