Golden Layers and Where to Find Them: Improved Knowledge Editing for Large Language Models Via Layer Gradient Analysis

Shrestha Datta[1], **Hongfu Liu**[2], **and Anshuman Chhabra**[*1]
[1]University of South Florida, Tampa, FL, USA
[2]Brandeis University, Waltham, MA, USA
{shresthadatta, anshumanc}@usf.edu, hongfuliu@brandeis.edu

Abstract

Knowledge editing in Large Language Models (LLMs) aims to update the model's prediction for a specific query to a desired target while preserving its behavior on all other inputs. This process typically involves two stages: identifying the layer to edit and performing the parameter update. Intuitively, different queries may localize knowledge at different depths of the model, resulting in different sample-wise editing performance for a fixed editing layer. In this work, we hypothesize the existence of fixed *golden* layers that can achieve *near-optimal* editing performance similar to sample-wise optimal layers. To validate this hypothesis, we provide empirical evidence by comparing golden layers against ground-truth sample-wise optimal layers. Furthermore, we show that golden layers can be reliably identified using a proxy dataset and generalize effectively to unseen test set queries across datasets. Finally, we propose a novel method, namely *Layer Gradient Analysis (LGA)*, that estimates golden layers efficiently via gradient-attribution, avoiding extensive trial-and-error across multiple editing runs. Extensive experiments on several benchmark datasets demonstrate the effectiveness and robustness of our LGA approach across different LLM types and various knowledge editing methods.

1 Introduction

Large Language Models (LLMs) encode extensive factual and relational knowledge acquired during pre-training, enabling strong performance across a wide range of downstream tasks. However, this knowledge is implicitly distributed across a large number of model parameters, making it expensive to directly access or update after the pre-training stage. As a result, LLMs may produce outdated, incorrect, or undesirable outputs, and rectifying such errors through full retraining or large-scale fine-tuning is often computationally prohibitive. Moreover, doing so can also potentially degrade previously learned desirable behaviors and useful model knowledge unrelated to the information that needs to be edited [21, 27]. Knowledge editing methods [51] in LLMs thus seek to address this challenge by modifying a model's prediction for a specific test query to a desired target while preserving its behavior on all other inputs. This capability is increasingly important for real-world deployment, where models must be adaptable, reliable, and aligned with evolving real-world information.

Most existing knowledge editing approaches follow a *locate-then-edit* paradigm [48]. Given a test query and a desired new target knowledge, knowledge editing methods first identify where the relevant knowledge is stored in the model and then perform a constrained parameter update at that location. In practice, this typically reduces to selecting a *layer* (or in some cases, block of layers) to edit, followed by an optimization-based update to those parameters. Prior work has shown that

*Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the choice of editing layer is a critical factor in determining both rewrite success and the degree of unintended effect to other unrelated model knowledge [16, 43]. Consequently, several methods attempt to identify important layers using causal signals, such as *Causal Mediation Analysis (CMA)* [31], which has emerged as a widely adopted standard; or via gradients, such as Salient Layers Editing Model (SaLEM) [33]. Despite the perceived effectiveness of these methods, prior work has shown that these strategies often fail to reliably identify the best editing layer [15, 16], motivating the need for improved layer-wise editing performance prediction methods.

Therefore, in this work, we take a closer look at how editing performance varies across layers and uncover a consistent empirical pattern. While the sample-wise optimal editing layer may differ across individual queries, we observe that, at the dataset level, a large majority of samples concentrate their optimal performance on the same layer or a small set of layers. This observation motivates us to define the concept of *golden layers*, which we define as fixed layers that, when used uniformly across samples, achieve aggregate editing performance that is statistically indistinguishable and/or near-optimal to that obtained by editing each sample at its own optimal layer. The existence of golden layers suggests that knowledge relevant for editing is not evenly distributed across all model layers, and that certain layers act as particularly effective intervention points. Importantly, this phenomenon implies that near-optimal editing performance can be achieved without per-sample layer selection, provided that these layers can be identified.

Further, motivated by these observations, in this paper, we propose *Layer Gradient Analysis (LGA)*, a novel and efficient method for estimating golden layers without performing any actual knowledge edits. Inspired by recent work on sample-wise gradient attribution [5, 19, 37], LGA leverages *layer-specific* gradient attribution to quantify how strongly each layer mediates the interaction between the model's existing knowledge and the desired new target knowledge. By aggregating first-order gradient signals across a proxy set of editing queries, LGA produces a robust estimate of which layers are most suitable for knowledge editing. This approach avoids exhaustive layer-wise trial-and-error, scales efficiently to large models, and directly targets the layers that yield near-optimal editing performance in practice. Through extensive experiments across multiple LLM architectures, datasets, and editing methods, we show that editing at layers selected by LGA consistently outperforms standard layer selection approaches such as CMA and SaLEM, while significantly reducing the computational overhead associated with them.

In sum, we highlight our major contributions in this work:

- We empirically demonstrate the existence of *golden layers* for knowledge editing in LLMs, showing that in most cases, fixed editing layers can achieve near-optimal or statistically indistinguishable performance compared to sample-wise optimal layer selection across datasets and models.
- Furthermore, we show that golden layers can be reliably estimated using proxy datasets and generalize effectively to unseen test queries, enabling potential editing strategies that are more computationally efficient than exhaustive layer-wise trial-and-error.
- To this end, we propose a novel first-order, gradient-attribution-based method for identifying golden layers, named Layer Gradient Analysis (LGA). Through extensive experiments across LLMs, editing strategies, and benchmark datasets, we demonstrate that LGA outperforms CMA and SaLEM in terms of both editing performance as well as computational efficiency.

2 Related Work

Locate-Then-Edit Knowledge Editing. There are several popular locate-then-edit methods that have been developed for knowledge editing in LLMs. *ROME* [31] performs editing by applying a rank-one update to the weights of a selected MLP layer, inserting a new key-value association that encodes the desired factual modification while approximately preserving previously stored associations. Building on *ROME*, *R-ROME* [12] addresses instability in sequential editing by enforcing consistent use of averaged key representations in the rank-one update formulation. *MEMIT* [32] extends single-fact editing methods to the mass-editing regime by simultaneously inserting thousands of factual associations via a batched least-squares update to transformer weights. *EMMET* [13] further generalizes this framework by formulating batched editing under equality constraints, providing a closed-form solution that unifies *ROME* and *MEMIT* through a preservation-memorization objective. Adjacent to this line of work, *PMET* [25] adopts an optimization-based approach that jointly optimizes hidden states in both the attention and feedforward modules. It is important to note that across all these methods, the selection of editing layers is typically guided by CMA [31], which we will discuss next.

Limitations of Causal Mediation Analysis. A common strategy in locate-then-edit knowledge editing methods is to select MLP intervention layers using CMA (sometimes also referred to as *causal tracing*), under the assumption that components identified as mediating factual recall will be effective targets for editing. CMA treats the transformer network as a causal graph over hidden activations and quantifies how intermediate components mediate the effect of an input query on the model's prediction [8, 46]. However, recent work provides a detailed empirical critique of this assumption, showing that CMA layer scores are often poorly correlated with editing success across a variety of knowledge editing problem variants, and that this weak correlation persists across multiple problem settings [16]. Follow-up work [15] further corroborates this across additional models and evaluation setups, demonstrating that the choice of *editing layer*, rather than causal tracing scores, remains a stronger predictor of editing success. Complementary work [34] examines the mechanistic underpinnings of editing interventions, showing that while CMA attributes decisive roles to early-to-mid MLP sublayers in factual recall, the impact of an edit on the learned representation manifold (e.g., distortion or shattering of entity subspaces) is not well predicted by these attribution scores. These findings highlight the limitations of CMA for layer selection in LLMs and motivate our work on developing efficient layer-selection strategies that are more robust and performant for editing success.

Gradient-Based Attribution. While gradient-based attribution methods such as saliency maps [1] and integrated gradients [42] have had long-standing success in interpretability research, recent work on data valuation has demonstrated their potential as effective estimators of training sample impact on model predictions [4, 22, 37, 50]. More specifically, first-order methods only utilize sample-gradients (obtainable in one-pass) to undertake this analysis [3, 5, 37], balancing both performance and computational efficiency. Recent work has also sought to utilize gradients to assess layer impact in applications such as pruning, mixture-of-experts allocation [2], influence analysis [47, 52], among others. Moreover, for knowledge editing, SaLEM [33] also utilizes first-order information for layer selection, but as our results will subsequently show, it fails to utilize the full power of the gradient signal. Thus, in our paper, we bridge this gap by proposing the novel and efficient first-order Layer Gradient Analysis (LGA) method as an alternative to CMA-based layer selection for model-agnostic editing.

Memorization. Identifying fixed layers for KE is closely related to memorization. Prior work finds that memorization is not spread evenly across a network: certain layers are impacted by it rather than all layers [40], it is concentrated in a small set of neurons scattered over multiple layers [29], and, in LLMs, it can be localized to specific weights that, when fine-tuned, make the model forget the memorized data [41]. These findings indicate the possibility of existence of multiple layers for a successful edit in KE. Besides, Memorization in LLMs can be localized by comparing the gradients of memorized and non-memorized paragraphs [41]. This motivates our use of both old- and new-knowledge gradients for layer selection.

3 Research Questions

Knowledge editing aims to modify a model M such that for a specific query Q prediction/knowledge ($\hat{K} = M(Q)$) is updated to a desired target K_{new}, while preserving the model's behavior on all other inputs. That is, after applying a knowledge editing process $\mathcal{E}$ on the original model M, we wish to obtain an edited model M' which satisfies $M'(Q) = K_{\text{new}}$ while ensuring $M(Q') = M'(Q')$, for all queries $Q' \neq Q$. It typically consists of two phases: (i) identifying the layer to edit, and (ii) modifying the parameters of the selected layer. After selecting a layer block L, the editing procedure $\mathcal{E}$ produces a knowledge-edited model $M' = \mathcal{E}(M; L, Q, K_{\text{new}})$. Different editing methods formalize $\mathcal{E}$ using various optimization-based or closed-form methods [12, 13, 25, 31, 32], but they share the common objective of enforcing the target behavior specified by (Q, K_{new}) while limiting collateral changes outside the edited layers.

Moreover, as demonstrated in past work [16], the choice of editing layer plays a significant role in final editing performance. Specifically, in our investigations on knowledge localization, we observe an intriguing phenomenon: despite the diversity of the underlying information or knowledge, a majority of samples consistently select the same editing layer. This observation motivates us to explore the following two fundamental questions:

- *Does there exist a fixed "golden" layer, or a small set of fixed "golden" layers, that achieves near-optimal performance for knowledge editing when compared to the sample-wise optimal layer?*
- *If such a golden layer exists, can it be efficiently identified without resorting to extensive trial-and-error across multiple editing runs?*

The identification of such a golden layer is of immense importance to the knowledge editing community and, more broadly, to research on LLMs. Moreover, existing methods for layer selection have been shown to fail at detecting the optimal editing layer [15, 16]. Therefore, they have to depend on sample-wise or layer-wise trial-and-error to select a potential layer for editing, leading to substantial computational overhead and limited scalability. A fixed or near-universal golden layer would eliminate repeated layer searches, enabling more efficient, stable, and reproducible knowledge edits. Beyond efficiency, golden layer identification also offers theoretical value. While prior work [9–11, 31] largely assumes that knowledge in LLMs is stored as key–value representations distributed across layers, recent studies [17, 20, 30, 49] have begun to question this view, suggesting that knowledge may be redundantly or unevenly encoded. By investigating the existence of golden layers, our work provides new empirical evidence on how knowledge is organized across model layers, thereby offering a principled foundation for more effective and interpretable knowledge editing methods.

Motivated by these questions, we structure our paper into two parts. First, we empirically investigate the existence of golden layers through extensive trial-and-error experiments. Second, we propose a novel gradient-attribution-based algorithm to estimate the impact of each layer on knowledge editing, enabling efficient golden-layer identification while avoiding costly repeated computations.

4 The Existence of Golden Layers

In this section, we detail the process of golden layer discovery. We first formalize the definition of a golden layer and provide empirical evidence of its existence by comparing it with the ground-truth, sample-wise optimal editing layer. We then demonstrate that the golden layer for a target dataset can be reliably estimated using a related proxy dataset (or even an independent dataset), showing that while the golden layer is largely invariant to individual samples, it systematically varies across different LLM architectures.

Throughout the paper, we conduct experiments across three different LLMs: GPT-2 XL [38], LLaMA2-7B [45], and Gemma3-12B [44], and evaluate on several commonly used knowledge editing benchmark datasets: *ZSRE* [24, 31], *WikiBio* [14], *WikiCounterfact* [53], *WikiRecent* [6], *Counterfact* [31]. Moreover, we evaluate editing performance using standard metrics: *Rewrite Accuracy* (↑), *Rephrase Accuracy* (↑), *Locality* (↑), *Portability* (↑), and *Fluency* (↑), but note that not all metrics are supported on all datasets. Additionally, for easy overall comparison we also provide an *Overall* metric that is simply a weighted average of all other normalized metrics, where higher values denote better performance.[2]

4.1 Defining Golden Layers

Following prior literature [12, 13, 31] and practical considerations, knowledge editing is typically performed on a single layer, with the goal of incorporating new knowledge while preserving existing and broadly shared knowledge. Given access to ground truth, the sample-wise optimal editing layer can be obtained by exhaustively performing knowledge edits at each layer and selecting the one that yields the best editing performance. At the individual sample level, different samples may exhibit different optimal editing layers. However, at the group level, namely, over a dataset with a sufficient number of samples, we observe that a majority of samples consistently favor the same editing layer. Motivated by this empirical regularity, we define golden layers as follows:

Definition *(Golden Layers).* *Given an LLM and a test set of samples for knowledge editing, golden layers are fixed layers for editing across all those samples that achieve, in aggregate, similar performance from that obtained by editing each sample at its own sample-wise optimal layer.*

Based on the above definition, we formulate our first research question: *do golden layers exist?* To examine this hypothesis, we conduct experiments on several benchmark datasets [24, 48]. Specifically, we exhaustively search for the sample-wise optimal editing layer for each individual query instance, according to *Rewrite Accuracy*, the most important metric in knowledge editing, and in parallel identify a single fixed layer that achieves the best average performance across the entire benchmark. By comparing the performance of this fixed layer with that of the sample-wise optimal layers, we empirically evaluate whether a golden layer can approximate the optimal editing behavior at the dataset level.

[2] Appendices B, K, and B.3 provide details regarding implementation, experimental setup, and the performance metrics of the editing pipeline.

Table 1: The performance of the Sample-Wise Optimal Layer (top performing layer for a sample in the test set) and Golden Layer (optimal fixed layer for all samples derived from the test set), for *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact* datasets with R-ROME on three LLMs: GPT-2 XL, LLaMA2-7B, and Gemma3-12B. The optimal layer selection is based on *Rewrite Accuracy*. The two performances are found to be statistically different using the two-sided Student's t-test.

Model	Method	*ZSRE*	*WikiBio*	*WikiCounterfact*	*WikiRecent*	*Counterfact*
GPT-2	Sample-Wise Optimal Layer	$1.0000_{\pm 0.0000}$	$0.8897_{\pm 0.1239}$	$0.9860_{\pm 0.0721}$	$0.9974_{\pm 0.0425}$	$1.0000_{\pm 0.0000}$
	Golden Layer	$0.9993_{\pm 0.0135}$	$0.8256_{\pm 0.1629}$	$0.9537_{\pm 0.1480}$	$0.9887_{\pm 0.0793}$	$0.9871_{\pm 0.1128}$
	Statistical Difference	No	Yes	Yes	Yes	Yes
LLaMA2	Sample-Wise Optimal Layer	$0.9691_{\pm 0.0747}$	$1.0000_{\pm 0.0006}$	$0.9949_{\pm 0.0345}$	$0.9897_{\pm 0.0677}$	$1.0000_{\pm 0.0000}$
	Golden Layer	$0.9667_{\pm 0.0771}$	$0.9912_{\pm 0.0325}$	$0.9922_{\pm 0.0532}$	$0.9792_{\pm 0.0910}$	$0.9995_{\pm 0.0164}$
	Statistical Difference	No	Yes	No	Yes	No
Gemma3	Sample-Wise Optimal Layer	$0.9997_{\pm 0.0082}$	$0.9296_{\pm 0.0827}$	$0.9443_{\pm 0.1333}$	$0.9991_{\pm 0.0143}$	$1.0000_{\pm 0.0000}$
	Golden Layer	$0.9763_{\pm 0.0908}$	$0.8538_{\pm 0.1208}$	$0.9415_{\pm 0.1625}$	$0.9789_{\pm 0.0950}$	$1.0000_{\pm 0.0000}$
	Statistical Difference	Yes	Yes	No	Yes	No

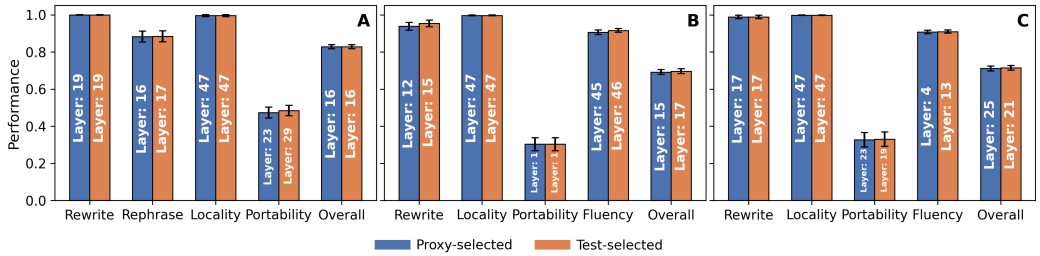

Figure 1: Performance of golden layers selected via the proxy and test sets with GPT-2 XL on (A) *ZSRE*, (B) *WikiCounterfact*, and (C) *Counterfact*. Editing performance evaluation is conducted on the test set queries. Error bars denote the standard error of the mean.

Table 1 compares the knowledge editing performance of the sample-wise optimal layer and the golden layer across all benchmark datasets and LLMs, using *Rewrite Accuracy* as the evaluation metric. For fair comparison, both settings employ the R-ROME editing method [12]. According to the Student's t-test, no statistically significant difference is observed between the golden layer and the sample-wise optimal layer in five cases, including GPT-2 XL on *ZSRE*, LLaMA2-7B on *ZSRE* and *WikiCounterfact*, and Gemma3-12B on *WikiCounterfact* and *Counterfact*. Although the statistical test does not pass in the remaining cases, the absolute performance gaps are generally small with the golden layers mostly achieving a *Rewrite Accuracy* exceeding 0.95, which is typically sufficient for practical deployment. Notable exceptions occur on *WikiBio* with GPT-2 XL and Gemma3-12B, where the golden layer exhibits lower performance with 0.8256 and 0.8538 relative to the sample-wise optimal values. Nevertheless, even in these cases, the performance remains competitive: a strong baseline, CMA [31], achieves *Rewrite Accuracies* of only 0.6894 and 0.6817, respectively. These results suggest that a fixed golden layer can serve as a reasonable approximation to sample-wise layer selection for knowledge editing, while substantially reducing the computational overhead associated with exhaustive per-sample layer searches.

4.2 Golden Layers Across Datasets

Above, we have demonstrated the existence of golden layers on the test set. However, in practical settings, the test set is typically unavailable during training or deployment. To address this limitation, we investigate whether golden layers can be identified using a proxy dataset as a surrogate. Specifically, we aim to estimate the golden layer on an accessible proxy set and examine whether it can effectively generalize to unseen test sets. To achieve this, we first seek a related proxy dataset and exhaustively check the layer performance in terms of knowledge editing, and select the layer with the best performance as the golden layer. In subsequent sections, we will propose an efficient method for golden layer estimation, obviating such expensive trial-and-error analysis.

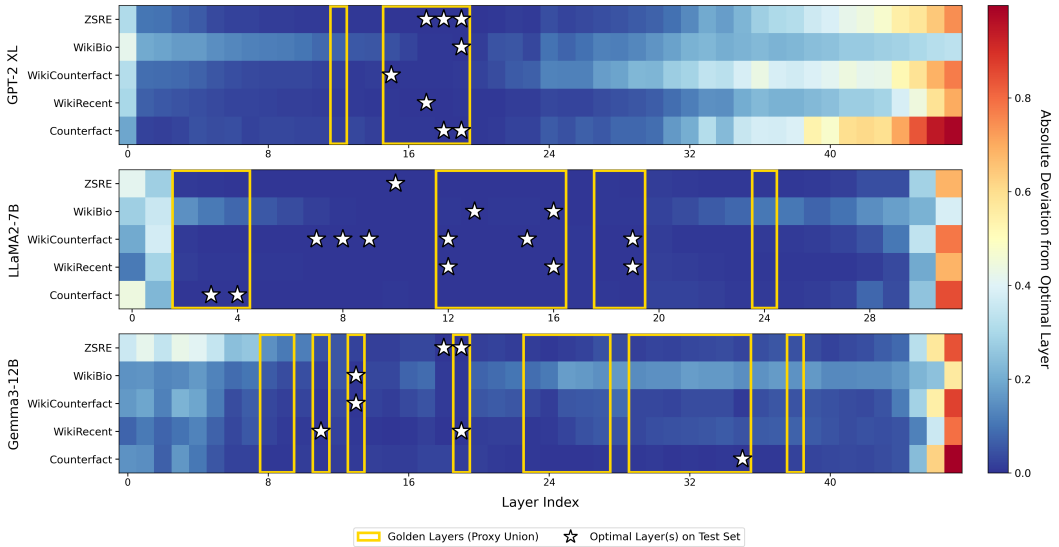

Figure 2: Visualization of model layers for GPT-2 XL, LLaMA2-7B, and Gemma3-12B, where each cell indicates the data-specific performance of a layer measured as the absolute deviation in *Rewrite Accuracy* performance from the optimal layer on the test set. Darker blue cells indicate better performing layers with ☆ denoting optimal layers (these can be tied in performance with multiple optimal layers on the same dataset). The golden cells across layers denote the union of the golden layers independently identified using the proxy set of each dataset. This union of proxy-set golden layers generally select the higher performing layers, and most often, the optimal layers themselves.

Figure 1 presents the knowledge editing performance of GPT-2 XL when using golden layers identified on a proxy set versus a test set across three benchmark datasets: *ZSRE*, *WikiCounterfact*, and *Counterfact*.[3] In these experiments, the proxy and test sets are randomly split from the same dataset (10%: proxy; 90%: test). We observe that the golden layers identified from the proxy set and the test set often coincide. For example, on *ZSRE* and *Counterfact*, the golden layers selected for the *Overall* and *Rewrite* metric, respectively, are identical across the two splits. Even in cases where the selected golden layers differ, the resulting editing performance remains very close. These results indicate that golden layers identified from an accessible proxy set can effectively generalize to unseen test sets. Furthermore, the observations suggest that multiple layers may exhibit comparable near-optimal performance, implying the existence of more than one potential golden layer.

To further examine the robustness of golden layers identified using a proxy set, we construct a proxy union that aggregates five proxy-estimated golden layer subsets, each drawn from the corresponding benchmark datasets. We then evaluate whether they generalize to the corresponding individual test sets. Figure 2 presents a heatmap of *Rewrite Accuracy*, comparing the golden layers identified from the proxy union with the optimal layers on each test set. Darker blue regions indicate smaller performance gaps, whereas red regions correspond to larger discrepancies. Stars denote the optimal layers on the test sets; note that multiple layers may be optimal when they achieve identical performance. The yellow bands highlight the golden layers estimated from the proxy union. As shown in the figure, the golden layers derived from the proxy union largely overlap with the test-set optimal layers, particularly for GPT-2 XL and LLaMA2-7B. This alignment suggests that golden layers estimated from an aggregated proxy set can generalize reasonably well to unseen test sets, supporting their robustness across datasets. Moreover, even though the proxy layers are drawn from different datasets, they can still identify the optimal layer or layers with *Rewrite Accuracy* close [4] to the optimal layer. Interestingly, as Figure 2 demonstrates, while golden layers overlap across optimal test-set layers for the same LLM, they appear in different parts of the network for different LLMs. This implies that golden layers are a *model-dependent* property, while being largely *data-independent*.

[3]Due to space constraints, we provide additional results for *WikiBio* and *WikiRecent* on GPT-2 XL in Appendix C.1 and for all datasets on LLaMA2-7B and Gemma3-12B in Appendix C.2.

[4]To further corroborate the cross-dataset generalizability of the proxy-estimated golden layers, we provide additional results in Appendix H owing to space constraints.

5 Golden Layer Estimation Using Layer Gradient Analysis (LGA)

In the previous sections, we have verified the existence of golden layers, and their potential benefits in serving as the editing layers for knowledge editing. Subsequently, efficient and accurate estimation of golden layers then becomes a key next step towards unlocking their potential in editing pipelines. Towards this goal, we now propose a novel golden layer estimation method that efficiently only operates on first-order information (i.e., layer-specific gradients), and identifies highly performant golden layers. Through the layers selected using our *Layer Gradient Analysis (LGA)* method, we consistently attain higher editing performance compared to standard methods for knowledge editing, such as *Causal Mediation Analysis (CMA)* [31] and *Salient Layers Editing Model (SaLEM)* [33].

5.1 The Proposed Layer Gradient Analysis Approach

Our proposed approach for accurately estimating golden is based on analyzing layer-specific gradients, motivated by recent work on gradient-based data attribution [5, 37, 50], which essentially utilize sample gradients to assess how a training sample z impacts the performance of a model (parameterized by weights $\hat{\theta}$ trained using loss l) on a validation/test sample v, simply using their inner product [37, 50], defined as: $\phi(z, v)=\nabla_{\hat{\theta}}\ell(\hat{\theta}; z)^\top \cdot \nabla_{\hat{\theta}}\ell(\hat{\theta}; v)$. Higher gradient similarity values denote more training sample *impact* on validation/test sample performance. Also note that this process is computationally efficient as gradients can be obtained in one-pass from the model post-training.

In this paper, we extend the gradient attribution inner product from the sample-level to the layer-specific impact. The key idea lies in realizing that the inner product can be restricted to specific layer weights, thereby enabling an assessment of how that specific layer block impacts model performance. That is, $\phi_L(z, v)=\nabla_{\theta_L}\ell(\hat{\theta}; z)^\top \cdot \nabla_{\theta_L}\ell(\hat{\theta}; v)$, where $\nabla_{\theta_L}\ell(\hat{\theta}; z)$ denotes the sample z's gradient of the L-th layer. Moreover, by aggregating these layer-specific attribution scores for a set of samples, we can obtain robust estimates for the layer's impact on the downstream task as $\sum_{(z,v)\in Z}\phi_L(z, v)$. These scores can then be used to compare between layers in terms of how beneficial they are for a task. This framework forms the basis of our Layer Gradient Analysis (LGA) approach.

Clearly, the loss function and the choice of samples will play a significant role in defining the task for which we wish to undertake the layer-attribution analysis. We now propose the procedure for golden layer estimation for knowledge editing in LLMs. For instance, consider, the LLM parameterized by network weights $\hat{\theta}$ trained using autoregressive cross entropy loss ℓ. Moreover, consider a set of editing queries in the proxy set $\mathcal{Q} = \{Q_i\}_{i=1}^n$ and the old model knowledge associated with query Q_i as K_i and the new target knowledge as K_i'. Intuitively, we posit that utilizing the gradients related to both the new and old knowledge for a query can help identify the impact of the layer in the knowledge editing task. To this end, we can estimate the impact of layers across all proxy set queries and predict the golden layer G^* through LGA:

$$\begin{aligned} G^* &= \arg\max_{L\in\mathcal{L}} \sum_{Q_i\in\mathcal{Q}} \phi_L(Q_i \cup K_i, Q_i \cup K_i'), \\ &= \arg\max_{L\in\mathcal{L}} \sum_{Q_i\in\mathcal{Q}} \nabla_{\theta_L}\ell(\hat{\theta}; Q_i \cup K_i)^\top \cdot \nabla_{\theta_L}\ell(\hat{\theta}; Q_i \cup K_i'). \end{aligned}$$

We provide the intuition behind each term of the LGA criterion and its formal justification in Appendix A. Now that we have obtained G^* using our LGA method, we can utilize a knowledge editing method $\mathcal{E}$ to undertake the editing process on a test query (Q_t, K_t') as $\mathcal{E}(M; G^*, Q_t, K_t')$. In the subsequent sections we present our findings for estimating golden layers and how that leads to superior editing performance compared to traditional layer identification methods, such as CMA.

5.2 Golden Layer Identification and Knowledge Editing Performance Comparison

We now compare editing performance when editing layer selection is conducted using LGA, CMA, and SaLEM. Note that our experimental setup is the same as in the previous section.

Comparing LGA, CMA, and SaLEM Across Different Editing Methods. We first evaluate the consistency of our layer selection strategy LGA relative to CMA across different editing methods. We edit the GPT-2 XL model on the *ZSRE*, *WikiBio*, *WikiCounterFact*, and *Counterfact* datasets (results for *WikiRecent* provided in Appendix C.3 due to space limitations). We employ three editing

Table 2: Editing performance evaluation of LGA, SaLEM, and CMA across different knowledge editing methods and the *ZSRE*, *WikiBio*, *WikiCounterFact*, *Counterfact* datasets on GPT-2 XL with R-ROME editing method. Different datasets support different performance metrics, where *Rewrite Accuracy* → RwA, *Rephrase Accuracy* → RpA, *Locality* → LOC, *Portability* → PRT, *Fluency* → FLC, and *Overall* → OV. Layers are identified using the proxy set for both methods and evaluation is undertaken on an unseen test set, demonstrating performance improvements attained by LGA over baselines.

ZSRE

Edit	Selection	RwA	RpA	LOC	PRT	OV
R-ROME	CMA	0.9665	0.7467	0.9608	0.4837	0.6942
	SaLEM	0.6910	0.5951	0.6405	0.3629	0.5724
	LGA (Ours)	0.9851	0.8151	0.9543	0.4757	**0.7106**
EMMET	CMA	0.9850	0.8796	0.6589	0.4202	0.7359
	SaLEM	0.8072	0.5816	0.7042	0.4441	0.6343
	LGA (Ours)	0.9862	0.8833	0.6857	0.3947	**0.7375**
ROME	CMA	0.9669	0.7464	0.9604	0.4832	0.7892
	SaLEM	0.6776	0.5765	0.6302	0.3633	0.5619
	LGA (Ours)	0.9862	0.8178	0.9559	0.4778	**0.8094**

WikiBio

Edit	Selection	RwA	LOC	FLC	OV
R-ROME	CMA	0.6894	0.5692	0.8927	0.7171
	SaLEM	0.4061	0.2563	0.8186	0.4937
	LGA (Ours)	0.7484	0.5587	0.8906	**0.7326**
EMMET	CMA	0.8258	0.3894	0.8612	**0.6921**
	SaLEM	0.5784	0.2811	0.8641	0.5745
	LGA (Ours)	0.8433	0.2793	0.8894	0.6707
ROME	CMA	0.6911	0.2902	0.8788	0.6200
	SaLEM	0.4123	0.1932	0.8214	0.4756
	LGA (Ours)	0.7487	0.2851	0.8840	**0.6393**

WikiCounterFact

Edit	Selection	RwA	LOC	PRT	FLC	OV
R-ROME	CMA	0.9135	0.5601	0.2651	0.9080	0.6741
	SaLEM	0.6370	0.3198	0.2365	0.7629	0.4965
	LGA (Ours)	0.9330	0.5669	0.2663	0.9072	**0.6808**
EMMET	CMA	0.9713	0.3931	0.2695	0.8039	0.6201
	SaLEM	0.6962	0.4342	0.1813	0.8603	0.5538
	LGA (Ours)	0.9841	0.4075	0.2831	0.8343	**0.6382**
ROME	CMA	0.9150	0.5582	0.2635	0.9142	0.6749
	SaLEM	0.6299	0.3036	0.2281	0.7635	0.4891
	LGA (Ours)	0.9379	0.5695	0.2659	0.9118	**0.6839**

Counterfact

Edit	Selection	RwA	RpA	LOC	PRT	OV
R-ROME	CMA	0.9420	0.6284	0.9613	0.4387	0.7426
	SaLEM	0.7981	0.5983	0.7111	0.3585	0.6165
	LGA (Ours)	0.9592	0.7154	0.9560	0.4362	**0.7667**
EMMET	CMA	0.9957	0.3330	0.5693	0.3905	0.5721
	SaLEM	0.9066	0.1359	0.6639	0.4153	0.5304
	LGA (Ours)	0.9925	0.3550	0.6565	0.4081	**0.6030**
ROME	CMA	0.9914	0.3545	0.8579	0.4409	0.6612
	SaLEM	0.7637	0.3131	0.5133	0.3568	0.4867
	LGA (Ours)	0.9968	0.3974	0.8667	0.4367	**0.6744**

methods: R-ROME [12], ROME [31], and EMMET [13] at the respective selected layers. The results are provided in Table 2. As can be observed from *Overall* scores, layers identified by LGA yield higher performance than those selected by CMA and SaLEM across all the configurations, with the exception of EMMET being used with *WikiBio*. Our findings hence suggest that LGA provides a more stable layer selection strategy than CMA and SaLEM across editing methods and datasets.

Across the editing methods, LGA consistently improves *Rewrite Accuracy*, achieving a maximum improvement of 8.56% and 84.29% on *WikiBio* compared to CMA and SaLEM, respectively. In addition, *Rephrase Accuracy*, which is only applicable to *ZSRE* and *Counterfact*, also exhibits consistent gains under LGA regardless of the editing method, with the greatest improvement being 9.16% and 161.22%, respectively. In terms of aggregate performance trends across datasets, LGA exhibits improved *Overall* score increases when applied with different editing methods, with average gains of 1.94% and 29.40% over CMA and SaLEM, respectively. Similar trends are observed across other metrics, such as *Locality*, across datasets. In sum, LGA serves as a better layer selection mechanism compared to CMA and SaLEM, and attains improved performance across metrics and datasets.

LGA vs CMA vs SaLEM Performance Analysis Across LLMs. We now undertake comparative performance evaluation across different and diverse LLM architectures. We employ the R-ROME editing method owing to its superlative editing performance compared to other methods. We apply LGA, SaLEM, and CMA to GPT-2 XL, LLaMA2-7B, and Gemma3-12B on the *ZSRE*, *WikiBio*, *WikiCounterFact*, and *Counterfact* datasets using the R-ROME editing method (results for *WikiRecent* provided in Appendix C.3 due to space constraints). We provide the results in Figure 3. As the figure demonstrates, in terms of *Overall* performance, LGA consistently outperforms CMA across all datasets for GPT-2 XL. For LLaMA2-7B, LGA achieves higher *Overall* performance on *ZSRE*, *WikiBio*, and *Counterfact*, while attaining competitive performance on *WikiCounterFact*. Furthermore, the gains are even more pronounced for Gemma3-12B, with significant *Overall* performance improvement attained for *ZSRE*, *WikiBio*, and *WikiCounterfact*. Similar trends hold for the comparison with SaLEM, as LGA attains improved performance on average.

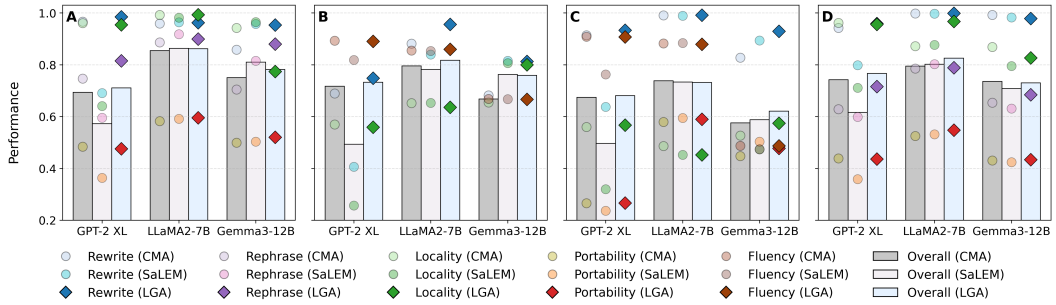

Figure 3: Performance comparison between LGA, SaLEM, and CMA across different LLMs and the (A) *ZSRE*, (B) *WikiBio*, (C) *WikiCounterfact*, and (D) *Counterfact* datasets. Overall, LGA outperforms both CMA and SaLEM, attaining improved knowledge editing performance.

With respect to *Rewrite Accuracy*, LGA selected layers consistently yield higher performance compared to CMA, with improvements reaching up to 19.2% on the *WikiBio* dataset when editing the Gemma3-12B model. A similar pattern is observed for *Rephrase Accuracy*, where the maximum improvement reaches 24.9%. For *Portability*, LGA based layer selection for editing Gemma3-12B exhibits consistently higher performance across all datasets. On the other hand, LGA improves over SaLEM by $+17.53\%$ in *Rewrite*, $+11.63\%$ in *Rephrase*, $+24.36\%$ in *Locality*, $+7.63\%$ in *Portability*, and $+5.11\%$ in *Fluency* averaged across all models and datasets. Overall, these results indicate that LGA maintains improved performance across metrics, with observable improvements over CMA.

Table 3: Editing performance comparison between LGA and random layer selection on *ZSRE* with R-ROME measured by metrics: *Rewrite Accuracy* $\rightarrow$ RwA, *Rephrase Accuracy* $\rightarrow$ RpA, *Locality* $\rightarrow$ LOC, *Portability* $\rightarrow$ PRT, and *Overall* $\rightarrow$ OV. LGA achieves an average *Overall* performance gain of 8.69% across the three models.

Model	Selection	RwA	RpA	LOC	PRT	OV
GPT-2 XL	Random	0.8539	0.5916	0.9701	0.4784	0.6347
	LGA (Ours)	0.9851	0.8151	0.9543	0.4757	**0.7106**
LLaMA2-7B	Random	0.9576	0.8115	0.9885	0.5650	0.8307
	LGA (Ours)	0.9625	0.8988	0.9927	0.5950	**0.8623**
Gemma3-12B	Random	0.8197	0.7692	0.7988	0.4465	0.7086
	LGA (Ours)	0.9535	0.8796	0.7736	0.5198	**0.7816**

Comparison with Random Layer Selection. Next, to verify that LGA's layer selection criterion is informative rather than arbitrary, we compare LGA-selected layers with randomly selected layers across three models on the *ZSRE* dataset as presented in Table 3. The random baseline uniformly samples three layers, and we report its mean performance. LGA consistently outperforms random selection across all three models, achieving an average *Overall* performance gain of 8.69%.

Computational Efficiency Analysis. We conduct experiments to assess the computational efficiency benefits of LGA. In Figure 4, we compare the runtime of LGA, CMA, and SaLEM over *Brute-Force (BF)* golden layer search for the R-ROME knowledge editing method on GPT-2 XL. While the runtime of all methods generally increases with longer input sequences and larger proxy sizes, LGA and SaLEM achieve substantial speedups over both CMA and BF across all datasets, demonstrating exceptional computational efficiency. In contrast, CMA remains relatively inefficient and, in some cases, exhibits runtime comparable to BF, particularly on *WikiBio*, which possesses high average token length. Quantitatively, LGA attains consistent speedups ranging from

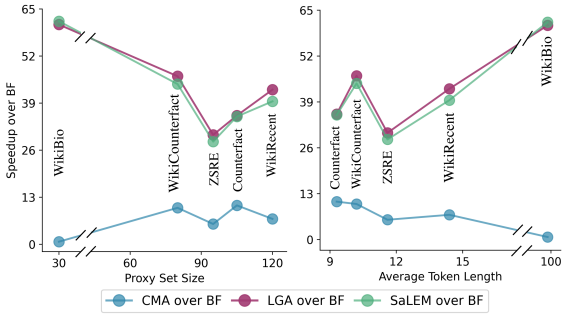

Figure 4: Runtime analysis of LGA, CMA, and SaLEM over Brute-Force (BF) golden layer search for editing via R-ROME on GPT-2 XL. Each of the five datasets: *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact*, are categorized in terms of the average query token length (*left*) and the proxy size (*right*). LGA is extremely computationally efficient and yet attains improved editing performance compared to baselines.

approximately $30\times$ to over $60\times$ relative to BF, while CMA has a lower bound of $1\times$. While LGA and SaLEM attain similar runtime performance, LGA leads to consistent gains in editing performance (as our results from previous sections show). These results indicate that LGA scales efficiently with dataset characteristics and proxy set size while attaining improved editing performance in comparison to CMA and LGA.

Additional Editing Settings and Miscellaneous Results. We also conduct additional experiments with other knowledge editing settings and miscellaneous ablations to demonstrate the efficacy of LGA over baselines. First, in Appendix D, we explore the *sequential editing* task [53] and find the layer selected by LGA maintains stronger performance under multiple consecutive edits compared to CMA and SaLEM. Furthermore, in Appendix E we study *long-form* [39], *compositional* [28], and *unstructured* [7] *editing*, where LGA again achieves consistently improved performance. We next provide ablation results in Appendix F demonstrating that CMA serves as a stronger baseline than random layer selection, and LGA outperforms both. Finally, since SaLEM and LGA both utilize gradient information for layer selection, we also undertake deeper analysis in Appendix G that shows how LGA utilizes cross-knowledge interactions for improved layer selection.

6 Conclusion

In this paper, we studied the knowledge editing task in LLMs, which seeks to update the model's output for a given query to new target knowledge, without impacting the other desirable knowledge learned by the model during pre-training. Generally, knowledge editing methods first identify a specific layer to edit using standard approaches such as CMA and SaLEM, and then perform a minimal parameter update at that layer. Motivated by prior work demonstrating the inefficacy of CMA at selecting the top-performing editing layers, we analyzed editing performance across layers and found evidence for the existence of *golden layers* that achieve on average, near-optimal or statistically similar editing performance compared to sample-wise optimal layers. We then proposed *Layer Gradient Analysis (LGA)*, a novel and computationally efficient gradient-based strategy for robustly estimating these golden layers using a given proxy set of queries. Through several experiments across various benchmark datasets, LLMs, and editing methods, we demonstrated the significant gains achieved by LGA over current layer-selection method baselines, in terms of both editing performance and computational efficiency.

Acknowledgments

We gratefully acknowledge the support of the Google TPU Builders Program, which provided support and access to computational resources, and Google Tunix library for post-training functionality and flexibility to enable this work.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems*, 2018.

[2] Hadi Askari, Shivanshu Gupta, Fei Wang, Anshuman Chhabra, and Muhao Chen. LayerIF: Estimating Layer Quality for Large Language Models using Influence Functions. In *Advances in Neural Information Processing Systems*, 2025.

[3] Irina Bejan, Artem Sokolov, and Katja Filippova. Make Every Example Count: On the Stability and Utility of Self-Influence for Learning from Noisy NLP Datasets. In *Empirical Methods in Natural Language Processing*, 2023.

[4] Anshuman Chhabra, Peizhao Li, Prasant Mohapatra, and Hongfu Liu. What Data Benefits My Classifier? Enhancing Model Performance and Interpretability through Influence-Based Data Selection. In *International Conference on Learning Representations*, 2024.

[5] Anshuman Chhabra, Bo Li, Jian Chen, Prasant Mohapatra, and Hongfu Liu. Outlier Gradient Analysis: Efficiently Identifying Detrimental Training Samples for Deep Learning Models. In *International Conference on Machine Learning*, 2025.

[6] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the Ripple Effects of Knowledge Editing in Language Models. *Transactions of the Association for Computational Linguistics*, 2024.

[7] Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. Everything is Editable: Extend Knowledge Editing to Unstructured Data in Large Language Models. In *International Conference on Learning Representations*, 2025.

[8] Hector Geffner, Rina Dechter, and Joseph Y Halpern. *Probabilistic and causal inference: the works of Judea Pearl*. 2022.

[9] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer Feed-Forward Layers are Key-Value Memories. In *Empirical Methods in Natural Language Processing*, 2021.

[10] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer Feed-Forward Layers build Predictions by Promoting Concepts in the Vocabulary Space. In *Empirical Methods in Natural Language Processing*, 2022.

[11] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting Recall of Factual Associations in Auto-Regressive Language Models. In *Empirical Methods in Natural Language Processing*, 2023.

[12] Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. Rebuilding ROME: Resolving Model Collapse during Sequential Model Editing. In *Empirical Methods in Natural Language Processing*, 2024.

[13] Akshat Gupta, Dev Sajnani, and Gopala Anumanchipalli. A Unified Framework for Model Editing. In *Findings of the Empirical Methods in Natural Language Processing*, 2024.

[14] Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with Grace: Lifelong Model Editing with Discrete K-Value Adaptors. In *Advances in Neural Information Processing Systems*, 2023.

[15] Peter Hase. *Interpretable and Controllable Language Models*. PhD thesis, The University of North Carolina at Chapel Hill, 2024.

[16] Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does Localization Inform Editing? Surprising Differences in Causality-based Localization vs. Knowledge Editing in Language Models. In *Advances in Neural Information Processing Systems*, 2023.

[17] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What Matters in Transformers? Not All Attention is Needed. *arXiv preprint arXiv:2406.15786*, 2024.

[18] David C Hoaglin, Boris Iglewicz, and John W Tukey. Performance of Some Resistant Rules for Outlier Labeling. *Journal of the American Statistical Association*, 1986.

[19] Cathy Jiao, Weizhen Gao, Aditi Raghunathan, and Chenyan Xiong. On the Feasibility of In-Context Probing for Data Attribution. In *Findings of the North American Chapter of the Association for Computational Linguistics*, 2025.

[20] Tianjie Ju, Weiwei Sun, Wei Du, Xinwei Yuan, Zhaochun Ren, and Gongshen Liu. How Large Language Models Encode Context Knowledge? A Layer-Wise Probing Study. In *Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, 2024.

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 2017.

[22] Pang Wei Koh and Percy Liang. Understanding Black-Box Predictions via Influence Functions. In *International Conference on Machine Learning*, 2017.

[23] Yoonho Lee, Annie S. Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical Fine-Tuning Improves Adaptation to Distribution Shifts. In *International Conference on Learning Representations*, 2023.

[24] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-Shot Relation Extraction via Reading Comprehension. In *Conference on Computational Natural Language Learning*, 2017.

[25] Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. PMET: Precise Model Editing in a Transformer. In *AAAI Conference on Artificial Intelligence*, 2024.

[26] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004.

[27] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-Tuning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2025.

[28] Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. Neighboring Perturbations of Knowledge Editing on Large Language Models. In *International Conference on Machine Learning*, 2024.

[29] Pratyush Maini, Michael C. Mozer, Hanie Sedghi, Zachary C. Lipton, J. Zico Kolter, and Chiyuan Zhang. Can Neural Network Memorization Be Localized? In *International Conference on Machine Learning*, 2023.

[30] Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in Large Language Models are More Redundant than You Expect. In *Findings of the Association for Computational Linguistics*, 2025.

[31] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems*, 2022.

[32] Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-Editing Memory in a Transformer. In *International Conference on Learning Representations*, 2023.

[33] Kshitij Mishra, Tamer Soliman, Anil Ramakrishna, Aram Galstyan, and Anoop Kumar. Correcting Language Model Outputs by Editing Salient Layers. In *Findings of the Association for Computational Linguistics*, 2024.

[34] Kento Nishi, Rahul Ramesh, Maya Okawa, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. Representation Shattering in Transformers: A Synthetic Study with Knowledge Editing. In *International Conference on Machine Learning*, 2025.

[35] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Association for Computational Linguistics*, 2002.

[36] Hongming Piao, Hao Wang, Dapeng Wu, and Ying Wei. Aˆ3E: Towards Compositional Model Editing. In *Advances in Neural Information Processing Systems*, 2026.

[37] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating Training Data Influence by Tracing Gradient Descent. In *Advances in Neural Information Processing Systems*, 2020.

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 2019.

[39] Domenic Rosati, Robie Gonzales, Jinkun Chen, Xuemin Yu, Yahya Kayani, Frank Rudzicz, and Hassan Sajjad. Long-Form Evaluation of Model Editing. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

[40] Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. On the Geometry of Generalization and Memorization in Deep Neural Networks. In *International Conference on Learning Representations*, 2021.

[41] Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. Localizing Paragraph Memorization in Language Models. *arXiv preprint arXiv:2403.19851*, 2024.

[42] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *International Conference on Machine Learning*, 2017.

[43] Ryosuke Takahashi, Go Kamoda, Benjamin Heinzerling, Keisuke Sakaguchi, and Kentaro Inui. Understanding the Side Effects of Rank-One Knowledge Editing. In *BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, 2025.

[44] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025.

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.

[46] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In *Advances in Neural Information Processing Systems*, 2020.

[47] Dmytro Vitel and Anshuman Chhabra. First is Not Really Better Than Last: Evaluating Layer Choice and Aggregation Strategies in Language Model Data Influence Estimation. In *International Conference on Learning Representations*, 2026.

[48] Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, et al. EasyEdit: An Easy-to-Use Knowledge Editing Framework for Large Language Models. In *Association for Computational Linguistics*, 2024.

[49] Yifan Wei, Xiaoyan Yu, Yixuan Weng, Huanhuan Ma, Yuanzhe Zhang, Jun Zhao, and Kang Liu. Does Knowledge Localization Hold True? Surprising Differences between Entity and Relation Perspectives in Language Models. In *Conference on Information and Knowledge Management*, 2024.

[50] Ziao Yang, Han Yue, Jian Chen, and Hongfu Liu. Revisit, Extend, and Enhance Hessian-Free Influence Functions. *Transactions on Machine Learning Research*, 2026.

[51] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing Large Language Models: Problems, Methods, and Opportunities. In *Empirical Methods in Natural Language Processing*, 2023.

[52] Chih-Kuan Yeh, Ankur Taly, Mukund Sundararajan, Frederick Liu, and Pradeep Ravikumar. First is Better than Last for Language Data Influence. In *Advances in Neural Information Processing Systems*, 2022.

[53] Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. A Comprehensive Study of Knowledge Editing for Large Language Models. *arXiv preprint arXiv:2401.01286*, 2024.

Appendix

A Intuition and Justification of the LGA Criterion

A.1 Intuition of LGA Definition

We put forward the intuition behind why maximizing the proposed impact of both new and old knowledge in LGA helps in identifying the golden layer:

- A related study on memorization [41] states that memorized examples, i.e., old knowledge, can be unlearned by fine-tuning only the high-gradient weights of these samples. Hence, LGA selects the layers most impacted by the old knowledge to achieve an unlearning effect on the old knowledge.
- Surgical Fine-Tuning [23] selects layers by their target-task (analogous to new knowledge) gradient magnitude on the target task, on the hypothesis that "layers with large gradient magnitudes may carry more information about the target task than others and can therefore be more useful". Thus, LGA selects the layers most impacted by the new knowledge to maximize the learning of the new task.
- Finally, by definition, LGA aggregates the impact over the proxy samples (Section 5.1), and other studies [31, 32] observe that strong causal mediation is limited, averaged across samples, to a fixed set of layers.

A.2 Formal Justification of the LGA Criterion

We provide a derivation showing that maximizing the LGA criterion minimizes a guaranteed upper bound on the required parameter-update magnitude. Essentially, we derive the minimum-norm weight update needed to achieve a post-edit first-order reduction in the new-knowledge loss. We then bound the new-knowledge gradient using the LGA criterion via the Cauchy-Schwarz inequality. Combining these results yields an upper bound on the required parameter update that is inversely proportional to the LGA score.

Definitions: Let θ be the pre-edit parameters, θ_L the layer-L parameters, and $\Delta\theta_L$ the edit update. For query Q, original knowledge K, and new target K', we define:

$$g_L = \nabla_{\theta_L}\ell(\theta; Q \cup K), \qquad g'_L = \nabla_{\theta_L}\ell(\theta; Q \cup K'), \qquad \phi_L = g_L^\top g'_L,$$

where both gradients are evaluated at θ, and assume $g'_L \neq 0$.

Edit constraint: Using the first-order Taylor approximation, we have:

$$\ell(\theta + \Delta\theta_L;\, Q \cup K') \approx \ell(\theta;\, Q \cup K') + (g'_L)^\top \Delta\theta_L.$$

Requiring the new-knowledge loss to decrease by $\delta > 0$ gives us:

$$(g'_L)^\top \Delta\theta_L = -\delta. \tag{C}$$

Minimum-norm update: Also consider the optimization:

$$\min_{\Delta\theta_L} \|\Delta\theta_L\| \quad \text{subject to} \quad (g'_L)^\top \Delta\theta_L = -\delta.$$

By Cauchy–Schwarz inequality, we have:

$$\delta = |(g'_L)^\top \Delta\theta_L| \leq \|g'_L\|\|\Delta\theta_L\| \quad \Longrightarrow \quad \|\Delta\theta_L\| \geq \frac{\delta}{\|g'_L\|}.$$

The equality holds only when the two are collinear. The smallest feasible update is therefore a scalar multiple of the new-knowledge gradient, $\Delta\theta_L = c\, g'_L$ for some scalar c. Substituting this into equation C determines c:

$$\Delta\theta_L = -\frac{\delta}{\|g'_L\|^2} g'_L, \qquad \|\Delta\theta_L\| = \frac{\delta}{\|g'_L\|}. \tag{1}$$

New-gradient norm bound: Applying Cauchy–Schwarz now to the LGA criterion, we have:

$$\phi_L = g_L^\top g'_L \leq \|g_L\|\|g'_L\|.$$

Thus, we can state:

$$\|g'_L\| \geq \frac{\phi_L}{\|g_L\|}. \tag{2}$$

Uniform layer bound: Let $G = \max_L \|g_L\|$. Since $\|g_L\| \leq G$:

$$\|g'_L\| \geq \frac{\phi_L}{G}. \tag{3}$$

Target relation: Substituting equation 3 into equation 1 gives us:

$$\|\Delta\theta_L\| \leq \frac{\delta G}{\phi_L}. \tag{4}$$

Maximizing the LGA Criterion Minimizes the Weight-Update Upper Bound. LGA favors layers where the new knowledge can be incorporated with a smaller guaranteed weight update, thereby better preserving unrelated knowledge. From equation 4, maximizing ϕ_L minimizes a guaranteed upper bound on the update required to install the new target.

Relationship Between the LGA Criterion and the Knowledge Gradient Magnitude. We can show that maximizing the LGA criterion requires a contribution from both gradients, rather than either one alone. From equation 2, and symmetrically:

$$\|g'_L\| \geq \frac{\phi_L}{\|g_L\|}, \qquad \|g_L\| \geq \frac{\phi_L}{\|g'_L\|}.$$

Maximizing ϕ_L therefore raises the guaranteed lower bound on each gradient magnitude relative to the other. That is, neither gradient can be small relative to its counterpart when the criterion is large. A positive ϕ_L additionally requires the two gradients to be aligned.

This joint dependence motivates using both gradients in LGA. This is also reflected empirically, where maximizing only one of the two gradients is not sufficient for effective editing. LGA(K'), which uses only the new-knowledge gradient, comparatively underperforms against LGA (Table 16).

B Implementation Details

Here we introduce the implementation details for knowledge editing in terms of datasets, models, editing methods, and editing performance metrics. Note that similar to CMA and other prior work in knowledge editing, we focus only on the MLP layer modules [9, 31] for layer selection/editing for both LGA and CMA. Furthermore, for our LGA method, in accordance with recent work demonstrating that outlier gradients comprise instances that are detrimental to training loss [3, 5] we exclude layers with overall outlying gradient scores (measured using Tukey's fences and the interquartile range [18]. Unless otherwise specified, we fix the Tukey constant to 1 across all evaluation tasks). Finally, we build LGA upon the EasyEdit knowledge editing framework [48] to ensure consistency in performance and standardized evaluation across all LLMs, datasets, and editing methods.

B.1 Datasets and LLMs

We conduct experiments on five datasets that are popularly used in prior editing work: *ZSRE* [24, 48], *WikiBio* [14, 53], *WikiCounterfact* [53], *WikiRecent* [6, 53], and *Counterfact* [31]. *ZSRE*, *Counterfact*, and *WikiCounterfact* are primarily used to test the introduction of unfactual knowledge, whereas *WikiBio* and *WikiCounterfact* focus on correcting existing knowledge. For each dataset, we use approximately 10% of samples as the proxy set and the remainder 90% as the test set; for example, resulting in approximately 100/1000, 30/270, 100/1000, and 120/1080 samples for the proxy/test sets of *ZSRE*, *WikiBio*, *WikiCounterfact*, and *WikiRecent*, respectively. Moreover, we experiment with three very different LLMs in our paper: GPT-2 XL [38], LLaMA2-7B [45], and Gemma3-12B [44].

B.2 Editing Methods

We primarily use the R-ROME [12] editing method in our experiments due to its superlative editing performance across metrics. This choice is further motivated by its design as an extension of ROME

that more efficiently supports sequential edits. To assess the generality of our approach across different editing methods, we additionally evaluate on other editing methods, such as EMMET [13] and ROME [31].

B.3 Performance Metrics

We evaluate editing performance using five commonly used metrics: *Rewrite Accuracy*, *Rephrase Accuracy*, *Locality*, *Portability*, and *Fluency*, following the evaluation protocol of Zhang et al. [53] as implemented in EasyEdit [48]. Not all datasets support all metrics [48, 53], and all metrics indicate better editing performance for higher values. Each metric is computed per edit request and averaged over the test set.

Rewrite Accuracy. measures whether the post-edit model produces the new target knowledge when queried with the edit prompt [31, 53]. The target is scored under teacher forcing: at each target token position, the model's greedy prediction is compared to the corresponding target token, and the score is the fraction of matching tokens.

Rephrase Accuracy. measures whether the edit generalizes to a semantically equivalent paraphrase of the edit prompt [51, 53]. It is computed identically to *Rewrite Accuracy*, but on the paraphrased prompt.

Locality. measures whether knowledge unrelated to the edit remains unaffected [31, 51, 53]. Following Zhang et al. [53], it is computed over out-of-scope prompts as the fraction of token positions at which the post-edit model's prediction matches the pre-edit model's prediction. These prompts are unrelated questions for *ZSRE*, neighborhood prompts about other subjects for *Counterfact*, relation-specificity prompts for *WikiBio*, and relation-specificity and forgetfulness prompts for *WikiCounterfact* and *WikiRecent*.

Portability. measures whether the edit generalizes beyond the edit prompt to related queries [51, 53]. It is computed identically to *Rewrite Accuracy*, but on one-hop reasoning questions for *ZSRE* and *Counterfact*, and on subject-aliasing, reasoning, and logical-generalization questions for *WikiCounterfact* and *WikiRecent*, averaged across the available question types.

Fluency. evaluates the generative capacity of the post-edit model [31, 53]. It is computed as the weighted average of the bi-gram and tri-gram entropies of the text generated from the edit prompt, where a lower value indicates more repetitive text. Since this entropy does not lie in $[0, 1]$, we min-max normalize it over all edit requests and layers of a given model and dataset.

Finally, we report an **Overall** score as the average of all metrics supported by a dataset, each min-max normalized over all edit requests and layers of a given model and dataset.

C Results on Additional Datasets

C.1 Proxy-Selected versus Test-Selected Layer Editing Performances for *WikiBio* and *Counterfact*

Figure 5 presents the knowledge editing performance on GPT-2 XL with R-ROME editing method when using golden layers identified on a proxy set versus a test set on *WikiBio* and *WikiRecent*. We observe that the editing performance of the golden layers identified from the proxy set and the test set remains very close, similar to trends observed in the main paper.

C.2 Proxy Golden Layer versus Test Golden Layer on *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, *Counterfact*

Tables 4, 5, 6, 7, 8, and 9, report *Rewrite Accuracy*, *Rephrase Accuracy*, *Locality*, *Portability*, *Fluency*, and *Overall* metrics evaluated on the test set of the Proxy Optimal Layer and the Golden Layer under R-ROME editing across *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact* datasets and across three LLMs: GPT-2 XL, LLaMA2-7B, and Gemma3-12B. We observe that the golden layers identified from the proxy set and the test set often coincide. The resulting editing performance remains largely consistent, even when the selected golden layers differ.

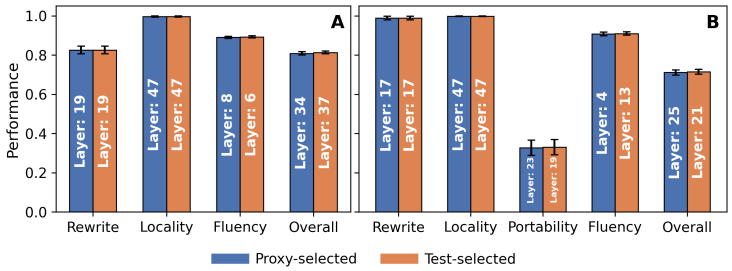

Figure 5: Performance of golden layers selected via the proxy and test sets with GPT-2 XL on (A) *WikiBio* and (B) *WikiRecent*. Editing performance evaluation is conducted on the test set queries.

Table 4: The *Rewrite Accuracy* of the Proxy Optimal Layer (top performing layer in the proxy set) and Golden Layer (top performing layer in the test set), for each of the editing performance metrics on the *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact* datasets on three LLMs, GPT-2 XL, LLaMA2-7B, and Gemma3-12B editing with R-ROME. The optimal layer selection is based on *Rewrite Accuracy* evaluated on the test set.

Model	Set	*ZSRE*		*WikiBio*		*WikiCounterfact*		*WikiRecent*		*Counterfact*	
		Layer	Rewrite	Layer	Rewrite	Layer	Rewrite	Layer	Rewrite	Layer	Rewrite
GPT-2 XL	Proxy	15	0.9962	19	0.8256	12	0.9380	17	0.9887	16	0.9796
	Test	19	0.9993	19	0.8256	15	0.9537	17	0.9887	18	0.9871
LLaMA2-7B	Proxy	12	0.9650	18	0.9835	3	0.9888	12	0.9783	2	0.9969
	Test	10	0.9667	13	0.9912	12	0.9922	19	0.9792	4	0.9995
Gemma3-12B	Proxy	27	0.9328	13	0.8538	19	0.9340	11	0.9784	8	0.9941
	Test	18	0.9763	13	0.8538	13	0.9415	19	0.9789	35	1.0000

Table 5: The *Rephrase Accuracy* of the Proxy Optimal Layer (top performing layer in the proxy set) and Golden Layer (top performing layer in the test set), for each of the editing performance metrics on the *ZSRE* and *Counterfact* datasets on three LLMs, GPT-2 XL, LLaMA2-7B, and Gemma3-12B editing with R-ROME. The optimal layer selection is based on *Rephrase Accuracy* evaluated on the test set.

Model	Set	*ZSRE*		*Counterfact*	
		Layer	Rephrase	Layer	Rephrase
GPT-2 XL	Proxy	16	0.8823	18	0.8034
	Test	17	0.8838	18	0.8034
LLaMA2-7B	Proxy	7	0.9136	6	0.8023
	Test	6	0.9176	15	0.8196
Gemma3-12B	Proxy	39	0.9108	38	0.9221
	Test	40	0.9294	40	0.9353

Table 6: The *Locality* of the Proxy Optimal Layer (top performing layer in the proxy set) and Golden Layer (top performing layer in the test set), for each of the editing performance metrics on the *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact* datasets on three LLMs, GPT-2 XL, LLaMA2-7B, and Gemma3-12B editing with R-ROME. The optimal layer selection is based on *Locality* evaluated on the test set.

Model	Set	*ZSRE*		*WikiBio*		*WikiCounterfact*		*WikiRecent*		*Counterfact*	
		Layer	Locality	Layer	Locality	Layer	Locality	Layer	Locality	Layer	Locality
GPT-2 XL	Proxy	47	0.9959	46	0.9882	47	0.9965	47	0.9977	47	0.9968
	Test	47	0.9959	47	0.9964	47	0.9965	47	0.9977	47	0.9968
LLaMA2-7B	Proxy	23	0.9895	23	0.8436	30	0.7402	25	0.6865	2	0.9747
	Test	3	0.9927	23	0.8436	30	0.7402	24	0.7016	2	0.9747
Gemma3-12B	Proxy	11	0.9795	46	0.7870	46	0.7411	0	0.6523	0	0.9305
	Test	11	0.9795	15	0.8476	46	0.7411	46	0.7005	0	0.9305

Table 7: The *Portability* of the Proxy Optimal Layer (top performing layer in the proxy set) and Golden Layer (top performing layer in the test set), for each of the editing performance metrics on the *ZSRE*, *WikiCounterfact*, *WikiRecent*, and *Counterfact* datasets on three LLMs, GPT-2 XL, LLaMA2-7B, and Gemma3-12B editing with R-ROME. The optimal layer selection is based on *Portability* evaluated on the test set.

Model	Set	*ZSRE*		*WikiCounterfact*		*WikiRecent*		*Counterfact*	
		Layer	Portability	Layer	Portability	Layer	Portability	Layer	Portability
GPT-2 XL	Proxy	23	0.4740	1	0.3032	23	0.3271	25	0.4392
	Test	29	0.4844	1	0.3032	19	0.3302	15	0.4450
LLaMA2-7B	Proxy	11	0.5979	17	0.6154	8	0.5719	15	0.5336
	Test	12	0.5993	16	0.6199	16	0.5839	3	0.5476
Gemma3-12B	Proxy	23	0.5442	43	0.6594	26	0.5806	1	0.4291
	Test	22	0.5480	42	0.6698	23	0.6103	0	0.4370

Table 8: The *Fluency* of the Proxy Optimal Layer (top performing layer in the proxy set) and Golden Layer (top performing layer in the test set), for each of the editing performance metrics on the *WikiBio*, *WikiCounterfact*, and *WikiRecent* datasets on three LLMs, GPT-2 XL, LLaMA2-7B, and Gemma3-12B editing with R-ROME. The optimal layer selection is based on *Fluency* evaluated on the test set.

Model	Set	*WikiBio*		*WikiCounterfact*		*WikiRecent*	
		Layer	Fluency	Layer	Fluency	Layer	Fluency
GPT-2 XL	Proxy	8	0.8897	45	0.9058	4	0.9081
	Test	6	0.8927	46	0.9157	13	0.9095
LLaMA2-7B	Proxy	5	0.8595	6	0.8824	4	0.8800
	Test	6	0.8597	3	0.8837	4	0.8800
Gemma3-12B	Proxy	32	0.7071	35	0.5824	32	0.6031
	Test	35	0.7180	32	0.5973	32	0.6031

Table 9: The *Overall* of the Proxy Optimal Layer (top performing layer in the proxy set) and Golden Layer (top performing layer in the test set), for each of the editing performance metrics on the *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact* datasets on three LLMs, GPT-2 XL, LLaMA2-7B, and Gemma3-12B editing with R-ROME. The optimal layer selection is based on *Overall* evaluated on the test set.

Model	Set	*ZSRE*		*WikiBio*		*WikiCounterfact*		*WikiRecent*		*Counterfact*	
		Layer	Overall	Layer	Overall	Layer	Overall	Layer	Overall	Layer	Overall
GPT-2 XL	Proxy	16	0.8286	34	0.8083	15	0.6916	25	0.7108	9	0.7726
	Test	16	0.8286	37	0.8133	17	0.6961	21	0.7139	16	0.7789
LLaMA2-7B	Proxy	6	0.8638	16	0.8771	14	0.7656	13	0.7625	3	0.8255
	Test	7	0.8645	16	0.8771	14	0.7656	12	0.7640	3	0.8255
Gemma3-12B	Proxy	19	0.8261	13	0.7780	41	0.6411	15	0.6474	9	0.7361
	Test	19	0.8261	15	0.7793	41	0.6411	21	0.6529	11	0.7386

C.3 *WikiRecent* Results Across Different Editing Methods and LLMs

Table 10 presents the editing performance of three editing methods: R-ROME, EMMET, and ROME on GPT-2 XL for the *WikiRecent* dataset. Notably, multiple layers often achieved identical performance scores; in such cases, we report results for the first tied layer. Table 11 presents the editing performance of R-ROME editing method on three models GPT-2 XL, LLaMA2-7B, and Gemma3-12B for the *WikiRecent* dataset. The table shows trends consistent with those reported in the main paper, with LGA-selected layers achieving significantly higher Overall editing performance than CMA across all datasets. LGA also consistently outperforms CMA on *Rewrite Accuracy* and *Rephrase Accuracy*, demonstrating robust improvements of our proposed approach in the case of *WikiRecent* dataset, similar to the datasets presented in the main paper: *ZSRE*, *WikiBio*, *WikiCounterfact*, and *Counterfact*.

Table 10: Editing performance evaluation of LGA, SaLEM, and CMA across different knowledge editing methods and *WikiRecent* dataset on GPT-2 XL. Different performance metrics are reported, where *Rewrite Accuracy* $\rightarrow$ RwA, *Locality* $\rightarrow$ LOC, *Portability* $\rightarrow$ PRT, *Fluency* $\rightarrow$ FLC, and *Overall* $\rightarrow$ OV. Layers are identified using the proxy set for both methods and evaluation is undertaken on an unseen test set.

Edit	Selection	$\text{RwA}_{(\uparrow)}$	$\text{LOC}_{(\uparrow)}$	$\text{PRT}_{(\uparrow)}$	$\text{FLC}_{(\uparrow)}$	$\text{OV}_{(\uparrow)}$
R-ROME	CMA	0.9622	0.6124	0.3038	0.9084	0.6967
	SaLEM	0.9828	0.6170	0.3192	0.9090	0.7070
	LGA (Ours)	0.9735	0.6058	0.3161	0.9048	0.7000
EMMET	CMA	0.9837	0.5029	0.2979	0.8568	0.6603
	SaLEM	0.9922	0.5485	0.3097	0.8475	0.6745
	LGA (Ours)	0.9892	0.5199	0.3044	0.8491	0.6656
ROME	CMA	0.9674	0.6105	0.3033	0.9106	0.6980
	SaLEM	0.9874	0.6136	0.3208	0.9089	0.7077
	LGA (Ours)	0.9773	0.6027	0.3149	0.9077	0.7007

D Sequential Editing

For sequential editing experiments, we apply 10 edits sequentially, where the same randomly selected samples are used across datasets. This choice follows prior work showing that performance degradation becomes noticeable at around 10 sequential edits [32, 53]. Additionally, prior studies have also evaluated smaller sequential edit settings to analyze sequential editing behavior [53], further motivating this setting. We present the performance of sequential edits in Table 12, where it is evident that LGA selected layers perform better compared to CMA and SaLEM selected layers.

Table 11: Editing performance evaluation of LGA, CMA, and SaLEM for *WikiRecent* dataset and across three models: GPT-2 XL, LLaMA2-7B and Gemma3-12B with R-ROME. Different performance metrics are reported, where *Rewrite Accuracy* → RwA, *Locality* → LOC, *Portability* → PRT, *Fluency* → FLC, and *Overall* → OV. Layers are identified using the proxy set for both methods, and the evaluation is undertaken on an unseen test set.

Model	Selection	RwA	LOC	PRT	FLC	OV
GPT-2 XL	CMA	0.9622	0.6124	0.3038	0.9084	0.6967
	SaLEM	0.9828	0.6170	0.3192	0.9090	0.7070
	LGA (Ours)	0.9735	0.6058	0.3161	0.9048	0.7000
LLaMA2-7B	CMA	0.9748	0.5629	0.5731	0.8718	0.7456
	SaLEM	0.9768	0.5478	0.5769	0.8708	0.7431
	LGA (Ours)	0.9750	0.5560	0.5699	0.8771	0.7445
Gemma3-12B	CMA	0.9710	0.5796	0.5216	0.5020	0.6436
	SaLEM	0.9604	0.5166	0.5558	0.4884	0.6303
	LGA (Ours)	0.9775	0.5779	0.5327	0.5013	0.6474

Table 12: Sequential editing performance with 10 sequential edits for LGA, SaLEM, and CMA across three models: GPT-2 XL, LLaMA2-7B, and Gemma3-12B on five datasets: *ZSRE*, *WikiBio*, *WikiCounterFact*, *CounterFact*, and *WikiRecent*. The same randomly selected edit samples are applied sequentially for each dataset. Different datasets support different performance metrics, where *Rewrite Accuracy* → RwA, *Rephrase Accuracy* → RpA, *Locality* → LOC, *Portability* → PRT, *Fluency* → FLC, and *Overall* → OV. Layers are identified using the proxy set for each method, and evaluation is conducted on unseen test data. Results demonstrate performance differences under sequential editing, with LGA-selected layers achieving a mean overall performance gain of 7.9% and 511% over CMA and SaLEM, respectively.

ZSRE

Edit	Selection	RwA	RpA	LOC	PRT	OV
GPT-2 XL	CMA	0.5083	0.5083	0.6917	0.4583	0.5417
	SaLEM	0.1750	0.2083	0.0250	0.0833	0.1229
	LGA	0.8267	0.7817	0.7545	0.6117	0.7436
LLAMA2-7B	CMA	0.9689	0.8822	1.0000	0.6862	0.8843
	SaLEM	0.9489	0.9489	0.8908	0.6198	0.8521
	LGA	0.9689	0.9689	0.9933	0.7364	0.9169
GEMMA3-12B	CMA	0.6500	0.5917	0.6622	0.5833	0.6218
	SaLEM	0.6106	0.5856	0.8847	0.6483	0.6823
	LGA	0.4894	0.4117	0.6601	0.5167	0.5195

WikiBio

Edit	Selection	RwA	LOC	FLC	OV
GPT-2 XL	CMA	0.5120	0.4042	0.8770	0.4581
	SaLEM	0.0865	0.0912	0.8368	0.0889
	LGA	0.5566	0.3542	0.9083	0.4554
LLaMA2-7B	CMA	0.8767	0.5217	0.6215	0.6992
	SaLEM	0.8882	0.6279	0.7436	0.7580
	LGA	0.9367	0.5158	0.6439	0.7263
GEMMA3-12B	CMA	0.7119	0.7336	0.6771	0.7075
	SaLEM	0.6574	0.7783	0.6157	0.7179
	LGA	0.8124	0.8001	0.6666	0.7597

WikiCounterFact

Edit	Selection	RwA	LOC	PRT	FLC	OV
GPT-2 XL	CMA	0.7444	0.3983	0.3449	0.8768	0.4959
	SaLEM	0.0333	0.0014	0.0000	0.7531	0.0116
	LGA	0.9800	0.4877	0.4826	0.8769	0.6501
LLaMA27B	CMA	0.9857	0.4137	0.6217	0.8671	0.6737
	SaLEM	0.9857	0.5243	0.6571	0.9125	0.7224
	LGA	1.0000	0.4126	0.6907	0.9146	0.7011
GEMMA3-12B	CMA	0.5278	0.5011	0.3534	0.4155	0.4608
	SaLEM	0.3556	0.4712	0.2789	0.4243	0.3685
	LGA	0.4806	0.6750	0.4196	0.4143	0.5250

Counterfact

Edit	Selection	RwA	RpA	LOC	PRT	OV
GPT-2 XL	CMA	0.5000	0.4000	0.3000	0.2350	0.3588
	SaLEM	0.0000	0.1000	0.0000	0.0500	0.0375
	LGA	1.0000	0.4000	0.3000	0.2267	0.4817
LLaMA2-7B	CMA	1.0000	0.9000	0.4500	0.2250	0.6438
	SaLEM	1.0000	0.9000	0.3500	0.2583	0.6271
	LGA	1.0000	0.8000	0.9000	0.3083	0.7521
GEMMA3-12B	CMA	1.0000	0.6000	0.9000	0.3000	0.7000
	SaLEM	0.9000	0.5000	0.7000	0.3000	0.6000
	LGA	0.9000	0.5000	0.8000	0.3000	0.6250

WikiRecent

Edit	Selection	RwA	RpA	LOC	PRT	OV
GPT-2 XL	CMA	0.8130	0.6465	0.3522	0.8885	0.6039
	SaLEM	0.8702	0.5677	0.3460	0.9298	0.5946
	LGA	0.8759	0.6559	0.3431	0.8492	0.6250
LLaMA2-7B	CMA	0.9500	0.6109	0.6145	0.8115	0.7251
	SaLEM	0.8708	0.6491	0.4976	0.9009	0.6725
	LGA	0.9333	0.6209	0.5769	0.7983	0.7104
GEMMA3-12B	CMA	0.9333	0.6894	0.5760	0.5673	0.7329
	SaLEM	0.7238	0.7073	0.4219	0.3157	0.6177
	LGA	0.9071	0.6142	0.5030	0.4689	0.6748

E Composite, Longform, and Unstructured Editing

E.1 Composite Editing

We evaluate compositional model editing on the *PeakCF* dataset, following the recent protocols while reporting SRS (compositional success rate), SR-1 (single-edit success rate), GSR (generalization success rate), and LSR (Rouge-L-based generation score) as evaluation metrics [28, 36]. Table 13 presents the performance of layers selected by LGA, SaLEM, and CMA. LGA achieves improvements of 2.38% and 49.30% over CMA and SaLEM, respectively, in terms of average performance. Layer selection is performed using a proxy set of size 30, and evaluation is conducted on a test set of size 270 randomly selected from the *PeakCF* dataset. For the task of composite editing, Tukey constant is fixed to 0.5.

Table 13: Compositional editing performance of LGA-selected layers, SaLEM, and CMA on the *PeakCF* dataset across three models: GPT-2 XL, LLaMA2-7B, and Gemma3-12B. We use R-ROME for editing and evaluate using SRS, SR-1, GSR, and LSR metrics. Layer selection uses a proxy set of size 30, and evaluation is performed on 270 randomly sampled test examples from the PeakT dataset. Layers selected by LGA achieve average overall performance improvements of 2.38% and 49.30% over CMA and SaLEM, respectively.

Model	Method	SRS ↑	SR-1 ↑	GSR ↑	LSR ↑	Overall ↑
GPT-2 XL	CMA	0.4562	0.1630	0.4023	0.0105	0.3462
	SaLEM	0.2663	0.1556	0.2073	0.0154	0.1611
	LGA	0.5513	0.4278	0.4357	0.0197	**0.3586**
LLAMA2-7B	CMA	0.6417	0.4944	0.5297	0.0282	0.4235
	SaLEM	0.6417	0.4944	0.5297	0.0282	0.4235
	LGA	0.6513	0.4926	0.5398	0.0290	**0.4282**
GEMMA3-12B	CMA	0.3986	0.2593	0.3150	0.0214	0.2485
	SaLEM	0.3417	0.1741	0.2845	0.0200	0.2050
	LGA	0.3939	0.2889	0.3103	0.0252	**0.2546**

E.2 Longform Editing

Following prior work on long-form model editing, we evaluate on the *WikiBio* and *UnKE* datasets under a long-form generation setting. Unlike short-form editing, which only evaluates whether edited facts appear in short continuations (i.e., next-token or limited-length generation), long-form editing requires the model to consistently maintain and integrate factual updates throughout extended natural language generation, such as full biographies [39]. For instance, given the query ‘‘`Eleanor Arnason is an American science fiction and fantasy writer`’’ and an edited knowledge is a long statement such as ‘‘`She is best known for her novel A Woman of the Iron People (1991), which won the James Tiptree, Jr. Award and was a finalist for the Nebula Award for Best Novel.`’’ The performance of LGA-selected layers on the *UnKE* dataset is presented in Table 14 while *WikiBio* dataset is presented in Figure 3 and Table 2, where LGA overall outperforms CMA and SaLEM.

E.3 Unstructured Editing

We perform unstructured model editing on the *UnKE* dataset, adhering to recent protocols and reporting BLEU [35] (Bilingual Evaluation Understudy) and MMLU (general knowledge accuracy benchmark score) along with various rogue scores [26] like: ROUGE-1 (unigram overlap score), ROUGE-2 (bigram overlap score), ROUGE-L (longest common subsequence-based score), Para-ROUGE-L (ROUGE-L score of paraphrased queries), BERTScore (embedding-based semantic similarity score) as evaluation metrics [7]. Table 14 presents the performance of layers selected by LGA achieve the highest overall score on GPT2-XL and Llama2-7B, while remaining competitive on Gemma3-12B, outperforming CMA and SaLEM in 2 out of 3 settings. Layer selection is carried out using a proxy set of size 15, while evaluation is performed on a test set of size 135 randomly sampled from the *UnKE* dataset. For the task, the Tukey constant is set to 0.5.

Table 14: Unstructured editing performance of LGA-selected layers, SaLEM, and CMA on the *UnKE* dataset across three models: GPT-2 XL, LLaMA2-7B, and Gemma3-12B. We use R-ROME for editing and evaluate using BLEU, ROUGE-1 (R-1), ROUGE-2 (R-2), ROUGE-L (R-L), Para-ROUGE-L (Para-R-L), BERTScore, and MMLU metrics. Layer selection uses a proxy set of size 15, and evaluation is performed on 135 randomly sampled test examples from the *UnKE* dataset. Layers selected by LGA best overall score on GPT2-XL and Llama2-7B, and remain competitive on Gemma3-12B, winning 2 out of 3 settings compared to CMA and SaLEM.

Model	Method	BLEU ↑	R-1 ↑	R-2 ↑	R-L ↑	Para-R-L ↑	BertScore ↑	MMLU ↑	Overall ↑
GPT2-XL	CMA	0.3182	0.1940	0.0732	0.1841	0.1678	0.6125	0.2133	0.2519
	SaLEM	0.3474	0.2219	0.0990	0.2097	0.1685	0.6926	0.2207	0.2800
	LGA	0.3452	0.2272	0.1036	0.2179	0.1855	0.6895	0.2237	**0.2846**
Llama2-7B	CMA	0.5735	0.5310	0.3889	0.5152	0.3776	0.8250	0.4178	0.5184
	SaLEM	0.5435	0.4837	0.3228	0.4654	0.3610	0.8005	0.4015	0.4826
	LGA	0.6342	0.6492	0.5508	0.6398	0.3230	0.8594	0.4296	**0.5837**
Gemma3-12B	CMA	0.4060	0.2971	0.1309	0.2788	0.2229	0.6774	0.6741	0.3839
	SaLEM	0.4836	0.4105	0.2321	0.3862	0.3084	0.7541	0.7289	**0.4720**
	LGA	0.3856	0.2578	0.1017	0.2394	0.2197	0.6501	0.6963	0.3644

F Comparison with Random Layer Selection

Table 15 compares the performance of layers selected by LGA with randomly selected layers across three models on the *ZSRE* dataset. For the random baseline, three layers are sampled uniformly at random and the mean performance is reported. As shown in the table, LGA consistently outperforms random selection in overall performance across all models, with notable improvements in rewrite and rephrase accuracy.

Table 15: Editing performance evaluation of LGA-selected layers and randomly selected layers, denoted as *Random*, for the *ZSRE* dataset across three models: GPT-2 XL, LLaMA2-7B, and Gemma3-12B using R-ROME. For the random baseline, three layers are selected uniformly at random, and their mean performance is reported. The table presents performance metrics, namely *Rewrite Accuracy*, *Rephrase Accuracy*, *Locality*, *Portability*, and *Overall*. Layers for LGA are identified using the proxy set, and evaluation is conducted on an unseen test set, demonstrating performance improvements over random selection.

Dataset	Model	Method	Rewrite ↑	Rephrase ↑	Locality ↑	Portability ↑	Overall ↑
ZSRE	GPT-2 XL	CMA	0.9665	0.7467	0.9608	0.4837	0.6942
		Random	0.8539	0.5916	0.9701	0.4784	0.6347
		LGA	0.9851	0.8151	0.9543	0.4757	**0.7106**
	LLAMA2-7B	CMA	0.9587	0.8859	0.9923	0.5827	0.8549
		Random	0.9576	0.8115	0.9885	0.5650	0.8307
		LGA	0.9625	0.8988	0.9927	0.5950	**0.8623**
	GEMMA3-12B	CMA	0.8581	0.7041	0.9421	0.4989	0.7508
		Random	0.8197	0.7692	0.7988	0.4465	0.7086
		LGA	0.9535	0.8796	0.7736	0.5198	**0.7816**

G LGA Without Cross-Knowledge Interaction

SaLEM considers only the gradient of the new knowledge K_i'. To ensure a fair comparison using only the information utilized by SaLEM, we further evaluate a simplified variant of LGA that removes cross-knowledge interaction between the old-knowledge context $(Q_i \cup K_i)$ and the new-knowledge context $(Q_i \cup K_i')$, and instead selects layers based solely on the ℓ_2-norm of the gradient computed from $(Q_i \cup K_i')$, denoting as LGA(K'). From Table 16, compared to SaLEM, the ℓ_2-norm variant achieves a small relative improvement of approximately $+1.6\%$ in overall performance, while LGA yields a substantial improvement of approximately $+13.3\%$. These results further highlight the

importance of modeling cross-knowledge interactions between old and new knowledge for effective layer selection.

Table 16: Editing performance comparison between CMA, SaLEM, a simplified ℓ_2-norm variant of LGA, and the full LGA method for layer selection on the *WikiBio* dataset across three models: GPT-2 XL, LLaMA2-7B, and Gemma3-12B using R-ROME. The ℓ_2-norm variant removes cross-knowledge interaction between the old-knowledge context $(Q_i \cup K_i)$ and the new-knowledge context $(Q_i \cup K_i')$, and instead selects layers based solely on the gradient computed from $(Q_i \cup K_i')$, denoting as LGA(K'). The table reports *Rewrite Accuracy*, *Rephrase Accuracy*, *Locality*, *Portability*, and *Overall editing performance*. Results show that while the ℓ_2-norm variant achieves a small improvement over SaLEM, LGA significantly outperforms both methods, highlighting the importance of modeling cross-knowledge interactions for effective layer selection.

Dataset	Model	Method	Rewrite ↑	Locality ↑	Fluency ↑	Overall ↑
WikiBio	GPT-2 XL	CMA	0.6894	0.5692	0.8927	0.7171
		SaLEM	0.4061	0.2563	0.8186	0.4937
		LGA (K')	0.4061	0.2563	0.8186	0.4937
		LGA	0.7484	0.5587	0.8906	0.7326
	LLaMA2-7B	CMA	0.8818	0.6515	0.8547	0.7960
		SaLEM	0.8396	0.6526	0.8529	0.7817
		LGA (K')	0.9560	0.6357	0.8597	0.8172
		LGA	0.9560	0.6357	0.8597	0.8172
	GEMMA3-12B	CMA	0.6817	0.6546	0.6683	0.6682
		SaLEM	0.8153	0.8062	0.6668	0.7628
		LGA (K')	0.8124	0.8001	0.6666	0.7597
		LGA	0.8124	0.8001	0.6666	0.7597

H Leave-One-Dataset-Out Proxy Selection

For every test set across the datasets (*ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact*) and LLMs (GPT-2 XL, LLaMA2-7B, Gemma3-12B), we evaluate the *Rewrite Accuracy* of the editing layers selected by the (independent) proxy set of each of the other datasets. We then compare this independent proxy-selected layer performance to the layer selected by the (dependent) proxy set drawn from the same dataset as the test set. Note that the performance in both cases is evaluated on the same fixed target test set. Table 17 reports the results.

Across these settings, we observe that the independent proxy set layer performance remains very close to the dependent proxy set layer performance, with an average gap of 0.0179. This indicates cross-dataset generalizability of the proxy set based on *Rewrite Accuracy*.

We further conduct the same evaluation using the *Overall* metric. Since *Overall* aggregates the metrics supported by each dataset (Appendix B.3), we compare only datasets that support the same set of metrics: *ZSRE* and *Counterfact* (*Rewrite Accuracy*, *Rephrase Accuracy*, *Locality*, and *Portability*), and *WikiCounterfact* and *WikiRecent* (*Rewrite Accuracy*, *Locality*, *Portability*, and *Fluency*). As shown in Table 18, the independent proxy set layer performance remains close to the dependent proxy set layer performance, with an average gap of 0.0197.

I Limitations

As we show through extensive experiments across multiple LLM architectures, benchmark datasets, and editing methods, our proposed Layer Gradient Analysis (LGA) approach outperforms CMA in editing performance and computational efficiency. As a result, our work enables more reliable, efficient, and targeted updates to LLMs, which is critical as these models are increasingly deployed in real-world applications. However, as with any work, there are limitations to our approach that can be addressed in future work. For instance, a limitation that can be addressed is the need for further study and benchmarks of golden layer estimation and knowledge editing methods in LLMs, extending beyond the models, datasets, and editing strategies explored in this work. Future work could also explore strategies to make LGA more robust to extreme or unusual gradients, further improving its reliability and applicability across various LLMs and datasets.

Table 17: Cross-dataset generalizability of the proxy set selected golden layer via leave-one-dataset-out evaluation with GPT-2 XL, LLaMA2-7B, and Gemma3-12B on the *ZSRE*, *WikiBio*, *WikiCounterfact*, *WikiRecent*, and *Counterfact* test sets. For each evaluation test set, we report the *Rewrite Accuracy* of the layers selected by the (independent) proxy sets drawn from the other datasets, their mean, and that of the layer selected by the (dependent) proxy set drawn from the same dataset as the test set. The independent proxy sets attain performance close to the dependent proxy set.

Evaluation Test set: *ZSRE*

	Independent proxy					Dependent proxy
Model	*WikiBio*	*WikiCounterfact*	*WikiRecent*	*Counterfact*	Mean	*ZSRE*
GPT-2 XL	0.9993	0.9935	0.9989	0.9976	0.9973	0.9962
LLaMA2-7B	0.9621	0.9650	0.9650	0.9587	0.9627	0.9654
Gemma3-12B	0.9754	0.9754	0.9317	0.9488	0.9578	0.9328

Evaluation Test set: *WikiBio*

	Independent proxy					Dependent proxy
Model	*ZSRE*	*WikiCounterfact*	*WikiRecent*	*Counterfact*	Mean	*WikiBio*
GPT-2 XL	0.7813	0.7484	0.8192	0.8019	0.7877	0.8256
LLaMA2-7B	0.9838	0.9851	0.9864	0.8396	0.9487	0.9835
Gemma3-12B	0.7050	0.8522	0.8448	0.6807	0.7707	0.8522

Evaluation Test set: *WikiCounterfact*

	Independent proxy					Dependent proxy
Model	*ZSRE*	*WikiBio*	*WikiRecent*	*Counterfact*	Mean	*WikiCounterfact*
GPT-2 XL	0.9537	0.9433	0.9491	0.9498	0.9490	0.9380
LLaMA2-7B	0.9907	0.9893	0.9922	0.9806	0.9882	0.9913
Gemma3-12B	0.8827	0.9340	0.9380	0.9056	0.9151	0.9340

Evaluation Test set: *WikiRecent*

	Independent proxy					Dependent proxy
Model	*ZSRE*	*WikiBio*	*WikiCounterfact*	*Counterfact*	Mean	*WikiRecent*
GPT-2 XL	0.9845	0.9846	0.9789	0.9855	0.9834	0.9887
LLaMA2-7B	0.9779	0.9760	0.9780	0.9674	0.9748	0.9783
Gemma3-12B	0.9232	0.9789	0.9789	0.9316	0.9532	0.9784

Evaluation Test set: *Counterfact*

	Independent proxy					Dependent proxy
Model	*ZSRE*	*WikiBio*	*WikiCounterfact*	*WikiRecent*	Mean	*Counterfact*
GPT-2 XL	0.9796	0.9871	0.9689	0.9828	0.9796	0.9796
LLaMA2-7B	0.9979	0.9952	0.9984	0.9979	0.9974	0.9969
Gemma3-12B	0.9930	0.9866	0.9866	0.9952	0.9904	0.9946

Table 18: Cross-dataset generalizability of the proxy set via leave-one-dataset-out evaluation using the *Overall* metric with GPT-2 XL, LLaMA2-7B, and Gemma3-12B, between datasets that support the same set of metrics. For each evaluation test set, we report the *Overall* performance of the layer selected by the (independent) proxy set drawn from the other dataset and that of the layer selected by the (dependent) proxy set drawn from the same dataset as the test set. The independent proxy set attains performance close to the dependent proxy set.

	Evaluation Test set: *ZSRE*		Evaluation Test set: *Counterfact*	
Model	Independent proxy *Counterfact*	Dependent proxy *ZSRE*	Independent proxy *ZSRE*	Dependent proxy *Counterfact*
GPT-2 XL	0.8106	0.8286	0.7789	0.7726
LLaMA2-7B	0.8623	0.8638	0.8077	0.8255
Gemma3-12B	0.7564	0.8261	0.6722	0.7361
	Evaluation Test set: *WikiCounterfact*		Evaluation Test set: *WikiRecent*	
Model	Independent proxy *WikiRecent*	Dependent proxy *WikiCounterfact*	Independent proxy *WikiCounterfact*	Dependent proxy *WikiRecent*
GPT-2 XL	0.6788	0.6916	0.7087	0.7108
LLaMA2-7B	0.7609	0.7656	0.7604	0.7625
Gemma3-12B	0.6209	0.6411	0.6307	0.6474

J Broader Impacts

Our work on knowledge editing in LLMs offers a promising direction for enabling efficient and targeted updates to model knowledge. There are many potential positive societal consequences of our work, such as enabling LLMs to efficiently incorporate updated knowledge without requiring brute-force re-training, improved editing performance at scale with lower computational overhead, as well as more robust and reliable models, among other benefits. Since knowledge editing requires access to model internals and can only be undertaken by model developers, there are no specific societal risks associated with our proposed approaches as such. Furthermore, our approach provides insights into which layers are most responsible for incorporating new knowledge, offering a practical perspective on model behavior and novel mechanistic insights. These insights may guide future research toward more effective and efficient knowledge editing in language models.

K Code and Reproducibility

We provide our code and implementation in an open-source repository: `https://github.com/Shrestha30/Golden-Layers-LGA`. All the experiments were conducted on a Linux server with 6x NVIDIA DGX B200 GPUs with 192 GB VRAM/GPU. All the utilized LLMs are original (unquantized) open-source versions from HuggingFace. During generation, we used greedy decoding without sampling. To ensure reproducibility, all sources of randomness, including PyTorch, NumPy, and Python's random module, were fixed using a seed of 42. The gradient of the loss with respect to the model's existing knowledge was computed for each input sequence, considering up to the number of tokens in the ground-truth target if available, or up to number of tokens in the new target knowledge. Our codebase was built upon the *EasyEdit* [48] framework, and evaluations of the edited models were performed following their standard conventions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions: empirical demonstration of golden layers, their generalizability via proxy datasets, and the LGA method for efficient golden layer identification. All claims are validated by experiments in Sections 4 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix I provides a discussion on Limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper is empirical in nature and does not include formal theoretical results, theorems, or proofs. The contributions are validated experimentally.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Appendices K and B describe relevant details, e.g. hardware, libraries, random seeds, decoding strategy, implementation details, etc. enabling full reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: An anonymized open-source repository is provided at (`https://anonymous.4open.science/r/Golden-Layers-LGA-6D08/`). All datasets used are publicly available benchmarks (ZSRE, WikiBio, WikiCounterfact, WikiRecent, Counterfact).

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendices K and B provide implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports mean ± standard deviation for golden layer vs. sample-wise optimal layer comparisons, with two-sided Student's t-test results explicitly reported. Figure 1 includes the standard error of the mean as error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix K specifies the compute environment: a Linux server with 6× NVIDIA DGX B200 GPUs, each with 192 GB VRAM. The paper also reports runtime comparisons between methods in Figure 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Code of ethics has been reviewed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, broader impacts are discussed in Appendix J.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper proposes a layer selection method for knowledge editing and does not require specific safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and models are cited with original references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new assets released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing and no research with human subjects. All experiments are conducted programmatically on LLMs using benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research was conducted. IRB approval is not required for this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.