A DYNAMIC MULTI-LAYER EDITOR FOR LIFELONG KNOWLEDGE ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

With the development of large language models, knowledge editing algorithms for updating and revising these models, particularly lifelong editing algorithms, have been continuously emerging in recent years. However, current methods typically update parameters in a single layer while overlooking the inter-layer correlations. To address this issue, we propose the DMKE algorithm. This method divides the language model into hierarchical blocks using windows and sliding strides, and selects the most suitable editing block for parameter updates by evaluating the degree of pattern matching between different blocks and the editing knowledge. Experiments demonstrate that in lifelong editing scenarios, our algorithm achieves significant performance improvements, comprehensively surpassing existing strong baseline methods.

1 INTRODUCTION

In recent years, with the continuous increase in model size, the performance of natural language processing tasks has been significantly improved (Brown et al., 2020; Black et al., 2022; Kale et al., 2022), mainly due to the fact that large language models encode abundant world knowledge through pre-training on large-scale corpora (Zhang et al., 2023; OpenAI et al., 2024; Zhao et al., 2026). However, outdated or incorrect knowledge may be retained in the models. For example, after January 20th, 2025, when we ask who the President of the United States is, we expect the answer given by the large language model to be Trump instead of Biden. Therefore, it is necessary to quickly update the model parameters and inject new knowledge into the large model without affecting irrelevant inputs and outputs (Jiang et al., 2024). For example, when updating the knowledge that "the current President of the United States is Trump", it is necessary not only to modify the answer to "Who is the spouse of the current President of the United States?" but also not to affect the answer to "What is Biden's year of birth?". The cost of retraining these models using updated corpora is extremely high (Touvron et al., 2023; Patterson et al., 2021), and performing additional fine-tuning on models for individual examples that need correction often leads to overfitting (Cao et al., 2021; Zhu et al., 2020).

To address this, knowledge editing methods (Sinitsin et al., 2020) have been proposed to modify specific knowledge by directly updating the internal parameters of LLMs. Existing approaches have made significant progress in efficiency and scalability. Some methods leverage low-rank gradient structures Mitchell et al. (2022a) or causal mediation analysis Meng et al. (2023a) to enable fast, single-edit modifications on very large pre-trained models. However, these approaches cannot support batch editing. Subsequent algorithms (Meng et al., 2023b; Chen et al., 2024; Zhang et al., 2024) have attempted to update thousands of memories simultaneously, yet they rely on fixed-layer selection strategies that ignore the correlation with the editing data. Hu et al. (2024) further restricts the editing scope to a single layer, lacking the flexibility needed for complex edits.

To overcome these limitations, we propose DMKE (Dynamic Multi-layer Knowledge Editor), a novel knowledge editing algorithm that achieves precise layer localization and efficient parameter modification through three key innovations: (1) a dynamic window-stride grouping mechanism that adaptively selects relevant MLP layers based on the characteristics of the knowledge to be edited; (2) a gradient-based knowledge relevance scoring module that identifies the optimal layer group for each editing target; and (3) a systematic analysis of how parameter configurations affect editing performance.

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

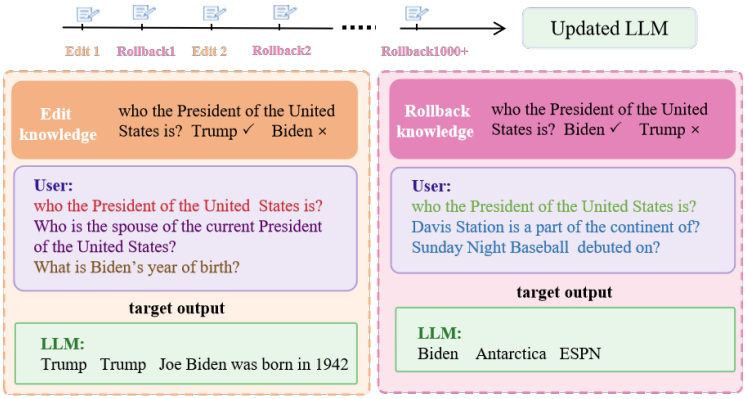


Figure 1: The long-term editing process of DMKE. Edit and rollback alternate: in the edit phase, questions in red, purple, and brown are used to evaluate the effectiveness, generalization, and locality of editing, respectively; in the rollback phase, questions in green and blue are used to evaluate the accuracy of rollback and the retention of unedited knowledge, respectively.

We evaluate DMKE on CounterFact Meng et al. (2023a) with GPT-2 XL Radford et al. (2019) and GPT-J-6B Wang & Komatsuzaki (2021). DMKE sets a new state-of-the-art across metrics, maintaining robust performance under 1,024 sequential edits. It surpasses leading baselines by 17.1 percentage points on GPT-2 XL and 41.1 percentage points on GPT-J-6B. Optimal configurations vary by model, with the best results occurring when window size is approximately twice the stride.

2 Related Work

The purpose of knowledge editing is to modify the inherent knowledge within a large language model (LLM) Yao et al. (2023) in order to achieve the desired output. Although fine-tuning a large model can also achieve the desired output by directing the model towards specific content, fine-tuning often leads to overfitting (Tirumala et al., 2022), which may result in catastrophic forgetting (Luo et al., 2025). Currently, knowledge editing methods can be categorized into the following types:

1) Memory-based methods. This approach does not modify the model's parameters but instead utilizes external memory as a medium to guide the model's output. For instance, GRACE Hartvigsen et al. (2023) uses a codebook as a query dictionary. Chen et al. (2024) employed retrieval-based methods to integrate retrieved knowledge into the model. Some models also use additional model structures as memory carriers to store knowledge (Dai et al., 2023; Wang et al., 2024; Yogatama et al., 2021; Mitchell et al., 2022b).

2) Meta-learning. Meta-learning refers to the process by which an individual or system acquires learning mechanisms, focusing on how to optimize the learning process itself. Sinitsin et al. (2020) proposed a two-layer meta-learning approach for model initialization and rapid fine-tuning. MEND Mitchell et al. (2022a) decouples finetuning gradients to improve the generalization of edits. To address the scalability issue in the number of concurrent editing operations in existing hypernetworks, MALMEN Tan et al. (2024) formulates parameter shift aggregation as a least squares problem, and then uses the normal equation to update the LM parameters.

3) Locate-Then-Edit. KN Dai et al. (2022) utilizes a knowledge attribution method to discover that the activation of knowledge neurons is positively correlated with the expression of the corresponding facts. Based on this observation, they employ knowledge neurons to edit (such as update or delete) specific knowledge without performing fine-tuning. ROME Meng et al. (2023a) uses causal tracing to locate the layer where knowledge is stored, and after identification, it performs Rank-One editing within that layer. MEMIT Meng et al. (2023b) distributes the trained Rank-One parameters evenly across multiple layers, successfully extending the multi-instance editing capability built upon ROME. Gupta et al. (2023) confirmed the causal link between commonsense judgments and editable parameters in Transformers, and proposed improvements to the MEMIT algorithm, including vary-

ing edit tokens and refining the layer selection strategy. WilKE Hu et al. (2024) selects the editing layer based on the degree of pattern matching for different editing knowledge across various layers.

3 PRELIMINARY

3.1 AUTOREGRESSIVE LANGUAGE MODEL

The generation process of an autoregressive language model is sequential and unidirectional: given a sequence $x_1, x_2, ..., x_{t-1}$, the model predicts the probability distribution for the t-th token, then samples from x_t, appends to the sequence, and continues predicting x_{t+1} in the same manner. Mathematically, this corresponds to decomposing the joint probability of the sequence as:

$$P(x_{1:T}) = \prod_{t=1}^{T} P(x_t|x_{1:t-1}), \tag{1}$$

where $x_{1:T}$ is a length-T token sequence and the prediction at step t conditions on $x_{<t}$:

$$P_\theta(x_t|x_{<t}) = Softmax(W_o \cdot h_t^{(L)}), \tag{2}$$

$h_t^{(L)}$ is the hidden state at position t in the L-th layer of the Transformer. W_o is the output projection matrix. Furthermore, the hidden state $h_t^{(l)}$ can be further expressed as:

$$\begin{aligned} h_t^{(l)} &= h_t^{(l-1)} + a_t^{(l)} + m_t^{(l)}, \\ a_t^{(l)} &= attn^{(l)}(a_1^{(l-1)}, a_2^{(l-1)}, ...a_t^{(l-1)}), \\ m_t^{(l)} &= W_{proj}^{(l)} \cdot \sigma(W_{fc}^{(l)} \gamma(a_t^{(l)} + h_t^{(l-1)})). \end{aligned} \tag{3}$$

where $attn^{(l)}$, $W_{proj}^{(l)}$, and $W_{fc}^{(l)}$ are the layer-l attention, attention output projection, and FFN intermediate weight; σ, γ are the rectified and normalization functions.

3.2 LIFELONG EDITING IN LARGE LANGUAGE MODELS

The goal of model editing is to use the given editing example (x_e, y_e) to modify the model, such that the base model's output becomes y_e on x_e and its equivalence neighborhood $x'_e \in N(x_e)$, while maintaining the model's behavior unchanged on unrelated inputs $x_{loc} \in E_{loc}$. For example, given the editing example "The Eiffel Tower is in Rome", the model should recognize that visiting it would require going to Rome, but should not assume that every other tourist attraction is also in Rome. In other words, the model's knowledge of facts should be robust to variations in phrasing and context without altering other facts. Assuming the models before and after editing are denoted as: f and f_θ, then model editing should satisfy:

$$f_\theta(x) = \begin{cases} y_e & \text{if } x \in N(x_e) \\ f(x) & \text{if } x \in E_{\text{loc}} \end{cases}. \tag{4}$$

Lifelong editing refers to continuing edits based on the previously edited model while still satisfying the aforementioned editing requirements, which can be expressed by the formula:

$$f_{\theta^i}(x^i) = \begin{cases} y_e^i & \text{if } x^i \in N(x_e^i) \\ f_{\theta^{i-1}}(x^i) & \text{if } x^i \in E_{\text{loc}}^i \end{cases}, \tag{5}$$

where $f_{\theta^i}(\cdot)$ denotes the model after the i-th edit, and x_e^i represents the knowledge for the i-th edit. In addition, lifelong editing must also satisfy the requirement of not altering previously completed edits, i.e., the Retention metric in WilKE Hu et al. (2024), which can be expressed by the formula:

$$f_{\theta^i}(x_e^j) = y_e^j, \tag{6}$$

where j satisfies $1 \leq j < i$.

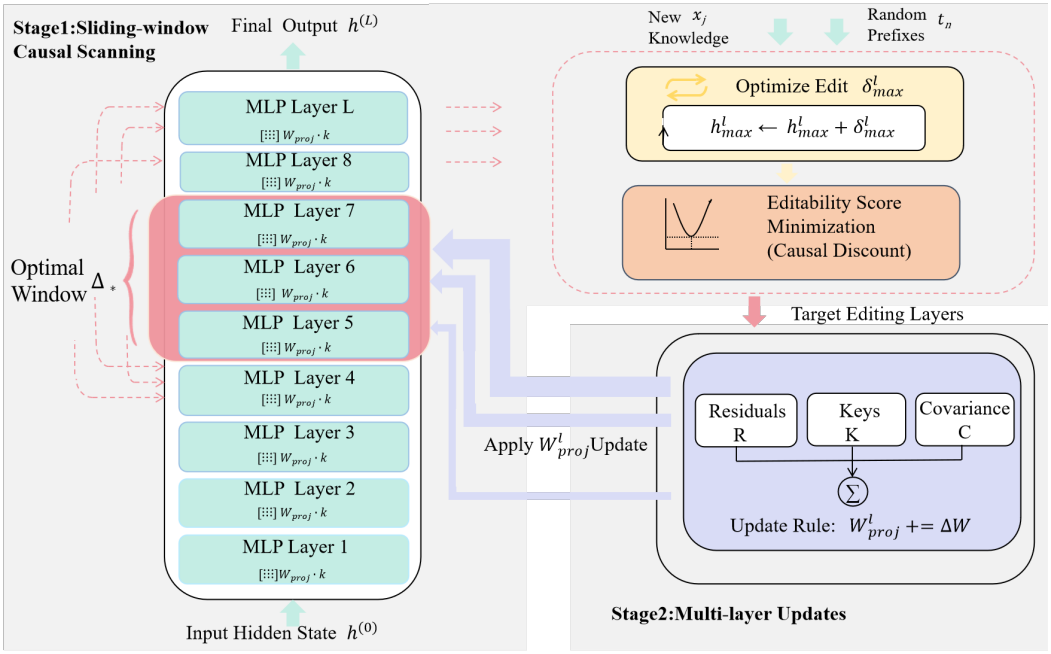


Figure 2: Schematic diagram of the DMKE knowledge adaptation process. The framework consists of two stages: (1) sliding-window causal scanning across MLP layers to identify target editing layers via layer-group editability scores, and (2) multi-layer weight updates on the selected layers.

4 Proposed Method

Numerous studies have demonstrated that knowledge in models is stored in a key-value pair format, where $m_t^{(l)}$ in equation 3 is regarded as the value v^l (Meng et al., 2023a; Gupta et al., 2023; Fang et al., 2025). Then it can be simplified as:

$$v^l = W_{proj}^{(l)} \cdot k^l, \tag{7}$$

where $k^l = \sigma(W_{fc}^{(l)} \gamma(a_t^{(l)} + h_t^{(l-1)}))$. Therefore, once the layer that needs to be modified is determined, knowledge editing can be achieved by modifying W_{proj}^l.

Our proposed method, DMKE , is inspired by the causal mediation hypothesis in mechanistic interpretability. We hypothesize that a specific piece of factual knowledge corresponds to a causal pathway passing through a consecutive set of MLP modules. To locate this pathway, we design an algorithm based on a sliding-window traversal mechanism.

We view the forward pass as a causal chain of L layers. To locate the optimal intervention region, DMKE applies a sliding-window traversal over consecutive layers with window size w and stride s (e.g., in Figure 2, $w = 3$, $s = 2$), producing grouped layer sets $\Delta_1, \Delta_2, \ldots$.

For each grouped layer set $\Delta_i (i = 1, 2...)$, DMKE computes a layer-group editability score that quantifies the causal impact of intervening on that window. The algorithm evaluates the following objective and selects the MLP group corresponding to the minimum score as the optimal editing layer set:

$$\sum_{l=l^i_{\min}}^{l^i_{\max}} \frac{1}{l^i_{\max} - l + 1} \left\| \frac{\delta_{l^i_{max}}}{\left\| \mathbf{W}_{\text{proj}}^{(l)} \right\|_2 \cdot \sigma\left(\mathbf{x} \mathbf{W}_{\text{fc}}^{(l)} \right)^2} \right\|_2 . \tag{8}$$

Intuitively, this scoring function serves as a causal discount factor: edits applied closer to the $l^i_{\max}$ are penalized more heavily (since $l^i_{\max} - l + 1$ is smaller), as layers with greater causal distance

exert weaker direct influence on the final prediction. Consequently, the window that minimizes this score represents the path of least resistance for knowledge integration—i.e., the most efficient intervention zone along the causal chain, which DMKE identifies as the target editing layers.

In the above formula, $l^i_{\min} = min(\Delta_i), l^i_{\max} = max(\Delta_i), \delta_{l^i_{max}} = v^*_{l^i_{max}} - W^{(l^i_{max})}_{proj} \cdot k^{l^i_{max}}$

where $\delta_{l^i_{max}}$ can be solved by optimizing the following problem using gradient descent:

$$\underset{\delta_{l^i_{\max}}}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} - \log P_{f_\theta\left(h^{l^i_{\max}}_j \leftarrow h^{l^i_{\max}}_j + \delta_{l^i_{\max}}\right)} \left[y_j \mid t_n \oplus x_j\right]. \tag{9}$$

In summary, the optimization objective of DMKE is to maximize the model's prediction of the desired output y_j when presented with a new knowledge item x_j along with a set of random prefixes t_n. These random prefixes are concatenated with the new knowledge to enhance generalization. The objective is achieved by modifying the model's forward pass: the original hidden state $h^{l^i_{\max}}_j$ is replaced with the updated hidden state $h^{l^i_{\max}}_j + \delta_{l^i_{\max}}$, which corresponds to the optimal value $v^*_{l^i_{max}}$ introduced earlier,Further details can be referenced in Meng et al. (2023b).

After selecting the most knowledge-relevant MLP subset Δ_*, DMKE computes the optimized edit $\delta_{l^i_{\max}}$(hereafter abbreviated $\delta_{l_{\max}}$) for its final layer. The projection-weight editing procedure then proceeds in two steps. First, for the selected MLP group, we compute the layer-wise residual vectors and the corresponding key values using the following equations, respectively:

$$\frac{\delta_{l_{\max}}}{l_{\max} - l + 1} \quad l \in [l_{\min}, l_{\max}], \tag{10}$$

$$k_l = \frac{1}{N} \sum_{n=1}^{N} k(t_n \oplus x_j) \quad l \in [l_{\min}, l_{\max}]. \tag{11}$$

Each key k_l is computed as the input to $W^{(l)}_{proj}$ at each layer l in equation 7. Second, each layer's projection weights are updated as:

$$\begin{aligned} R &= [\frac{\delta_{l_{\max}}}{l_{\max} - l_{\min} + 1} ... \delta_{l_{\max}}], \\ K &= [k_{l_{\min}} ... k_{l_{\max}}], \end{aligned} \tag{12}$$

$$\begin{aligned} W^{(l)}_{\text{proj}} += R^l {K^l}^T (C^l + K^l {K^l}^T)^{-1} \\ l \in [l_{\min}, l_{\max}], \end{aligned} \tag{13}$$

where C^l denotes the uncentered covariance statistic estimated from the empirical sample of input vectors at layer l. For further computational details regarding the above update rule, we refer the reader to Meng et al. (2023b). Algorithm 1 presents a comprehensive summary of the proposed DMKE and Figure 2 illustrates the complete workflow.

5 EXPERIMENTS

We evaluate DMKE on GPT-2 XL and GPT-J-6B against four representative baselines: MEND Mitchell et al. (2022a), which applies a low-rank decomposition to the gradient derived from standard fine-tuning, making its parameterization tractable; ROME Meng et al. (2023a), which modifies feed-forward weights via rank-one updates based on the hypothesis that feed-forward computations correspond to factual association recall; MEMIT Meng et al. (2023b), a scalable multi-layer update algorithm that inserts new memories through explicitly calculated parameter updates; and WilKE

Algorithm 1 DMKE Algorithm

Require: Editing examples $\mathcal{E} = \{(x_{ei}, y_{ei})\}$, autoregressive language model f, upper limit layer L to edit, window size w, sliding stride s
Ensure: Modified model f_θ containing edits
1: Group the MLPs of the first L layers of the model f using w and s
2: **for** each group Δ_i (where $i = 1, 2, \dots$) **do**
3: Compute $\delta_{l^i_{\max}}$ for $l^i_{\max}$ ▷ Eq. 9
4: Compute the layer-group editability score ▷ Eq. 8
5: **end for**
6: Choose layer editing group Δ_*
7: **for** $l \in \Delta_*$ **do**
8: Compute residual vectors ▷ Eq. 10
9: Compute k_l ▷ Eq. 11
10: **end for**
11: Organize $\mathbf{R}$ and $\mathbf{K}$ vectors ▷ Eq. 12
12: **for** $l \in \Delta_*$ **do**
13: Edit layer's projection weights ▷ Eq. 13
14: **end for**

Hu et al. (2024), a lifelong knowledge editing algorithm that selects the editing layer based on the degree of pattern matching for different editing knowledge.

We employ the CounterFact dataset, one of the most widely used benchmarks in knowledge editing. To ensure a fair comparison, all methods (including ours) follow WilkeHu et al. (2024) in the selection of both edit and evaluation data.

5.1 Long-term Editing Effectiveness of the Model

To evaluate the lifelong knowledge editing, we performed 1,024 knowledge edits using our proposed algorithm, DMKE, as well as all baseline methods. The parameter settings are provided in the appendix A.

Table 1: Performance comparison on CounterFact after 1024 edits on GPT-2XL and GPT-J-6B. The best results per model are in bold. The hyperparameters of DMKE are set to $(w = 5, s = 3)$ for GPT-2XL and $(w = 4, s = 2)$ for GPT-J-6B.

Model	Editor	ES(%)	GS(%)	LS(%)	ERS(%)	ORS(%)	S(%)
GPT-2XL	MEND	0.6	0.1	0.5	0.0	0.0	0.0
	ROME	16.2	9.0	6.2	11.5	7.1	8.6
	MEMIT	**91.0**	**54.3**	**34.5**	34.2	25.9	32.4
	WilKE	60.3	45.0	9.8	8.3	7.1	11.8
	DMKE	82.8	45.9	30.6	**63.7**	**54.4**	**49.5**
GPT-J-6B	MEND	3.5	1.3	3.2	9.9	8.6	3.1
	ROME	40.0	31.6	9.9	25.7	10.0	16.5
	MEMIT	37.0	27.1	11.4	0.0	0.0	0.0
	WilKE	72.3	55.7	8.5	16.9	8.2	14.7
	DMKE	**98.7**	**68.1**	**31.2**	**80.6**	**60.0**	**57.6**

Following prior work, we adopt five widely used metrics to evaluate editing performance: Edit Success (ES), Generalization Success (GS), Locality Success (LS), Edit Reversion Success (ERS), and Original Retention Success (ORS). ES measures the success of a knowledge edit; GS measures the generalization ability of the edited model on related knowledge; LS measures the locality of an edit; ERS and ORS evaluate robustness under rollback editing—ERS measures whether the model restores the original knowledge after rollback, while ORS measures whether the model preserves unedited original knowledge after rollback. The harmonic mean of these five metrics, S, provides an

overall performance score. A detailed description of the edit-rollback process and the definitions of the metric calculation formulas are provided in Appendix B.

For each algorithm, we perform the editing process three times, and the results in the Table 1 represent the mean of the three runs. On the GPT-2 XL model, the performance of each algorithm exhibits significant differences. Specifically, MEMIT leads in basic editing capability, achieving an ES of 91.0% and a GS of 54.3%, both significantly higher than those of other methods, demonstrating its strong direct knowledge injection ability. DMKE excels in knowledge stability and relevance preservation. Although its ES (82.8%) and GS (45.9%) are slightly lower than those of MEMIT, it substantially outperforms MEMIT in both ERS (63.7%) and ORS (54.4%). This indicates that DMKE better maintains the original semantic associations and structural consistency of the model while modifying knowledge. The composite metric S (49.5%) further confirms the advantage of DMKE, as it effectively balances editing effectiveness with model integrity, avoiding the relevance loss issue observed with MEMIT's high edit success rate. In contrast, MEND is almost entirely ineffective, with all metrics approaching zero. ROME and WilKE, while exhibiting some editing capability, perform only modestly in terms of stability and relevance preservation.

On the larger-scale GPT-J-6B model, the performance trends of the algorithms change significantly. DMKE demonstrates strong editing robustness, achieving the highest ES (98.7%) and GS (68.1%), with an LS (31.2%) significantly outperforming other methods, proving that it can maintain high precision and strong generalization capability even on large models. MEMIT exhibits severe adaptability limitations: although it performed excellently on GPT-2 XL, its ERS and ORS on GPT-J-6B drop sharply to 0.0, indicating that this method severely disrupts the original knowledge structure when editing large models. ROME performs best in baselines, but its S(16.5%) remains far below that of DMKE (57.6%). WilKE achieves a relatively high ES (72.3%), but shows mediocre performance in LS (8.5%) and S (14.7%), exhibiting a "high editing volume, low consistency" issue.

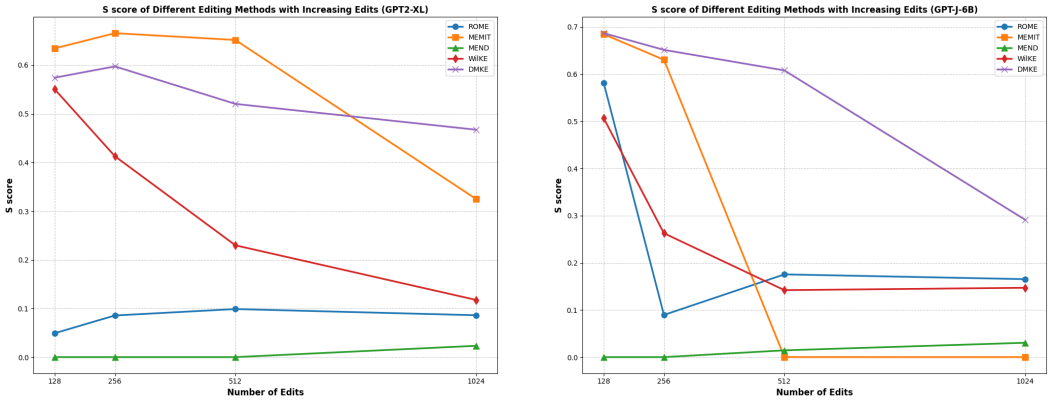
In summary, DMKE significantly outperforms other algorithms in long-term editing effectiveness, particularly on GPT-J-6B. Compared to WilKE, DMKE achieves improvements of 26.4% in ES, 12.4% in GS, 22.7% in LS, 63.7% in ERS, and 51.8% in ORS, resulting in an overall performance improvement of 42.9%. Compared to MEMIT, DMKE shows approximately 10% lower ES and GS on GPT-2 XL, but its ERS and ORS are about 25% higher. Overall, DMKE delivers better performance. Evidently, compared to baselines, the DMKE algorithm exhibits more stable performance.

5.2 Evaluating Editing Stability

To evaluate the stability of model editing across different scales, we sequentially apply 128, 256, 512, and 1024 edits to the target model, and systematically tracked and compared the performance trajectories of various methods, including MEND, ROME, MEMIT, WilKE, and DMKE. The resulting line charts intuitively illustrate the relationship between the number of edits and the overall performance metric S, clearly revealing how editing effectiveness degrades or remains robust as the scale increases, thereby quantifying the long-term stability of each method. To establish a comparative baseline with MEMIT and WilKE, both the window size w and sliding stride s of DMKE were set to 5 for GPT-2 XL, and to 6 for GPT-J-6B.

Figure 3a and 3b present the editing effects on GPT-2 XL and GPT-J-6B, respectively. Overall, as the number of edits increases from 128 to 1024, DMKE consistently demonstrates significant advantages in stability. Specifically, in Figure 3a, DMKE exhibits the smoothest performance decline curve, with a trend close to the ideal gradual forgetting process. In contrast, WilKE shows severe performance degradation, while MEMIT also experiences noticeable performance decay in long-term editing scenarios (exceeding 512 edits). In Figure 3b, DMKE's advantage in maintaining performance stability becomes even more pronounced, with a performance drop significantly smaller than that of other baseline methods. Furthermore, DMKE demonstrates good adaptability across models of different parameter scales. On the 1.5B-parameter GPT-2 XL, its performance remains stable between 0.52 and 0.60 across the entire editing range. On the 6B-parameter GPT-J-6B, its performance ranges from 0.29 to 0.69, significantly outperforming other algorithms. Even under an extremely high number of edits, DMKE maintains relatively stable performance. This advantage may be attributed to its improved editing localization mechanism: compared to traditional methods, DMKE can more precisely locate the subset of parameters that need to be modified, reducing interference with unrelated parameters and effectively mitigating the risk of parameter conflicts.

5.3 Ablation Study of DMKE



(a) GPT-2XL ($w = 5, s = 5$) (b) GPT-J-6B ($w = 6, s = 6$)

Figure 3: Comparison of the S score decay curves for each algorithm on GPT-2XL and GPT-J-6B.

Ablation experiments were carried out to examine the impact of varying the sliding stride and window size in DMKE.

Table 2: Performance comparison of different stride parameters on 1024 counterfactual edits. ↑ indicates higher is better. Best results per model are in bold.

Model	Stride	ES(%)↑	GS(%)↑	LS(%)↑	ERS↑	ORS(%)↑	S(%)↑
GPT-2XL ($w = 5$)	1	57.4	26.7	**35.7**	**73.5**	**62.8**	43.0
	3	**82.8**	**45.9**	30.6	63.7	54.4	**49.5**
	5	78.4	40.2	29.5	61.1	52.6	46.7
GPT-J-6B ($w = 6$)	2	**98.1**	**71.9**	**31.2**	**42.1**	**27.4**	**40.8**
	4	86.4	62.8	28.4	24.3	16.2	25.2
	6	97.2	70.4	29.2	25.0	15.4	29.1

Table 3: Performance comparison of different window sizes on 1024 counterfactual edits. ↑ indicates higher is better. Best results per model are in bold.

Model	Window	ES(%)↑	GS(%)↑	LS(%)↑	ERS↑	ORS(%)↑	S(%)↑
GPT-2XL ($s = 3$)	3	71.9	33.2	**36.1**	**67.4**	**58.9**	48.2
	5	**82.8**	**45.9**	30.6	63.7	54.4	**49.5**
	7	77.9	45.0	22.2	22.6	18.0	22.6
	9	80.0	43.3	26.0	41.5	32.3	38.6
GPT-J-6B ($s = 2$)	2	**98.7**	70.0	**34.8**	56.4	31.1	48.1
	4	**98.7**	68.1	31.2	**80.6**	**60.0**	**57.6**
	6	98.1	**71.9**	31.2	42.1	27.4	40.8
	8	**98.7**	69.0	32.0	78.2	54.8	57.5

Sliding stride We evaluate the effect of sliding stride on DMKE's performance in Table 2. For GPT-2 XL, with window size fixed at $w = 5$, we vary the stride over $s \in \{1, 3, 5\}$. DMKE achieves its best performance at $s = 3$, attaining a total score of 49.5% and substantially outperforming MEMIT (32.4%). For GPT-J-6B, with $w = 6$, we vary the stride over $s \in \{2, 4, 6\}$. The best performance is achieved at $s = 2$, with a total score of 40.8%, significantly surpassing WilKE (14.7%).

Window Size We evaluate the influence of window size on DMKE's performance in Table 3. For GPT-2 XL, varying window size $w \in \{3, 5, 7, 9\}$ at a fixed stride of $s = 3$, DMKE achieves its best score (49.5%) at $w = 5$, outperforming MEMIT (32.4%). For GPT-J-6B, varying window size $w \in \{2, 4, 6, 8\}$ at a fixed stride of $s = 2$, DMKE achieves its best score (57.6%) at $w = 4$, outperforming WilKE (14.7%).

The ablation study reveals that DMKE achieves optimal performance when the window size is approximately twice the sliding stride.

5.4 Time Cost for Long-term Model Editing

Among the existing long-term knowledge editing approaches, namely WilKE and DMKE, we conduct a computational complexity analysis by measuring their execution times. For DMKE, we set the window size w and sliding stride s to 5 for GPT-2 XL.

As illustrated in Figure 4, when performing 1,024 sequential edits on GPT-2 XL, DMKE completes the process in 3.2 hours, whereas WilKE requires 8.4 hours—a reduction of 5.2 hours (approximately 62%). In contrast, when editing GPT-J-6B, the execution times of the two methods are comparable. These results demonstrate that DMKE not only enhances editing efficacy but also significantly reduces computational overhead.

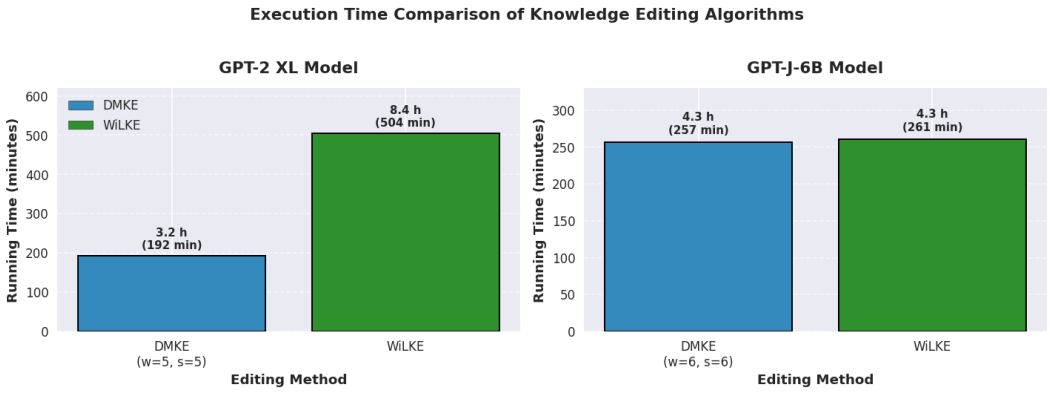


Figure 4: Comparative analysis of time costs for scaling 1024 edits with DMKE and WilKE on GPT-2XL and GPT-J-6B.

6 Conclusion

In this paper, we present DMKE, a dynamic multi-layer knowledge editing algorithm that enables precise, efficient, and stable updates to large language models. Evaluated on the CounterFact, DMKE substantially outperforms state-of-the-art methods. Its core innovations—a dynamic grouping mechanism for adaptive layer selection and a gradient-based relevance scoring module for precise localization—enable robust performance in long-term editing tasks. By systematically analyzing the impact of parameter configurations, we demonstrate that DMKE exhibits significantly less performance degradation than baselines, particularly in large-scale sequential editing. Consequently, DMKE provides a robust, scalable solution for real-time knowledge updates, effectively bridging precise editing with long-term stability and paving the way for advanced dynamic knowledge maintenance.

Limitations

DMKE achieves promising results but has limitations. Optimal window-stride configurations are model-dependent and require task-specific tuning. Evaluation is limited to English benchmarks and Transformer-based models; future work should explore multilingual settings and other architectures. Addressing these limitations remains an important direction for future work.

AI USE STATEMENT

Throughout the paper, the use of LLMs is solely restricted to the polishing of textual elements, such as lexical or phrasal substitutions, and does not extend beyond this scope.

REFERENCES

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model, 2022. URL https://arxiv.org/abs/2204.06745.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models, 2021. URL https://arxiv.org/abs/2104.08164.

Qizhou Chen, Taolin Zhang, Xiaofeng He, Dongyang Li, Chengyu Wang, Longtao Huang, and Hui Xue. Lifelong knowledge editing for llms with retrieval-augmented continuous prompt learning, 2024. URL https://arxiv.org/abs/2405.03279.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Damai Dai, Wenbin Jiang, Qingxiu Dong, Yajuan Lyu, and Zhifang Sui. Neural knowledge bank for pretrained transformers. In Fei Liu, Nan Duan, Qingting Xu, and Yu Hong (eds.), *Natural Language Processing and Chinese Computing*, pp. 772–783, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-44696-2.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Shi Jie, Xiang Wang, Xiangnan He, and Tat seng Chua. Alphaedit: Null-space constrained knowledge editing for language models, 2025. URL https://arxiv.org/abs/2410.02355.

Anshita Gupta, Debanjan Mondal, Akshay Sheshadri, Wenlong Zhao, Xiang Li, Sarah Wiegreffe, and Niket Tandon. Editing common sense in transformers. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8214–8232, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.511. URL https://aclanthology.org/2023.emnlp-main.511/.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47934–47959. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/95b6e2ff961580e03c0a662a63a71812-Paper-Conference.pdf.

Chenhui Hu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Wilke: Wise-layer knowledge editor for lifelong knowledge editing, 2024. URL https://arxiv.org/abs/2402.10987.

Yuxin Jiang, Yufei Wang, Chuhan Wu, Wanjun Zhong, Xingshan Zeng, Jiahui Gao, Liangyou Li, Xin Jiang, Lifeng Shang, Ruiming Tang, Qun Liu, and Wei Wang. Learning to edit: Aligning LLMs with knowledge editing. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4689–4705, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.258. URL https://aclanthology.org/2024.acl-long.258/.

Amruta Kale, Tin Chi Nguyen, Fred Harris, Chenhao Li, Jiyin Zhang, and Xiaogang Ma. Provenance documentation to enable explainable and trustworthy ai: A literature review. *Data Intelligence*, 5:139–162, 2022. URL https://api.semanticscholar.org/CorpusID:246886309.

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, 2025. URL https://arxiv.org/abs/2308.08747.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023a. URL https://arxiv.org/abs/2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer, 2023b. URL https://arxiv.org/abs/2210.07229.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale, 2022a. URL https://arxiv.org/abs/2110.11309.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15817–15831. PMLR, 17–23 Jul 2022b. URL https://proceedings.mlr.press/v162/mitchell22a.html.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel

Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training, 2021. URL https://arxiv.org/abs/2104.10350.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks, 2020. URL https://arxiv.org/abs/2004.00345.

Chenmien Tan, Ge Zhang, and Jie Fu. Massive editing for large language models via meta learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=L6L1CJQ2PE.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 38274–38290. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/fa0509f4dab6807e2cb465715bf2d249-Paper-Conference.pdf.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. Memoryllm: Towards self-updatable large language models, 2024. URL https://arxiv.org/abs/2402.04624.

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 Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.632. URL https://aclanthology.org/2023.emnlp-main.632/.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics*, 9:362–373, 2021. doi: 10.1162/tacl_a_00371. URL https://aclanthology.org/2021.tacl-1.22/.

Ningyu Zhang, Bozhong Tian, Siyuan Cheng, Xiaozhuan Liang, Yi Hu, Kouying Xue, Yanjie Gou, Xi Chen, and Huajun Chen. Instructedit: Instruction-based knowledge editing for large language models, 2024. URL https://arxiv.org/abs/2402.16123.

Zihan Zhang, Meng Fang, Ling Chen, Mohammad-Reza Namazi-Rad, and Jun Wang. How do large language models capture the ever-changing world knowledge? a review of recent advances. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8289–8311, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.516. URL https://aclanthology.org/2023.emnlp-main.516/.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Yiwen Hu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *Frontiers of Computer Science*, 20(12), May 2026. ISSN 2095-2236. doi: 10.1007/s11704-026-60308-3. URL http://dx.doi.org/10.1007/s11704-026-60308-3.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models, 2020. URL https://arxiv.org/abs/2012.00363.

A Parameter Settings of Long-term Editing Effectiveness

For ROME, we set the editing layer to layer 17 for GPT-2 XL and layer 5 for GPT-J-6B. For MEMIT, we applied layers 13–17 for GPT-2 XL and layers 3–8 for GPT-J-6B. For WilKE, we traversed the first 24 layers of GPT-2 XL and the first 7 layers of GPT-J-6B. The maximum layer for DMKE was kept consistent with WilKE.

B Long-Term Editing Process and Metric Calculation

This appendix describes the long-term editing process of DMKE and provides formal definitions of the evaluation metrics.

B.1 The Edit-Rollback Process of DMKE

The long-term editing process of DMKE alternates between edit and rollback operations, as shown in Figure 1. Let P denote the sample set. The i-th edit knowledge$(x_e^i, y_e^i) \in P$, equivalence neighborhood samples $N(x_e^i)$, unrelated to the edited knowledge $(x^i, y^i) \in E_{\text{loc}}^i$ correspond to the red, purple, and brown texts in Figure 1, respectively. Samples outside the edit set constitute a set Q satisfying $P \cap Q = \emptyset$. Q consists of factual triplets, as shown by the green samples and their answers in Figure 1. Specifically, one edit-rollback cycle proceeds as follows: Use (x_e^i, y_e^i) to edit the model

$f^*_{\theta_{i-1}}$ and obtain f_{θ_i}. For the edited model, use original knowledge (x^i_e, y^i_o) to edit f_{θ_i} and obtain $f^*_{\theta_i}$. Here is a simple example: In Figure 1, y^i_e and y^i_o represent Trump and Biden, respectively.

ES, GS, and LS are edit-phase metrics, whereas ERS and ORS are rollback-phase metrics. In particular, ES and ERS belong to the edit and rollback phases, respectively; despite operating in different phases, they jointly characterize the success rate of editing. We now describe each metric and its computation in detail.

B.2 ES: Edit Success

ES measures the success of a knowledge edit and is computed as follows:

$$ES = \frac{1}{|P|} \sum_{i=1}^{|P|} \mathbb{I}\big(f^*_{\theta_i}(x^i_e) = y^i_e\big). \tag{14}$$

B.3 GS: Generalization Success

GS measures the generalization ability of the edited model on related knowledge, i.e., whether the model can still produce the new knowledge under different paraphrases of the same fact.Its calculation is given as follows:

$$GS = \frac{1}{|P|} \sum_{i=1}^{|P|} \frac{1}{|N(x^i_e)|} \sum_{j=1}^{|N(x^i_e)|} \mathbb{I}\big(f^*_{\theta_i}(x^j) = y^i_e\big), x^j \in N(x^i_e). \tag{15}$$

B.4 LS: Locality Succes

LS measures the locality of an edit, i.e., whether the edited model's performance on unrelated knowledge is affected, and is computed as follows:

$$LS = \frac{1}{|P|} \sum_{i=1}^{|P|} \frac{1}{|E^i_{\text{loc}}|} \sum_{j=1}^{|E^i_{\text{loc}}|} \mathbb{I}\big(f^*_{\theta_i}(x^j) = y^i\big), (x^i, y^i) \in E^i_{\text{loc}}. \tag{16}$$

B.5 ERS: Edit Reversion Success

ERS stands for Edit Reversion Success, which measures whether the model still remembers all the original knowledge after the "edit-rollback" process. This metric is designed to test whether the editing algorithm has truly altered the model's memory of that knowledge, or merely temporarily masked it. Its calculation is given as follows.

$$ERS = \frac{1}{|P|} \sum_{i=1}^{|P|} \mathbb{I}\big(f^*_{\theta_i}(x^i_e) = y^i_o\big) \tag{17}$$

B.6 ORS: Original Retention Success

ORS which measures whether the model still remembers those original knowledge items that have never been edited after multiple edit-rollback cycles. It is evaluated on knowledge that has never been edited, and when computing this metric, we use the data outside the editing samples. Its calculation is given as follows:

$$ORS = \frac{1}{|Q|} \sum_{i=|P|+1}^{|P|+|Q|} \mathbb{I}\big(f^*_{\theta_i}(x^i) = y^i\big), (x^i, y^i) \in Q. \tag{18}$$