SHAPE THE RESIDUALS, NOT THE UPDATE: SPECTRALLY CONSTRAINED BATCH EDITING FOR SUBJECT-CENTRIC KNOWLEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Locate-then-edit methods dominate batch model editing, yet they are evaluated almost exclusively on subject-diverse benchmarks where edits concern unrelated entities. Real-world updates are often subject-centric: many facts about the same entity change together. On HetionetEdit, a benchmark we construct from a biomedical knowledge graph, representative methods lose roughly 30 Efficacy points. We trace the failure not to the optimized residuals, which remain near-perfect, but to their internalization into the weights. A spectral analysis shows why: soft- and hard-constraint updates are one operator with different filters, and on directions where keys are nearly indistinguishable the soft filter discards the residual while the hard filter amplifies it without bound. Same-subject edits create exactly such directions while demanding different residuals. We therefore constrain the residuals rather than the update: a single reparameterization keeps them in the subspace the keys can express and bounds their separation by that of the keys. The resulting method, Spectrally Constrained Residual Optimization (SCRO), improves the average editing score on HetionetEdit from 36.5 to 50.5 across four LLMs, remains highly competitive on CounterFact and ZsRE, and preserves general language understanding. Code is available at `https://anonymous.4open.science/status/SCRO-ED97`.

1 INTRODUCTION

Large language models store a vast amount of factual knowledge in their parameters, and part of it inevitably becomes outdated or is found to be wrong. Model editing (Meng et al., 2022; 2023; Levy et al., 2017) corrects individual facts without retraining. This work focuses on the locate-then-edit paradigm (Meng et al., 2022; 2023; Dai et al., 2025; Fang et al., 2025), the dominant approach to *batch* editing: it treats an MLP output projection as a key-value memory and proceeds in two stages, first optimizing a target residual for each edit with the weights frozen, and then internalizing all residuals into the weights at once through a closed-form update.

Batch editing is evaluated almost exclusively on CounterFact (Meng et al., 2022) and ZsRE (Levy et al., 2017). Both are sampled from Wikipedia, and their edits are *subject-diverse*: each fact concerns a different entity, and edits within a batch are essentially unrelated. Real-world updates rarely look like this. When a biomedical database is revised, for instance, a single drug may simultaneously acquire new targets, indications, and side effects, so that many facts about the same entity change together. We refer to this regime as *subject-centric multi-relation* (SCMR) batch editing and, to study it, construct **HetionetEdit** from the biomedical knowledge graph Hetionet (Himmelstein et al., 2017), in which most subjects carry several edited relations. On HetionetEdit, two representative methods, MEMIT (Meng et al., 2023) (soft constraint) and EMMET (Gupta et al., 2024) (hard constraint), lose roughly 30 Efficacy points relative to CounterFact and ZsRE under identical model, batch size, and algorithm.

Where does the degradation come from? Recall that the locate-then-edit pipeline consists of two stages: *residual optimization*, which learns a target residual for each edit with the weights frozen, and *weight internalization*, which writes all residuals into the weights through a closed-form update. Since the two stages are separable, we evaluate each on its own. The residuals themselves are not the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

problem: injected directly into the frozen forward pass, they achieve near-perfect Efficacy regardless of subject structure. The failure arises when the residuals are internalized into the weights, and it becomes more severe as more relations of the same subject are edited together. Decomposing the internalization error into how well the update reproduces the residuals and how much it disturbs other hidden states reveals that the two methods fail in mirror image: MEMIT leaves other states largely intact but stops reproducing the residuals, whereas EMMET reproduces them exactly at the cost of increasingly disturbing other states.

A spectral analysis explains why. In a suitably whitened coordinate system, the soft and hard updates turn out to be the same operator equipped with different filters, and the filters disagree only on directions along which the editing keys are nearly indistinguishable. On such a direction, the soft filter discards the residual and the hard filter realizes it at a gain that grows as the keys become closer; no filter can do both, because the trade-off is fixed by how much residual is demanded along how weak a direction. SCMR amplifies exactly this configuration: edits on the same subject read their keys at the same subject token, so the keys nearly coincide, while the residuals must point to different objects. Since prior methods optimize each residual independently, with no knowledge of the key structure, nothing prevents large residual differences from landing on tiny key differences.

The diagnosis points to the remedy. The keys and the background statistics are fixed once the batch is given, but the residuals are a non-unique product of optimization with considerable slack. We therefore ask what the residuals must satisfy for the hard update, the only one that fits exactly, to also be stable, and obtain two conditions: *feasibility*, the residuals lie in the subspace the keys can express, so that the update asks for nothing it cannot realize; and *stability*, the residuals separate two keys no more than the keys themselves are separable. Both are imposed by construction through a single reparameterization that is optimized jointly across the batch. The resulting method, **Spectrally Constrained Residual Optimization (SCRO)**, changes only how residuals are computed and leaves the closed-form update untouched.

Across four LLMs, SCRO raises the average editing score on the SCMR benchmark HetionetEdit by 14.03 points over the strongest baseline, remains highly competitive on the subject-diverse benchmarks CounterFact and ZsRE, and leaves general language understanding essentially intact.

Our contributions are:

- We construct HetionetEdit, a benchmark for subject-centric multi-relation batch editing built from a biomedical knowledge graph, and show that representative locate-then-edit methods degrade sharply on it.
- We localize the failure to weight internalization rather than residual optimization, and give a unified spectral account in which the soft and hard formulations are one operator with two filters bound by a fitting–stability trade-off that SCMR intensifies.
- We derive two conditions under which residuals can be internalized exactly and stably, and propose SCRO, a reparameterized residual optimization that satisfies both by construction.
- Experiments on four LLMs (Qwen2.5-7B, LLaMA-3.1-8B, DeepSeek-7B, Falcon-7B) with $10{,}000$ edits show that SCRO improves the average editing score on the SCMR benchmark HetionetEdit from 36.5 (best baseline) to 50.5, remains highly competitive on the subject-diverse benchmarks CounterFact and ZsRE, preserves downstream GLUE performance close to the unedited model.

2 Preliminaries

2.1 Factual Knowledge as Key-Value Associations

Let $\boldsymbol{h}^l$ denote the hidden representation at layer l of a decoder-only language model. Abstracting away normalization, residual ordering, and bias terms, a decoder layer updates the representation through self-attention (Attn) and multi-layer perceptron (MLP) modules:

$$\boldsymbol{h}^l = \boldsymbol{h}^{l-1} + \boldsymbol{a}^l + \boldsymbol{m}^l, \qquad \boldsymbol{a}^l = \mathrm{Attn}^l(\boldsymbol{h}^{l-1}), \qquad \boldsymbol{m}^l = \mathrm{MLP}^l(\boldsymbol{a}^l + \boldsymbol{h}^{l-1}). \tag{1}$$

Following the key-value interpretation of MLPs (Kohonen, 1972; Geva et al., 2021), we write the MLP output projection as

$$\underbrace{\boldsymbol{m}^l}_{\boldsymbol{v}} = \boldsymbol{W}^l_{\text{out}} \underbrace{\sigma\big(\boldsymbol{W}^l_{\text{in}}(\boldsymbol{a}^l + \boldsymbol{h}^{l-1})\big)}_{\boldsymbol{k}}, \tag{2}$$

where $\boldsymbol{W}^l_{\text{in}}$ and $\boldsymbol{W}^l_{\text{out}}$ are the input and output projections and $\sigma(\cdot)$ is the activation. The activated intermediate representation $\boldsymbol{k}$ serves as a *key* and the output $\boldsymbol{v}$ as its *value*. For a factual association (s, ρ, o) with subject s, relation ρ, and object o, the key encodes the subject–relation pattern (s, ρ) and the value promotes the object o (Meng et al., 2022; 2023); in practice the key is read at the last token of the subject. For example, *"Michael Jordan played for the Bulls"* has $s =$ "Michael Jordan", $\rho =$ "played for", $o =$ "Bulls".

2.2 Batch Model Editing

Problem Setup. Model editing modifies selected factual associations without retraining. We follow the locate-then-edit paradigm, which identifies the target layers by causal tracing (Appendix A.2) and treats their MLP output projection $\boldsymbol{W}_0 \in \mathbb{R}^{d_{\text{v}} \times d_{\text{k}}}$ as an editable memory. The goal is an update $\boldsymbol{\Delta}$ such that $\boldsymbol{W}^* = \boldsymbol{W}_0 + \boldsymbol{\Delta}$ maps the key of (s, ρ) to a value $\boldsymbol{v}^* = \boldsymbol{W}^*\boldsymbol{k}$ that promotes a new object o^* (Meng et al., 2023; Fang et al., 2025; Dong et al., 2025; Dai et al., 2025). For a batch of N edits, the keys and target values are stacked as $\boldsymbol{K} = [\boldsymbol{k}_1, \ldots, \boldsymbol{k}_N] \in \mathbb{R}^{d_{\text{k}} \times N}$ and $\boldsymbol{V}^* = [\boldsymbol{v}^*_1, \ldots, \boldsymbol{v}^*_N] \in \mathbb{R}^{d_{\text{v}} \times N}$, and the *desired residual* is $\boldsymbol{R} = \boldsymbol{V}^* - \boldsymbol{W}_0\boldsymbol{K} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N]$. To preserve unrelated knowledge, a background key matrix $\boldsymbol{K}_0$ is sampled and summarized by $\boldsymbol{C}_0 = \boldsymbol{K}_0\boldsymbol{K}_0^\top$. With far more background samples than d_{k}, $\boldsymbol{C}_0$ is positive definite in practice, and we assume $\boldsymbol{C}_0 \succ 0$ throughout.

Two Formulations. The *soft-constraint* formulation, exemplified by MEMIT (Meng et al., 2023), balances preservation against fitting:

$$\min_{\boldsymbol{\Delta}} \|\boldsymbol{\Delta}\boldsymbol{K}_0\|_F^2 + \|\boldsymbol{\Delta}\boldsymbol{K} - \boldsymbol{R}\|_F^2 \implies \boldsymbol{\Delta}_S = \boldsymbol{R}\boldsymbol{K}^\top\big(\boldsymbol{C}_0 + \boldsymbol{K}\boldsymbol{K}^\top\big)^{-1}. \tag{3}$$

The *hard-constraint* formulation, exemplified by EMMET (Gupta et al., 2024), enforces fitting exactly:

$$\min_{\boldsymbol{\Delta}} \|\boldsymbol{\Delta}\boldsymbol{K}_0\|_F^2 \text{ s.t. } \boldsymbol{\Delta}\boldsymbol{K} = \boldsymbol{R} \implies \boldsymbol{\Delta}_H = \boldsymbol{R}\big(\boldsymbol{K}^\top\boldsymbol{C}_0^{-1}\boldsymbol{K}\big)^{-1}\boldsymbol{K}^\top\boldsymbol{C}_0^{-1}. \tag{4}$$

Two-Stage Pipeline. Both closed forms consume the same three quantities, and editing naturally splits into two stages. **Stage-I (Residual Optimization, RO)** produces $\boldsymbol{C}_0$, $\boldsymbol{K}$, and $\boldsymbol{R}$. The first two are fixed once the batch is given: $\boldsymbol{C}_0$ is pre-sampled and $\boldsymbol{K}$ is read from a forward pass. In contrast, $\boldsymbol{R}$ is obtained by optimization and is non-unique. Each $\boldsymbol{r}_i$ acts as an intervention on the frozen model: added to the MLP output at the key position; it is optimized so that the forward pass assigns high probability to o^*_i under a locality regularizer (Meng et al., 2023; Gupta et al., 2024) (Appendix A.3). The $\boldsymbol{r}_i$ are optimized *independently*, unaware of the closed form that will later consume $\boldsymbol{R}$. **Stage-II (Weight Internalization, WI)** computes $\boldsymbol{\Delta}$ from these quantities via equation 3 or equation 4 and adds it to $\boldsymbol{W}_0$, so that the edited knowledge is carried by the weights rather than by an external injection.

3 Why Batch Editing Fails under SCMR

3.1 Subject-Centric Multi-Relation Batch Editing

Existing batch editing methods are evaluated on CounterFact (Meng et al., 2022) and ZsRE (Levy et al., 2017), both derived from Wikipedia and *subject-diverse*: edits are largely independent and rarely share a subject. Real-world batch updates, however, often originate from databases or knowledge graphs, where one entity is updated along many relations at once (e.g., a drug's targets, indications,

Method	CounterFact	ZsRE	HetionetEdit
MEMIT	99.2	96.8	**69.8**
EMMET	99.2	96.4	**65.4**

Table 1: Eff. (%) on subject-diverse (CounterFact, ZsRE) and SCMR (HetionetEdit) data; Qwen2.5-7B, batch size 500.

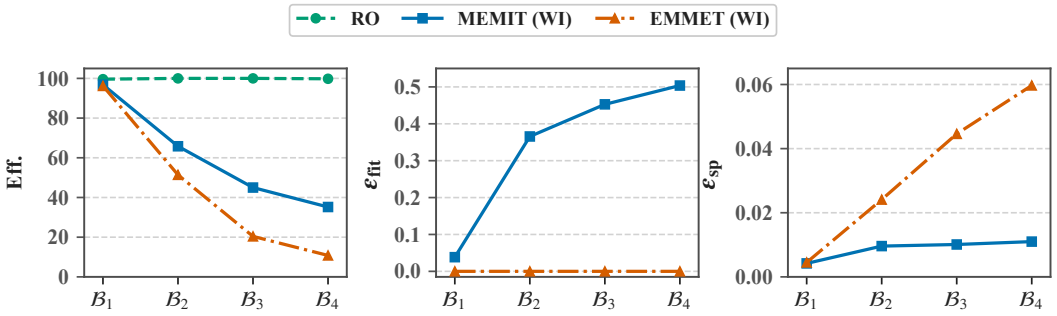


Figure 2: **SCMR breaks internalization.** On controlled subsets $\mathcal{B}_1, \ldots, \mathcal{B}_4$: (a) RO vs. WI Efficacy; (b) fitting error ϵ_{fit}; (c) spillover ϵ_{sp}. Qwen2.5-7B, batch size 500.

and side effects). We term this setting *subject-centric multi-relation* (SCMR) batch editing. For a batch $\mathcal{B} = \{(s_i, \rho_i, o_i^*)\}_{i=1}^N$, let

$$m(s) = \left|\{\rho_i : (s_i, \rho_i, o_i^*) \in \mathcal{B},\ s_i = s\}\right| \tag{5}$$

be the number of relations edited on subject s. Subject-diverse dataset has $m(s) = 1$ for almost all subjects; SCMR dataset has $m(s) > 1$ for most. To study SCMR we construct **HetionetEdit** from the biomedical knowledge graph Hetionet (Himmelstein et al., 2017) (Appendix B). Figure 1 shows that 96% of subjects in CounterFact and 93% in ZsRE have $m(s) = 1$, against only 14% in HetionetEdit, whose distribution peaks at $m(s) = 3$.

We evaluate MEMIT (soft) and EMMET (hard) on the three datasets with Qwen2.5-7B (Yang et al., 2024), batch size 500, and the standard Efficacy (Eff.) metric (Appendix D.3). Table 1 shows both methods near-perfect on CounterFact and ZsRE but losing about 30 points on HetionetEdit. Model, batch size, and algorithms are identical; only the subject structure differs. *Where does the degradation come from?*

3.2 The Failure Lies in Internalization, Not in the Residuals

Two-stage evaluation. To isolate the effect of $m(s)$, we build subsets $\mathcal{B}_1, \ldots, \mathcal{B}_4$ of HetionetEdit in which every subject satisfies $m(s) = i$. We evaluate the two stages of Section 2.2 separately. **RO Eff.** injects each optimized $\boldsymbol{r}_i$ into the frozen forward pass at its key position and measures the quality of $\boldsymbol{R}$ itself; **WI Eff.** evaluates the model after $\boldsymbol{\Delta}$ is added to $\boldsymbol{W}_0$, the standard protocol. Their gap is the cost of internalization. In Figure 2(a), RO Eff. is uniformly high (99.8%) across subsets: the optimizer finds residuals that produce the target objects regardless of subject structure. WI Eff. matches RO on $\mathcal{B}_1$ but declines monotonically with i for both methods, reaching 35.2% (MEMIT) and 10.8% (EMMET) on $\mathcal{B}_4$. The failure lies not in *optimizing* $\boldsymbol{R}$ but in *writing it into the weights*.

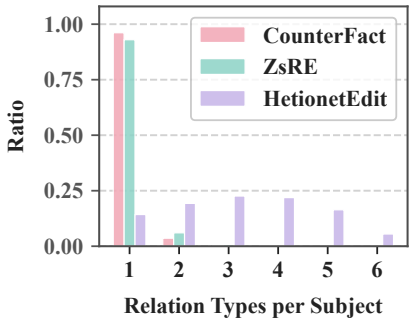


Figure 1: Distribution of $m(s)$ across datasets.

Decomposing the internalization error. Let $\boldsymbol{T} = [\boldsymbol{K}, \boldsymbol{H}_{\text{nk}}]$ collect the hidden states entering $\boldsymbol{W}_0$ for all tokens of the edit samples: the N keys and the remaining *non-key* states $\boldsymbol{H}_{\text{nk}}$, i.e., every other token position. RO perturbs only the key positions, whereas WI applies $\boldsymbol{\Delta}$ everywhere:

$$\text{RO: } [\boldsymbol{W}_0\boldsymbol{K} + \boldsymbol{R},\ \boldsymbol{W}_0\boldsymbol{H}_{\text{nk}}], \qquad \text{WI: } [\boldsymbol{W}_0\boldsymbol{K} + \boldsymbol{\Delta}\boldsymbol{K},\ \boldsymbol{W}_0\boldsymbol{H}_{\text{nk}} + \boldsymbol{\Delta}\boldsymbol{H}_{\text{nk}}]. \tag{6}$$

Their difference, $[\boldsymbol{\Delta}\boldsymbol{K} - \boldsymbol{R},\ \boldsymbol{\Delta}\boldsymbol{H}_{\text{nk}}]$, is the internalization error at the edited layer: the closer $\boldsymbol{\Delta}\boldsymbol{K}$ is to $\boldsymbol{R}$ and $\boldsymbol{\Delta}\boldsymbol{H}_{\text{nk}}$ to $\boldsymbol{0}$, the more closely WI reproduces RO (we ignore the non-linear propagation of these perturbations through subsequent layers). We quantify the two blocks by

$$\epsilon_{\text{fit}} = \frac{\|\boldsymbol{\Delta}\boldsymbol{K} - \boldsymbol{R}\|_F}{\|\boldsymbol{R}\|_F}, \qquad \epsilon_{\text{sp}} = \frac{\|\boldsymbol{\Delta}\boldsymbol{H}_{\text{nk}}\|_2}{\|\boldsymbol{W}_0\boldsymbol{H}_{\text{nk}}\|_2}, \tag{7}$$

	Fitting (residual realized along $\boldsymbol{v}_i$)	**Stability** (amplification bound)
Soft (MEMIT)	fraction $\frac{1}{1+\sigma_i^2}$ of $\boldsymbol{R}\boldsymbol{v}_i$ dropped; $\to$ all as $\sigma_i \to 0$	$\leq \frac{1}{2}\|\boldsymbol{R}\boldsymbol{V}_r\|_2$, independent of σ_i
Hard (EMMET)	exact, independent of σ_i	$\geq \|\boldsymbol{R}\boldsymbol{v}_i\|_2/\sigma_i$; $\to \infty$ as $\sigma_i \to 0$

Table 2: Summary of the behavior of the two filters on a weak direction $\boldsymbol{v}_i$ ($\sigma_i \ll 1$).

the *fitting error* and *spillover*. Figure 2(b,c) shows the two methods failing in complementary ways. MEMIT's spillover stays nearly flat, but its fitting error grows from 0.03 to 0.50: the soft constraint gives up on the residuals. EMMET's fitting error is almost zero, yet its spillover grows 15-fold and its Eff. falls the furthest: the hard constraint is met at the price of perturbing every other hidden state. Both failure modes trace to the closed-form solutions, which we analyze next.

3.3 Why SCMR Breaks Internalization: A Spectral View

One operator, two filters. Let $\tilde{\boldsymbol{K}} = \boldsymbol{C}_0^{-1/2}\boldsymbol{K}$ be the whitened key matrix with compact SVD $\tilde{\boldsymbol{K}} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$, $\boldsymbol{\Sigma}_r = \mathrm{diag}(\sigma_1 \geq \cdots \geq \sigma_r > 0)$, $r = \mathrm{rank}(\tilde{\boldsymbol{K}})$; a small σ_i indicates a combination of keys that is nearly indistinguishable. Both updates share one form (Appendix C.1),

$$\boldsymbol{\Delta}_A = \boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_A\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}, \qquad \boldsymbol{\Lambda}_S = \mathrm{diag}\Big(\tfrac{\sigma_i}{1+\sigma_i^2}\Big), \quad \boldsymbol{\Lambda}_H = \mathrm{diag}\Big(\tfrac{1}{\sigma_i}\Big), \tag{8}$$

and differ only in the diagonal filter $\boldsymbol{\Lambda}_A$, with $A = S$ for the soft and $A = H$ for the hard constraint formulation.

Fitting. Using $\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}\boldsymbol{K} = \boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$, we obtain (Appendix C.2):

$$\boldsymbol{\Delta}_A\boldsymbol{K} - \boldsymbol{R} = \underbrace{-\boldsymbol{R}(\boldsymbol{I} - \boldsymbol{V}_r\boldsymbol{V}_r^\top)}_{\textbf{truncation}} \underbrace{- \boldsymbol{R}\boldsymbol{V}_r(\boldsymbol{I} - \boldsymbol{\Lambda}_A\boldsymbol{\Sigma}_r)\boldsymbol{V}_r^\top}_{\textbf{shrinkage}}. \tag{9}$$

The **truncation** term, shared by both filters, is the part of $\boldsymbol{R}$ outside the row space of $\tilde{\boldsymbol{K}}$ and vanishes iff $\boldsymbol{R} = \boldsymbol{R}\boldsymbol{V}_r\boldsymbol{V}_r^\top$. The **shrinkage** term is zero for the hard filter ($\boldsymbol{\Lambda}_H\boldsymbol{\Sigma}_r = \boldsymbol{I}$) and equals $\sum_i \frac{1}{1+\sigma_i^2}(\boldsymbol{R}\boldsymbol{v}_i)\boldsymbol{v}_i^\top$ for the soft filter: on weak directions ($\sigma_i \ll 1$) the soft update drops almost all of $\boldsymbol{R}\boldsymbol{v}_i$. For the same $\boldsymbol{R}$, the soft filter therefore always incurs at least the fitting error of the hard filter, with the excess concentrated on weak directions, consistent with Figure 2(b).

Stability. We assess the effect of an update on non-key states through an upper bound on $\|\boldsymbol{\Delta}_A\boldsymbol{H}_{\mathrm{nk}}\|_2$. Substituting equation 8 and applying sub-multiplicativity of the spectral norm (Appendix C.3),

$$\|\boldsymbol{\Delta}_A\boldsymbol{H}_{\mathrm{nk}}\|_2 \leq \underbrace{\Big\|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_A\Big\|_2}_{\textbf{amplification bound}} \underbrace{\Big\|\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-\frac{1}{2}}\boldsymbol{H}_{\mathrm{nk}}\Big\|_2}_{\textbf{projection magnitude}}. \tag{10}$$

The **projection magnitude** measures how strongly the whitened non-key states $\boldsymbol{C}_0^{-1/2}\boldsymbol{H}_{\mathrm{nk}}$ project onto the directions spanned by the whitened keys $\tilde{\boldsymbol{K}}$. It involves only $\boldsymbol{C}_0$, $\boldsymbol{K}$, and $\boldsymbol{H}_{\mathrm{nk}}$, and is therefore fixed once the batch is given, independent of the filter. The **amplification bound** measures how much the update magnifies these projected components into the output perturbation, and is where the filters differ (Appendix C.3):

$$\|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_S\|_2 \;\leq\; \tfrac{1}{2}\|\boldsymbol{R}\boldsymbol{V}_r\|_2 \quad \text{whereas} \quad \|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_H\|_2 \;\geq\; \max_i \frac{\|\boldsymbol{R}\boldsymbol{v}_i\|_2}{\sigma_i}. \tag{11}$$

The soft filter is uniformly bounded regardless of the spectrum. The hard filter has no such guarantee: on any weak direction ($\sigma_i \ll 1$) that still carries residual energy ($\|\boldsymbol{R}\boldsymbol{v}_i\|_2$ not correspondingly small), its amplification bound grows as $\|\boldsymbol{R}\boldsymbol{v}_i\|_2/\sigma_i$. The hard filter is thus potentially, though not necessarily, less stable than the soft one; whether the bound is actually reached depends on how the residual energy is distributed across the spectrum, consistent with Figure 2(c).

Why SCMR breaks internalization. Table 2 summarizes the behaviour of the two filters on a weak direction $\boldsymbol{v}_i$ ($\sigma_i \ll 1$): each is insensitive to the spectrum on one criterion and fails on the other. Such a direction is harmful only when it also carries residual energy, i.e., when $\|\boldsymbol{R}\boldsymbol{v}_i\|_2$ does not shrink with σ_i. SCMR produces exactly this combination, and in growing quantity: the batch contains many weak directions, and the residual energy on them remains large. Intuitively, consider two edits on the same subject. Their keys are read at the last subject token of prompts sharing that subject, so $\tilde{\boldsymbol{k}}_i - \tilde{\boldsymbol{k}}_j$ is short and therefore lies in the span of weak directions; their residuals target different objects, so $\boldsymbol{r}_i - \boldsymbol{r}_j = \boldsymbol{R}(\boldsymbol{e}_i - \boldsymbol{e}_j)$ is long, which forces $\boldsymbol{R}$ to carry substantial energy on those same weak directions.

4 METHOD

From diagnosis to design. Section 3 traced the WI drop to a mismatch: $\boldsymbol{R}$ are optimized in RO without knowledge of the spectral structure of $\tilde{\boldsymbol{K}}$, leaving the closed-form filter to trade fitting against stability. We reverse the reasoning: rather than asking which filter best absorbs a given $\boldsymbol{R}$, we ask what $\boldsymbol{R}$ must satisfy for the hard filter, the only one that fits exactly, to also be stable, and impose this during RO. By equation 9, the hard filter's fitting error reduces to the truncation term $\boldsymbol{R}(\boldsymbol{I} - \boldsymbol{V}_r\boldsymbol{V}_r^\top)$; by equation 10, its amplification bound is $\|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\|_2$. Both are controlled by requiring

$$\text{(feasibility)}\ \ \boldsymbol{R} = \boldsymbol{R}\boldsymbol{V}_r\boldsymbol{V}_r^\top, \qquad \text{(stability)}\ \ \left\|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\right\|_2 \le \alpha. \tag{12}$$

Feasibility places $\boldsymbol{R}$ in the row space of $\tilde{\boldsymbol{K}}$, so that $\boldsymbol{\Delta}_H\boldsymbol{K} = \boldsymbol{R}$ holds exactly. Stability caps the price of that exactness: since $\|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\|_2 \ge \|\boldsymbol{R}\boldsymbol{v}_i\|_2/\sigma_i$ for every i, it enforces $\|\boldsymbol{R}\boldsymbol{v}_i\|_2 \le \alpha\sigma_i$. Under equation 12, we have

$$\boldsymbol{\Delta}_H\boldsymbol{K} = \boldsymbol{R}, \qquad \|\boldsymbol{\Delta}_H\boldsymbol{H}_{\mathrm{nk}}\|_2 \le \alpha\left\|\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}\boldsymbol{H}_{\mathrm{nk}}\right\|_2. \tag{13}$$

Spectrally Constrained Residual Optimization. Conditions equation 12 preclude the practice of prior work, which optimizes each $\boldsymbol{r}_i$ independently and stacks the results into $\boldsymbol{R}$ (Meng et al., 2023; Dai et al., 2025): a residual matrix assembled this way is unlikely to satisfy either condition. We instead optimize $\boldsymbol{R}$ jointly through a reparameterization that satisfies both by construction. With a learnable $\boldsymbol{Z} \in \mathbb{R}^{d_{\mathrm{v}} \times r}$,

$$\boldsymbol{R} = \boldsymbol{Z}\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top, \qquad \|\boldsymbol{Z}\|_2 \le \alpha. \tag{14}$$

Feasibility follows from $\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{V}_r^\top = \boldsymbol{Z}\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top\boldsymbol{V}_r\boldsymbol{V}_r^\top = \boldsymbol{R}$, and stability from $\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1} = \boldsymbol{Z}$, so the amplification bound equals $\|\boldsymbol{Z}\|_2 \le \alpha$. The norm constraint is enforced by spectral projection after each gradient step. Optimization otherwise follows RO of prior work, using the objective in Eq. equation 18, with one change: each $\boldsymbol{r}_i = \boldsymbol{Z}\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top\boldsymbol{e}_i$ is read from $\boldsymbol{Z}$ rather than being a free variable, and the objective is summed over all edits and back-propagated into $\boldsymbol{Z}$. The resulting $\boldsymbol{R}$ is then substituted with $\boldsymbol{C}_0$ and $\boldsymbol{K}$ into the hard-constraint closed form Eq. equation 4 to obtain $\boldsymbol{\Delta}$.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models and Baselines. We evaluate on four LLMs from different families: Qwen2.5-7B (Yang et al., 2024), LLaMA-3.1-8B (Team, 2024), DeepSeek-7B (Bi et al., 2024), and Falcon-7B (Almazrouei et al., 2023). Baselines include fine-tuning (FT); the locate-then-edit methods MEMIT (Meng et al., 2023), EMMET (Gupta et al., 2024), PMET (Li et al., 2024), and AlphaEdit (Fang et al., 2025); and two improved methods well suited to the SCMR scenario, MEMIT-MERGE (Dong et al., 2025) and EAMET (Dai et al., 2025) (Appendix D.1).

Datasets and Metrics. We use the subject-diverse benchmarks CounterFact (Meng et al., 2022) and ZsRE (Levy et al., 2017), and our SCMR benchmark HetionetEdit (Appendix D.2). Following Dai et al. (2025), we report Efficacy (Eff.; target predicted on the edit prompt), Generalization (Gen.; on paraphrases), Specificity (Spe.; neighboring facts preserved), Fluency (Flu.; generation quality), and their average Avg. over Eff., Gen., and Spe. (Appendix D.3).

Table 3: Performance results of the batch editing task. Here, the abbreviations *Eff.* (Efficacy), *Gen.* (Generalization), *Spe.* (Specificity), *Flu.* (Fluency), and *Avg.* (Average) are employed to denote respective evaluation metrics. Top-performing results are emphasized using **bold** formatting, with secondary superior results distinguished through underlined notation.

Model	Method	HetionetEdit				ZsRE				CounterFact				
		Avg.↑	Eff.↑	Gen.↑	Spe.↑	Avg.↑	Eff.↑	Gen.↑	Spe.↑	Avg.↑	Eff.↑	Gen.↑	Spe.↑	Flu.↑
Qwen2.5-7B	FT	25.97	3.66	3.39	**70.85**	22.19	14.30	13.00	39.28	28.85	18.99	14.14	53.41	**547.11**
	MEMIT	18.92	5.70	1.19	49.88	39.41	48.85	36.51	32.86	**76.20**	94.98	**62.63**	70.98	531.25
	EMMET	16.47	0.14	0.01	49.25	5.92	1.75	1.19	14.82	71.46	89.76	57.86	66.77	518.19
	PMET	24.75	16.08	5.67	52.51	60.91	76.14	66.15	40.44	72.67	89.49	56.26	72.27	529.55
	ALPHAEDIT	20.41	10.00	1.71	49.53	31.75	39.11	29.58	26.56	65.12	79.77	50.65	64.95	500.80
	MEMIT-MERGE	46.15	53.79	**39.83**	44.82	68.16	85.03	79.75	39.71	75.52	95.58	60.16	70.83	530.77
	EAMET	30.76	26.74	17.60	47.94	**73.67**	90.35	83.65	**47.00**	75.66	93.94	58.75	74.29	541.55
	SCRO	**50.80**	**68.50**	36.20	47.70	73.55	**92.10**	**86.05**	42.51	76.16	**98.08**	49.86	**80.53**	546.12
LLaMA-3.1-8B	FT	22.71	0.72	0.75	**66.65**	21.92	12.47	10.62	**42.67**	19.36	1.67	1.44	54.96	434.19
	MEMIT	28.10	18.72	9.71	55.87	64.13	79.38	72.69	40.33	78.34	95.03	**61.68**	78.30	523.87
	EMMET	19.19	6.07	0.58	50.91	20.86	27.06	19.33	16.20	72.70	90.24	53.59	74.27	507.44
	PMET	29.45	20.33	12.15	55.86	61.50	75.58	71.48	37.45	75.72	90.64	58.91	77.62	518.61
	ALPHAEDIT	20.14	1.75	1.19	57.49	54.26	64.70	57.81	40.26	60.00	61.72	32.69	**85.60**	536.47
	MEMIT-MERGE	33.89	29.13	22.99	49.54	**70.35**	87.86	**83.10**	40.09	77.59	95.45	58.80	78.52	523.35
	EAMET	26.26	14.86	11.56	52.36	59.96	73.36	68.78	37.73	69.88	85.18	55.45	69.01	484.16
	SCRO	**50.65**	**68.70**	**34.49**	48.76	70.01	**88.08**	80.52	41.42	**80.18**	**98.85**	56.53	85.16	**540.70**
DeepSeek-7B	FT	22.63	1.57	1.55	**64.77**	22.14	15.00	12.28	39.14	28.48	3.53	2.50	79.42	**519.35**
	MEMIT	19.33	1.45	0.96	55.58	44.19	54.61	38.53	39.44	63.15	68.98	40.69	79.78	511.83
	EMMET	23.38	9.64	4.91	55.58	55.10	73.59	54.25	37.45	52.92	66.24	26.65	65.86	480.30
	PMET	18.70	0.52	0.32	55.25	29.12	28.67	19.80	38.90	52.78	49.41	26.60	**82.34**	518.98
	ALPHAEDIT	22.86	9.30	4.30	54.97	53.53	70.24	55.23	35.13	42.51	50.09	14.31	63.13	469.31
	MEMIT-MERGE	25.15	15.68	9.74	50.03	41.59	49.71	35.55	**39.51**	66.08	74.58	44.16	79.49	512.50
	EAMET	27.26	20.87	14.50	46.41	**65.90**	86.68	**72.73**	38.30	73.16	88.63	55.87	74.98	504.00
	SCRO	**49.80**	**71.46**	**33.35**	44.58	64.34	**87.09**	68.69	37.23	**78.39**	**98.32**	**57.75**	79.09	506.08
Falcon-7B	FT	24.56	3.91	3.86	**65.92**	19.53	13.64	12.68	32.28	28.42	16.53	13.54	55.19	166.98
	MEMIT	21.22	6.27	4.15	53.24	61.99	82.50	69.54	33.94	76.97	91.17	61.58	78.16	523.80
	EMMET	18.55	4.16	0.69	50.80	47.55	66.38	51.37	24.91	72.67	89.54	57.60	70.88	495.17
	PMET	20.18	4.17	2.93	53.45	52.63	68.92	55.34	33.63	71.37	83.73	51.77	78.62	523.40
	ALPHAEDIT	19.44	5.06	1.03	52.22	41.91	58.80	44.39	22.55	73.46	90.63	57.90	71.84	501.22
	MEMIT-MERGE	40.66	48.69	33.44	39.85	**69.67**	**92.56**	**82.57**	33.88	76.68	93.29	59.31	77.45	523.47
	EAMET	29.18	20.56	18.41	48.57	66.34	87.02	79.72	32.27	78.62	95.45	**64.55**	75.86	**526.70**
	SCRO	**50.69**	**73.44**	**36.88**	41.74	67.23	89.50	78.08	**34.11**	**79.83**	**98.73**	61.30	**79.47**	512.99

5.2 Batch Editing Performance

Table 3 reports the batch editing performance across four backbone models and three datasets with batch size 10000. We provide hyperparameter settings and hardware configuration in Appendix D.4.

Finding 1: SCRO comprehensively outperforms baselines in subject-centric multi-relation batch editing. SCRO achieves the best overall editing performance on HetionetEdit across all models. Although the improved batch editing methods well suited to the SCMR scenario, such as MEMIT-MERGE and EAMET, outperform classic batch editing methods, they still lag far behind SCRO. Specifically, in terms of Avg., which measures the overall editing performance, SCRO attains an average of 50.49% over four models, surpassing MEMIT-MERGE (36.46%) by 14.03% and EAMET (28.37%) by 22.12%. The above results clearly demonstrate that SCRO's spectrally constrained residual optimization offers a distinct advantage in the SCMR data regime, effectively mitigating the performance degradation caused by subject-centric multi-relation editing.

Finding 2: SCRO remains stable and competitive in subject-diverse factual batch editing. On CounterFact and ZsRE, baseline methods show strong overall performance but exhibit considerable instability across datasets and models. For example, MEMIT-MERGE achieves the worst Avg. (41.59%) on ZsRE with DeepSeek-7B yet the best Avg. (70.35%) with LLaMA-3.1-8B. In contrast, SCRO consistently ranks first or second across all datasets and models. Concretely, on CounterFact, SCRO attains an average Avg. of 78.64%, surpassing the next-best baseline MEMIT-MERGE (73.97%) by 4.67%. On ZsRE, the next-best baseline shifts to EAMET, with SCRO reaching 68.78% and outperforming it by 2.31%. The gains are steady but moderate, which suggests that on subject-diverse benchmarks the spectral constraints addressed by SCRO are not the dominant bottleneck; other factors, such as the propagation of updates through subsequent layers, likely contribute to

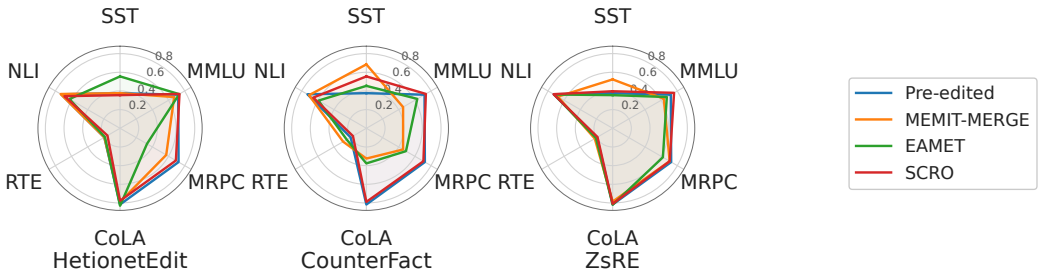


Figure 3: General capability tests on Qwen2.5-7B after batch editing. Radar plots compare downstream performance on SST, MMLU, MRPC, CoLA, RTE, and NLI across HetionetEdit, CounterFact, and ZsRE.

the remaining gap. Nevertheless, SCRO delivers stable and competitive performance across both conventional and SCMR settings.

5.3 General Capability Tests

We further evaluate the edited models on six downstream tasks from the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) (see Appendix D.5 for details). Figure 3 reports the results for the improved batch editing methods well suited to the SCMR scenario, MEMIT-MERGE and EAMET, and our method SCRO, on the Qwen2.5-7B model after editing on the HetionetEdit, CounterFact, and ZsRE datasets.

Finding 3: SCRO preserves strong general language understanding ability. Across the three editing datasets, SCRO performs close to the pre-edited model on most downstream tasks, suggesting that our edits do not sacrifice general capabilities. In contrast, MEMIT-MERGE and EAMET exhibit varying degrees of average downstream task degradation on HetionetEdit and CounterFact, although they remain comparable to the pre-edited model on ZsRE. These results further confirm that SCRO effectively preserves general language understanding while achieving superior editing performance, highlighting the advantage of its solvability-aware design.

5.4 Ablation Study

SCRO imposes two conditions on $\boldsymbol{R}$, and Section 4 argues that neither suffices alone. Table 4 tests this by removing each in turn. *Feasibility only* places $\boldsymbol{R}$ in the row space of $\tilde{\boldsymbol{K}}$ but leaves $\|\boldsymbol{Z}\|_2$ unbounded, so the hard filter fits exactly while its amplification bound is uncontrolled; the resulting perturbation of non-key states, including the tokens of the edit prompts themselves, drives Eff. to 0.75%. *Stability only* bounds the amplification but lets $\boldsymbol{R}$ leave the row space, so the component outside it is truncated by the hard filter and never written into the weights; Eff. reaches only 11.97%. Both variants retain Spe. comparable to the full method, which is uninformative on its own: an update that fails to take effect trivially preserves neighboring facts. Only the combination yields a residual matrix that is both realizable and stably internalizable, raising Avg. from below 25 to 50.80. The two constraints are thus complementary, matching the fitting–stability trade-off of Section 3.3: each alone removes one failure mode and leaves the other intact.

Variant	Avg.	Eff.	Gen.	Spe.
Feasibility only	17.42	0.75	0.70	50.81
Stability only	24.71	11.97	8.45	53.70
SCRO (both)	50.80	68.50	36.20	47.70

Table 4: Ablation of the two constraints on HetionetEdit (Qwen2.5-7B, 10,000 edits).

6 Related Works

Model Editing. Knowledge editing aims to modify specific factual associations in language models without full retraining. Early methods learn auxiliary editors or hyper-networks to predict pa-

rameter updates (Mitchell et al., 2022a;b). More recent locate-then-edit methods directly modify internal parameters associated with factual recall. ROME (Meng et al., 2022) identifies causal MLP modules and edits factual associations through rank-one updates, while MEMIT (Meng et al., 2023) extends this paradigm to mass editing by applying closed-form updates over multiple layers. Follow-up methods further improve editing precision and preservation. For example, PMET (Li et al., 2024) refines hidden-state optimization for FFN updates, and ALPHAEDIT (Fang et al., 2025) constrains editing perturbations within the null space of preserved knowledge. Our work follows the locate-then-edit paradigm but focuses on the batch-level coupling.

Mitigate interference. Prior studies have identified interference in batch editing that degrades performance. To mitigate this, MEMIT-MERGE (Dong et al., 2025) addresses same-subject conflicts by merging value computations for shared subjects, and EAMET (Dai et al., 2025) reduces interference in massive editing by aligning key and residual embedding spaces. Nevertheless, these heuristic approaches offer only partial mitigation and cannot fundamentally eliminate interference.

Benchmarks. Existing editing methods are commonly evaluated on broad-coverage benchmarks such as CounterFact (Meng et al., 2022) and ZsRE (Levy et al., 2017). However, they do not fully capture domain-concentrated updates, where edit requests often come from coherent knowledge sources and are organized around recurring entities. Our work studies this subject-centered multi-relation setting through HetionetEdit.

7 Conclusion

We showed that locate-then-edit methods fail on subject-centric multi-relation updates not because the residuals are poor but because they cannot be internalized: near-identical keys must carry different residuals, forcing a fitting–stability trade-off that no closed-form filter escapes. SCRO resolves this by constraining the residuals to be feasible and stable under the hard filter, improving editing on HetionetEdit across four LLMs while remaining competitive on subject-diverse benchmarks.

Limitations

While this study provides valuable insights into SCMR batch model editing, several limitations should be acknowledged. First, our evaluation mainly focuses on HetionetEdit, CounterFact, and ZsRE, with HetionetEdit centered on biomedical knowledge. Although this enables controlled analysis of subject-centric multi-relation batch editing, it may not fully reflect domain-concentrated updates in other real-world scenarios. Second, our experiments are limited to representative locate-then-edit methods and 7B-scale backbone models. Future work can examine whether SCRO generalizes to larger models, instruction-tuned models, and more diverse architectures. Third, HetionetEdit retains one designated target object for subject-relation pairs that may admit multiple valid objects. Therefore, the benchmark evaluates controlled insertion of designated associations rather than exhaustive biomedical question answering. Future work should incorporate multi-answer annotations and set-based evaluation to better capture relations with multiple correct objects.

Ethical Considerations

This work studies factual model editing in a controlled research setting. The benchmark is intended to evaluate editing behavior and interference in batch updates, rather than to certify the correctness or safety of edited model outputs. Edited outputs should therefore not be treated as verified knowledge without external validation.

More generally, model editing methods can be misused to insert inaccurate, biased, or harmful associations into language models, particularly when edits are applied at scale or derived from untrusted sources. Practical deployment should include trusted update sources, fact validation, logging and provenance of parameter changes, and mechanisms for auditing or reverting edits. Our experiments analyze batch-editing behavior and do not evaluate these deployment safeguards.

AI USE STATEMENT

Generative AI tools were used to assist with code implementation and debugging, experiment planning, figure preparation, and language editing. All AI-assisted code, experimental results and manuscript content were manually inspected and verified by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide an anonymous implementation of SCRO at `https://anonymous.4open.science/status/SCRO-ED97`. The complete derivations and proofs of the spectral formulations are provided in Appendix C. The construction procedure, templates, and statistics of HetionetEdit are documented in Appendix B. Baseline definitions, datasets, evaluation metrics, hyperparameters, and hardware configurations are reported in Appendix D.

REFERENCES

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models. *CoRR*, abs/2311.16867, 2023. doi: 10.48550/ARXIV.2311.16867. URL `https://doi.org/10.48550/arXiv.2311.16867`.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *TAC*. NIST, 2009.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, Alex X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek LLM: scaling open-source language models with longtermism. *CoRR*, abs/2401.02954, 2024. doi: 10.48550/ARXIV.2401.02954. URL `https://doi.org/10.48550/arXiv.2401.02954`.

Yanbo Dai, Zhenlan Ji, Zongjie Li, and Shuai Wang. Eamet: Robust massive model editing via embedding alignment optimization. *arXiv preprint arXiv:2505.11876*, 2025.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *IWP@IJCNLP*. Asian Federation of Natural Language Processing, 2005.

Zilu Dong, Xiangqing Shen, and Rui Xia. Memit-merge: Addressing memit's key-value conflicts in same-subject batch editing for llms. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pp. 7952–7960. Association for Computational Linguistics, 2025. URL `https://aclanthology.org/2025.findings-acl.415/`.

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 5484–5495. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.446. URL https://doi.org/10.18653/v1/2021.emnlp-main.446.

Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. A unified framework for model editing. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 15403–15418. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.903. URL https://doi.org/10.18653/v1/2024.findings-emnlp.903.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net, 2021.

Daniel Scott Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E Baranzini. Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *elife*, 6:e26726, 2017.

Teuvo Kohonen. Correlation matrix memories. *IEEE Trans. Computers*, 21(4):353–359, 1972. doi: 10.1109/TC.1972.5008975. URL https://doi.org/10.1109/TC.1972.5008975.

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

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. PMET: precise model editing in a transformer. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 18564–18572. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29818. URL https://doi.org/10.1609/aaai.v38i17.29818.

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

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pp. 1631–1642. ACL, 2013.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR (Poster)*. OpenReview.net, 2019.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguistics*, 7:625–641, 2019.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, pp. 1112–1122. Association for Computational Linguistics, 2018.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

A Background on Locate-then-edit Model Editing

A.1 Locate-then-edit Paradigm

Locate-then-edit methods modify factual associations in language models in two steps. A *localization* step identifies a subset of internal parameters with a strong causal effect on factual recall. An *editing* step then computes a parameter perturbation that changes the target association while attempting to preserve unrelated behavior (Meng et al., 2022; 2023). Following the common formulation, we treat the MLP output projection $\boldsymbol{W}_0 \in \mathbb{R}^{d_{\mathrm{v}} \times d_{\mathrm{k}}}$ at a selected layer as an editable memory matrix. For an edit request that changes (s, ρ, o) to (s, ρ, o^*), the key $\boldsymbol{k}$ encodes the subject–relation pattern, and the residual $\boldsymbol{r} = \boldsymbol{v}^* - \boldsymbol{W}_0\boldsymbol{k}$ is optimized so that the value $\boldsymbol{v}^*$ promotes the corrected object o^*. The edited matrix is

$$\boldsymbol{W}^* = \boldsymbol{W}_0 + \boldsymbol{\Delta}, \tag{15}$$

where $\boldsymbol{\Delta}$ denotes the parameter update.

A.2 Causal Tracing for Parameter Localization

Causal tracing estimates which internal states are important for recalling a factual association (Meng et al., 2022; 2023). Given a factual prompt, it compares model predictions under three forward passes. In a *clean run*, the original prompt is processed and all hidden activations are recorded. In a *corrupted run*, noise is injected into the subject representation, which typically lowers the probability of the correct object. In a *corrupted-with-restoration run*, the corrupted input is processed again, but the hidden state at a selected layer l and token position t is restored to its clean value. A state whose restoration substantially recovers the correct prediction is deemed causally important.

Let $p_{\mathrm{clean}}(o)$, $p_{\mathrm{corr}}(o)$, and $p^{l,t}_{\mathrm{rest}}(o)$ denote the probabilities of the correct object under the three runs. The total effect and the indirect effect of restoring (l, t) are

$$\mathrm{TE} = p_{\mathrm{clean}}(o) - p_{\mathrm{corr}}(o), \qquad \mathrm{IE}^{l,t} = p^{l,t}_{\mathrm{rest}}(o) - p_{\mathrm{corr}}(o). \tag{16}$$

Prior work finds that MLP modules in middle layers, at the last subject token, carry the largest indirect effects (Meng et al., 2022; 2023), which motivates editing their output projection $\boldsymbol{W}_0$.

A.3 Computing the Parameter Perturbation

The closed forms in Eqs. equation 3 and equation 4 require $\boldsymbol{K}$, $\boldsymbol{R}$, and $\boldsymbol{C}_0$; the following three subsections describe how each is obtained.

A.3.1 Construction of Edit Keys

The edit-key matrix $\boldsymbol{K}$ is obtained from a single forward pass on the edit prompts. For the i-th request, let $p(s_i, \rho_i)$ denote the natural-language prompt formed by its subject and relation. Given a set of N_c context prefixes $\{c_j\}_{j=1}^{N_c}$, the key averages the MLP activation at the last token of s_i across prefixes (Meng et al., 2022; 2023):

$$\boldsymbol{k}_i = \frac{1}{N_c} \sum_{j=1}^{N_c} \phi_l\big(c_j \oplus p(s_i, \rho_i)\big), \qquad \phi_l(x) = \sigma\big(\boldsymbol{W}_{\text{in}}^l(\boldsymbol{a}^l + \boldsymbol{h}^{l-1})\big)\Big|_{\text{last token of } s_i}, \tag{17}$$

where $\oplus$ denotes text concatenation and ϕ_l extracts the MLP key representation (Eq. equation 2) at layer l. The keys are collected into $\boldsymbol{K} = [\boldsymbol{k}_1, \ldots, \boldsymbol{k}_N]$.

A.3.2 Construction of Target Residuals

The residual matrix $\boldsymbol{R}$ contains one desired residual per edit request. In MEMIT-style editing, also adopted by EMMET, each residual is obtained by optimizing an intervention vector on the frozen model (Meng et al., 2023; Gupta et al., 2024). Let $\mathcal{M}$ denote the original model and $\mathcal{M}_{l,t}(\boldsymbol{g}_i)$ the intervened model in which $\boldsymbol{g}_i$ is added to the MLP output at layer l and token position t, the last token of the subject. Let p_i' be a generic prompt of the form "$\{s_i\}$ is a", on which the intervened and the original model are compared to preserve the model's understanding of the subject. The residual is

$$\boldsymbol{r}_i = \arg\min_{\boldsymbol{g}_i} \; \frac{1}{N_c} \sum_{j=1}^{N_c} -\log P_{\mathcal{M}_{l,t}(\boldsymbol{g}_i)}[o_i^* \mid c_j \oplus p(s_i, \rho_i)] + \lambda_{\text{KL}}\, D_{\text{KL}}\big(P_{\mathcal{M}_{l,t}(\boldsymbol{g}_i)}[\cdot \mid p_i'] \,\|\, P_{\mathcal{M}}[\cdot \mid p_i']\big)\,. \tag{18}$$

The first term drives the intervened model to predict the target object under multiple prefixes; the second discourages distributional drift relative to the original model, with weight λ_{KL}. Since $\boldsymbol{g}_i$ is added directly to the MLP output $\boldsymbol{W}_0\boldsymbol{k}_i$, the optimized $\boldsymbol{g}_i$ is the residual $\boldsymbol{r}_i = \boldsymbol{v}_i^* - \boldsymbol{W}_0\boldsymbol{k}_i$ of Section 2.2. For multi-layer editing, the intervention is optimized at the last edited layer and the resulting residual is distributed across the edited layers (Meng et al., 2023). Each $\boldsymbol{r}_i$ is optimized independently of the others, and the results are collected into $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N]$.

A.3.3 Construction of Background Keys and Uncentered Covariance

Following prior locate-then-edit methods, the background key matrix $\boldsymbol{K}_0$ is built from naturally occurring text rather than from the edit requests (Meng et al., 2022; 2023). A large collection of Wikipedia passages is fed through the model, and the MLP key activations at all token positions are recorded as representative samples of the knowledge already stored in the editable module; ROME and MEMIT collect $100{,}000$ such vectors. In practice the background statistic is computed as

$$\boldsymbol{C}_0 = \lambda_c\, \mathbb{E}_{\boldsymbol{k}_0 \sim \boldsymbol{K}_0}\big[\boldsymbol{k}_0 \boldsymbol{k}_0^\top\big]\,, \tag{19}$$

which coincides with $\boldsymbol{K}_0\boldsymbol{K}_0^\top$ up to a scalar; the factor λ_c controls the trade-off between fitting new facts and preserving existing knowledge. Throughout the paper $\boldsymbol{C}_0$ denotes this scaled matrix, so that Eqs. equation 3 and equation 4 and the whitening $\tilde{\boldsymbol{K}} = \boldsymbol{C}_0^{-1/2}\boldsymbol{K}$ already absorb λ_c.

B Construction of HetionetEdit

We construct HetionetEdit from Hetionet (Himmelstein et al., 2017), a biomedical knowledge graph. The benchmark is designed to evaluate model editing under domain-concentrated updates, where multiple facts are organized around recurring biomedical entities.

B.1 Triple Extraction and Aggregation

We parse the original Hetionet JSON file and convert each edge into a factual triple. The source node is treated as the subject, the target node as the object, and the relation is defined by combining the edge type with the source and target node types. This yields a fine-grained relation type that distinguishes relations with the same edge label but different biomedical entity types.

Dataset	#Instances	#Subjects	#Relations per Subject
Sub1	2,506	2,506	1.0
Sub2	6,774	3,387	2.0
Sub3	11,907	3,969	3.0
Sub4	15,380	3,845	4.0
Sub5	14,405	2,881	5.0
Sub6	5,802	967	6.0
All	56,774	17,555	3.2

Table 5: Statistics of HetionetEdit subsets grouped by the number of relation types per subject. An instance corresponds to one subject–relation editing sample.

For triples sharing the same subject and fine-grained relation, we merge their objects into an object list:

$$(s, \rho) \mapsto \mathcal{O}_{s,\rho}, \tag{20}$$

where s is the subject, ρ is the fine-grained relation, and $\mathcal{O}_{s,\rho}$ denotes all objects associated with this subject–relation pair.

B.2 Editing Samples

Each aggregated subject–relation pair is converted into a natural-language editing instance using predefined relation templates. For each instance, we generate (i) one main editing prompt, (ii) three semantically equivalent paraphrases, and (iii) three generation prompts for evaluation, by populating the templates with the subject name. The corresponding answer is derived from $\mathcal{O}_{s,\rho}$. Some subject–relation pairs have long object lists. To keep the main benchmark compact, we store only the first object in the main dataset.

B.3 Neighborhood Prompts

To evaluate locality, we construct two types of neighborhood prompts. The first type uses the same subject but a different relation:

$$(s, \rho) \to (s, \rho'), \qquad \rho'. \tag{21}$$

This tests whether editing one relation affects another relation of the same subject.

The second type uses the same relation but a different subject:

$$(s, \rho) \to (s', \rho), \qquad s' \neq s. \tag{22}$$

This tests whether editing one subject affects another subject of the same relation.

To avoid data leakage between edited facts and locality facts, triples used for constructing neighborhood prompts are removed from the pool of ordinary editing instances.

B.4 Dataset Statistics

We split HetionetEdit by the number of relation types associated with each subject. This yields six subsets, denoted as HetionetEdit Sub1–Sub6. Each subset contains subjects with the same number of relation types, enabling controlled evaluation under different levels of subject-centric complexity. Table 5 reports the statistics of each subset. Table 6 shows an example instance from HetionetEdit.

C Details for Derivation

C.1 Proof of the Spectral Form

Throughout, $\tilde{\boldsymbol{K}} = \boldsymbol{C}_0^{-1/2}\boldsymbol{K} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$ is the compact SVD of the whitened key matrix, with $\boldsymbol{U}_r \in \mathbb{R}^{d_k \times r}$, $\boldsymbol{V}_r \in \mathbb{R}^{N \times r}$, $\boldsymbol{U}_r^\top\boldsymbol{U}_r = \boldsymbol{V}_r^\top\boldsymbol{V}_r = \boldsymbol{I}_r$, and $\boldsymbol{\Sigma}_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ with $\sigma_i > 0$. We use $\boldsymbol{C}_0 \succ 0$ and the two identities

$$\boldsymbol{K}^\top = \tilde{\boldsymbol{K}}^\top\boldsymbol{C}_0^{1/2} = \boldsymbol{V}_r\boldsymbol{\Sigma}_r\boldsymbol{U}_r^\top\boldsymbol{C}_0^{1/2}, \qquad \boldsymbol{K}^\top\boldsymbol{C}_0^{-1} = \tilde{\boldsymbol{K}}^\top\boldsymbol{C}_0^{-1/2} = \boldsymbol{V}_r\boldsymbol{\Sigma}_r\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}. \tag{23}$$

Field	Example
`sample_id`	5
Subject	`heart`
Subject ID	`Anatomy:UBERON:0000948`
Relation Type	`Anatomy - expresses - Gene`
Main Prompt	Which genes have measurable expression within the anatomical structure heart?
Paraphrase Prompts	What expressed genes are found in the anatomical structure heart? Which genes are expressed in the anatomical structure heart? Which genes can be detected in the anatomical structure heart?
Same-subject Different-relation Neighborhood	Prompt: Which genes are linked to higher expression in the anatomical structure heart? Answer: C6
Same-relation Different-subject Neighborhood	Prompt: Which genes have measurable expression within the anatomical structure nose? Answer: OMP
Target Object	F9
Target Object ID	`Gene:2158`

Table 6: An example instance from HetionetEdit. The sample illustrates the subject-level organization, fine-grained relation type, editing prompts, locality prompts, and target object.

Soft constraint. Factor the regularized Gram matrix through $\boldsymbol{C}_0^{1/2}$:

$$\boldsymbol{C}_0 + \boldsymbol{K}\boldsymbol{K}^\top = \boldsymbol{C}_0^{1/2}\big(\boldsymbol{I} + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}^\top\big)\boldsymbol{C}_0^{1/2}, \qquad \big(\boldsymbol{C}_0 + \boldsymbol{K}\boldsymbol{K}^\top\big)^{-1} = \boldsymbol{C}_0^{-1/2}\big(\boldsymbol{I} + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}^\top\big)^{-1}\boldsymbol{C}_0^{-1/2}. \tag{24}$$

Substituting into Eq. equation 3 and using equation 23,

$$\boldsymbol{\Delta}_S = \boldsymbol{R}\,\tilde{\boldsymbol{K}}^\top \boldsymbol{C}_0^{1/2}\,\boldsymbol{C}_0^{-1/2}\big(\boldsymbol{I} + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}^\top\big)^{-1}\boldsymbol{C}_0^{-1/2} = \boldsymbol{R}\,\tilde{\boldsymbol{K}}^\top\big(\boldsymbol{I} + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}^\top\big)^{-1}\boldsymbol{C}_0^{-1/2}. \tag{25}$$

Since $\tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}^\top = \boldsymbol{U}_r\boldsymbol{\Sigma}_r^2\boldsymbol{U}_r^\top$, the push-through identity gives $\big(\boldsymbol{I}+\boldsymbol{U}_r\boldsymbol{\Sigma}_r^2\boldsymbol{U}_r^\top\big)^{-1}\boldsymbol{U}_r = \boldsymbol{U}_r\big(\boldsymbol{I}+\boldsymbol{\Sigma}_r^2\big)^{-1}$, hence

$$\tilde{\boldsymbol{K}}^\top\big(\boldsymbol{I} + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{K}}^\top\big)^{-1} = \boldsymbol{V}_r\boldsymbol{\Sigma}_r\boldsymbol{U}_r^\top\big(\boldsymbol{I} + \boldsymbol{U}_r\boldsymbol{\Sigma}_r^2\boldsymbol{U}_r^\top\big)^{-1} = \boldsymbol{V}_r\boldsymbol{\Sigma}_r\big(\boldsymbol{I} + \boldsymbol{\Sigma}_r^2\big)^{-1}\boldsymbol{U}_r^\top. \tag{26}$$

Therefore $\boldsymbol{\Delta}_S = \boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_S\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}$ with $\boldsymbol{\Lambda}_S = \boldsymbol{\Sigma}_r(\boldsymbol{I} + \boldsymbol{\Sigma}_r^2)^{-1} = \mathrm{diag}\big(\sigma_i/(1+\sigma_i^2)\big)$.

Hard constraint. By equation 23, $\boldsymbol{K}^\top\boldsymbol{C}_0^{-1}\boldsymbol{K} = \tilde{\boldsymbol{K}}^\top\tilde{\boldsymbol{K}} = \boldsymbol{V}_r\boldsymbol{\Sigma}_r^2\boldsymbol{V}_r^\top$, whose inverse on its range (the Moore–Penrose pseudoinverse, equal to the inverse when $r = N$) is $\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-2}\boldsymbol{V}_r^\top$. Substituting into Eq. equation 4,

$$\boldsymbol{\Delta}_H = \boldsymbol{R}\,\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-2}\boldsymbol{V}_r^\top\;\boldsymbol{V}_r\boldsymbol{\Sigma}_r\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2} = \boldsymbol{R}\,\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}, \tag{27}$$

i.e., $\boldsymbol{\Delta}_H = \boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_H\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}$ with $\boldsymbol{\Lambda}_H = \boldsymbol{\Sigma}_r^{-1} = \mathrm{diag}(1/\sigma_i)$.

C.2 Fitting decomposition.

For either filter, $\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-1/2}\boldsymbol{K} = \boldsymbol{U}_r^\top\tilde{\boldsymbol{K}} = \boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$, so $\boldsymbol{\Delta}_A\boldsymbol{K} = \boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_A\boldsymbol{\Sigma}_r\boldsymbol{V}_r^\top$. Writing $\boldsymbol{R} = \boldsymbol{R}\boldsymbol{V}_r\boldsymbol{V}_r^\top + \boldsymbol{R}(\boldsymbol{I} - \boldsymbol{V}_r\boldsymbol{V}_r^\top)$ and subtracting,

$$\boldsymbol{\Delta}_A\boldsymbol{K} - \boldsymbol{R} = -\boldsymbol{R}(\boldsymbol{I} - \boldsymbol{V}_r\boldsymbol{V}_r^\top) - \boldsymbol{R}\boldsymbol{V}_r\big(\boldsymbol{I} - \boldsymbol{\Lambda}_A\boldsymbol{\Sigma}_r\big)\boldsymbol{V}_r^\top. \tag{28}$$

C.3 Proof of the Perturbation Bound

By the spectral form equation 8, the perturbation induced on the non-key states factorizes as

$$\boldsymbol{\Delta}_A\boldsymbol{H}_{\mathrm{nk}} = \big(\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_A\big)\big(\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-\frac{1}{2}}\boldsymbol{H}_{\mathrm{nk}}\big), \qquad A \in \{S, H\}. \tag{29}$$

The spectral norm is sub-multiplicative, $\|\boldsymbol{A}\boldsymbol{B}\|_2 \le \|\boldsymbol{A}\|_2\|\boldsymbol{B}\|_2$ for any conformable $\boldsymbol{A}, \boldsymbol{B}$. Applying this to equation 29 gives

$$\|\boldsymbol{\Delta}_A\boldsymbol{H}_{\mathrm{nk}}\|_2 \le \|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Lambda}_A\|_2\,\Big\|\boldsymbol{U}_r^\top\boldsymbol{C}_0^{-\frac{1}{2}}\boldsymbol{H}_{\mathrm{nk}}\Big\|_2, \tag{30}$$

which is Eq. equation 10. For the hard filter, $\boldsymbol{\Lambda}_H = \boldsymbol{\Sigma}_r^{-1}$ and the amplification bound reads $\|\boldsymbol{R}\boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\|_2$, the quantity constrained in Section 4.

Comparison of the two filters. For the soft filter, the entries of $\mathbf{\Lambda}_S$ satisfy $\sigma_i/(1+\sigma_i^2) \leq \frac{1}{2}$, so $\|\mathbf{\Lambda}_S\|_2 \leq \frac{1}{2}$ and

$$\|\boldsymbol{R}\boldsymbol{V}_r\mathbf{\Lambda}_S\|_2 \leq \|\boldsymbol{R}\boldsymbol{V}_r\|_2\|\mathbf{\Lambda}_S\|_2 \leq \tfrac{1}{2}\|\boldsymbol{R}\boldsymbol{V}_r\|_2. \tag{31}$$

For the hard filter, the i-th column of $\boldsymbol{R}\boldsymbol{V}_r\mathbf{\Lambda}_H$ is $\boldsymbol{R}\boldsymbol{v}_i/\sigma_i$, and the spectral norm of a matrix is at least the Euclidean norm of any of its columns, hence

$$\|\boldsymbol{R}\boldsymbol{V}_r\mathbf{\Lambda}_H\|_2 \geq \max_i \frac{\|\boldsymbol{R}\boldsymbol{v}_i\|_2}{\sigma_i}. \tag{32}$$

□

D Experimental Setup

D.1 Baseline Methods

We compare our method with representative baselines from three categories: (1) fine-tuning, serving as a direct optimization baseline; (2) classical locate-then-edit batch editing methods; and (3) improved batch editing methods well suited to the SCMR scenario. A brief description of each baseline method is provided below.

FT. FT directly updates a selected subset of model parameters using the editing objective while keeping the remaining parameters frozen. It serves as a simple optimization-based baseline that does not rely on an explicit key-value memory formulation.

MEMIT. MEMIT is a representative locate-then-edit method for mass model editing (Meng et al., 2023). It extends ROME-style (Meng et al., 2022) key-value editing to multiple MLP layers and computes closed-form updates to insert many factual associations into the model. Since MEMIT is designed for batched edits, it is one of the primary baselines for evaluating batch editing performance.

EMMET. EMMET formulates ROME and MEMIT under a unified preservation-memorization objective (Gupta et al., 2024). Unlike MEMIT, which uses a least-squares constraint for batched editing, EMMET performs mass editing with an equality-constrained formulation. It is therefore a suitable baseline for comparing against methods that explicitly consider exact realization of edit constraints.

PMET. PMET observes that Transformer-layer hidden states mix information from Attn, MLP, and residual connections (Li et al., 2024). To improve editing precision, it optimizes Transformer-component hidden states while using the optimized MLP component to update MLP weights. We include PMET as a stronger locate-then-edit baseline that refines the value optimization process.

ALPHAEDIT. ALPHAEDIT aims to reduce disruption to preserved knowledge by projecting editing perturbations onto the null space of the preserved knowledge matrix before applying them to model parameters (Fang et al., 2025). It is closely related to preservation-oriented editing and therefore provides a strong comparison for our subspace-preserving update.

MEMIT-MERGE. MEMIT-MERGE targets same-subject batch editing, where multiple edits share the same subject but correspond to different target facts (Dong et al., 2025). It attributes MEMIT's degradation in this setting to key-value conflicts caused by identical subject-derived keys being forced to map to different values, and mitigates the issue by merging the value computation process for same-subject edits.

EAMET. EAMET addresses robustness issues in massive editing by aligning key and residual embedding spaces (Dai et al., 2025). It is motivated by the observation that existing methods degrade under large-scale, context-rich, or same-subject editing settings due to embedding misalignment among knowledge items.

D.2 DATASETS

We evaluate on three datasets spanning two complementary scenarios: subject-diverse factual editing (ZsRE and CounterFact) and subject-centric multi-relation factual editing in the biomedical domain (HetionetEdit). The datasets are described below.

ZsRE. ZsRE is a question-answering benchmark widely used in model editing (Levy et al., 2017). Each edit instance contains a subject, a relation-style question, and a target answer. The dataset also provides paraphrased prompts, making it suitable for evaluating whether an edit generalizes beyond the original wording.

CounterFact. CounterFact is a counterfactual editing benchmark designed to test whether a model can replace an existing factual association with a new target object (Meng et al., 2022).

HetionetEdit. HetionetEdit is our domain-concentrated editing benchmark constructed from the biomedical knowledge graph Hetionet (Himmelstein et al., 2017). Unlike broad-coverage benchmarks, HetionetEdit contains subject-centric multi-relation structures, where the same biomedical subject may appear in multiple edit requests with different relation types. Construction details are provided in Appendix B.

D.3 EVALUATION METRICS

We follow the evaluation protocol used in prior model editing work (Dai et al., 2025). Let f_θ denote the edited model. For the i-th edit instance, let (s_i, ρ_i) denote the editing prompt, o_i^* denote the target answer after editing, and o_i denote the original answer before editing. We use different metric definitions for CounterFact/HetionetEdit and ZsRE, following their standard evaluation settings.

D.3.1 COUNTERFACT AND HETIONETEDIT METRICS

CounterFact and HetionetEdit are evaluated with Efficacy, Generalization, Specificity, and Fluency. HetionetEdit follows the same metric form as CounterFact.

Efficacy. Efficacy measures whether the edited model predicts the target answer on the original editing prompt:

$$\text{Eff.} = \mathbb{E}_i \left[\mathbf{1} \left\{ o_i^* = \arg\max_o P_{f_\theta}(o \mid s_i, \rho_i) \right\} \right]. \tag{33}$$

Generalization. Generalization evaluates whether the edit transfers to semantically equivalent paraphrases. Let $\mathcal{P}_i$ denote the paraphrase prompt set for the i-th edit:

$$\text{Gen.} = \mathbb{E}_{i,\, p \in \mathcal{P}_i} \left[\mathbf{1} \left\{ o_i^* = \arg\max_o P_{f_\theta}(o \mid p) \right\} \right]. \tag{34}$$

Specificity. Specificity measures whether the edit preserves neighboring facts. Let $\mathcal{N}_i$ denote the neighborhood prompt set for the i-th edit. Following the CounterFact-style protocol, we compare the probability of the original answer with that of the edited target:

$$\text{Spec.} = \mathbb{E}_{i,\, q \in \mathcal{N}_i} \left[\mathbf{1} \left\{ P_{f_\theta}(o_i \mid q) > P_{f_\theta}(o_i^* \mid q) \right\} \right]. \tag{35}$$

For HetionetEdit, $\mathcal{N}_i$ includes both same-subject different-relation prompts and same-relation different-subject prompts.

Fluency. Fluency measures the quality of generated continuations and penalizes excessive repetition. Following prior work, we compute an n-gram entropy-based score:

$$\begin{aligned} \text{Flu.} = &- \frac{2}{3} \sum_k g_2(k) \log_2 g_2(k) \\ &+ \frac{4}{3} \sum_k g_3(k) \log_2 g_3(k), \end{aligned} \tag{36}$$

where $g_2(k)$ and $g_3(k)$ denote the normalized frequencies of bigram and trigram k, respectively.

D.3.2 ZsRE Metrics

For ZsRE, we report efficacy, generalization, and locality following the standard question-answering evaluation setting.

Efficacy. Efficacy is the average top-1 accuracy on the original editing prompts:

$$\text{Eff.} = \mathbb{E}_i \left[\mathbf{1} \left\{ o_i^* = \arg\max_o P_{f_\theta}(o \mid s_i, \rho_i) \right\} \right]. \tag{37}$$

Generalization. Generalization is the average top-1 accuracy on generalization prompts. Let $\mathcal{P}_i$ denote the set of paraphrased prompts:

$$\text{Gen.} = \mathbb{E}_{i,\, p \in \mathcal{P}_i} \left[\mathbf{1} \left\{ o_i^* = \arg\max_o P_{f_\theta}(o \mid p) \right\} \right]. \tag{38}$$

Specificity. Specificity evaluates whether the edited model preserves correct behavior on neighboring prompts. Let $\mathcal{N}_i$ denote the neighborhood prompt set and $a_{i,q}$ denote the expected answer for neighborhood prompt q:

$$\text{Spe.} = \mathbb{E}_{i,\, q \in \mathcal{N}_i} \left[\mathbf{1} \left\{ a_{i,q} = \arg\max_o P_{f_\theta}(o \mid q) \right\} \right]. \tag{39}$$

D.4 Implementation Details

All experiments are conducted on a single A100 80GB GPU. For the baselines, we follow the official or commonly used hyperparameter settings when available. In our method, the editing layer is set to the 4th layer ($L = 4$). We employ a cosine decay learning rate schedule with the learning rate decaying to 10% of its base value, and train for 25 steps across all models and datasets. For all models, we use a base learning rate of 0.2 and $\alpha = 5$.

D.5 Details of GLUE

To evaluate whether model editing affects the general capabilities of the edited models, we conduct downstream evaluations on six representative tasks, covering grammatical acceptability, factual and commonsense reasoning, semantic similarity, sentiment analysis, and natural language inference.

CoLA. CoLA (Warstadt et al., 2019) evaluates grammatical acceptability by requiring models to determine whether a given sentence is linguistically acceptable.

MMLU. MMLU (Hendrycks et al., 2021) measures multi-task knowledge and reasoning ability across diverse domains. It is commonly used to evaluate the broad problem-solving capability of language models under zero-shot or few-shot settings.

NLI. NLI (Williams et al., 2018) evaluates natural language inference ability, where models are required to identify the logical relation between a premise and a hypothesis, including entailment, contradiction, and neutral.

MRPC. MRPC (Dolan & Brockett, 2005) evaluates semantic equivalence detection by asking models to determine whether two sentences are paraphrases of each other.

SST. SST (Socher et al., 2013) evaluates sentiment classification on movie review sentences, where models predict the sentiment polarity based on human annotations.

RTE. RTE (Bentivogli et al., 2009) evaluates textual entailment by requiring models to determine whether a premise sentence entails a given hypothesis.