The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)

Beyond Superficial Forgetting: Thorough Unlearning Through Knowledge Density Estimation and Block Re-insertion

Feng Guo[1*], Yuntao Wen[1*], Shen Gao[1†], Junshuo Zhang[1], Shuo Shang[1,2]
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]State Key Laboratory of AI Safety, Beijing, 100086
zxyz0051@163.com, shengao@uestc.edu.cn, {yuntaowenx, hanblingz026, jedi.shang}@gmail.com

Abstract

Machine unlearning, which selectively removes harmful knowledge from a pre-trained model without retraining from scratch, is crucial for addressing privacy, regulatory compliance, and ethical concerns in Large Language Models (LLMs). However, existing unlearning methods often struggle to thoroughly remove harmful knowledge, leaving residual harmful knowledge that can be easily recovered. To address these limitations, we propose **K**nowledge Density-Guided **Un**learning via **B**locks **R**einsertion (KUnBR), a novel approach that first identifies layers with rich harmful knowledge and then thoroughly eliminates the harmful knowledge via re-insertion strategy. Our method introduces knowledge density estimation to quantify and locate layers containing the most harmful knowledge, enabling precise unlearning. Additionally, we design a layer re-insertion strategy that extracts and re-inserts harmful knowledge-rich layers into the original LLM, bypassing gradient obstruction caused by cover layers and ensuring effective gradient propagation during unlearning. Extensive experiments conducted on several unlearning and general capability benchmarks demonstrate that KUnBR achieves state-of-the-art forgetting performance while maintaining model utility.

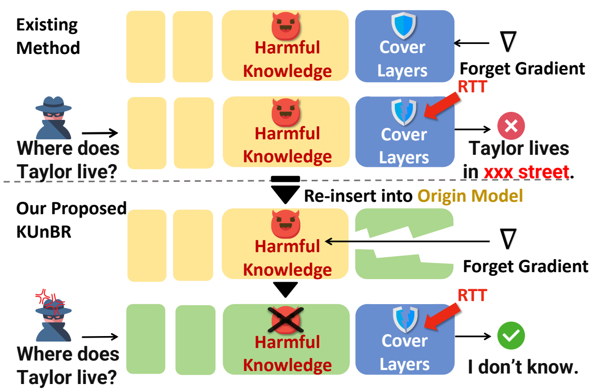

Figure 1: Existing unlearning methods fail to thoroughly remove harmful knowledge due to the presence of cover layers. They do not output harmful knowledge simply because the **cover layer filters out the harmful content**, but this knowledge **still resides in the parameters**. Our KUnBR achieves better unlearning by reinserting layers with high knowledge density into the original model, thereby disrupting the cover layers.

Introduction

Machine unlearning (Liu et al. 2025; Bourtoule et al. 2021a) refers to selectively removing specific subsets of knowledge, such as privacy-sensitive or harmful content, from a pre-trained model without retraining it from scratch (Carlini et al. 2021; Xu et al. 2024). This task has become increasingly crucial for the development of large language models (LLMs) (OpenAI 2024; AI@Meta 2024; Anthropic 2024; Guo et al. 2025), as it addresses growing concerns around data privacy (Carlini et al. 2021; Huang, Shao, and Chang 2022; Lee et al. 2024; Liu et al. 2024) and the ethical issue of AI systems (Bender et al. 2021). Unlearning is critical not only for addressing regulatory requirements such as "right to be forgotten", but also for ensuring that LLMs remain secure, reliable, and aligned with societal values.

Previous research has explored different unlearning methodologies, such as gradient ascent (Jang et al. 2022; Eldan and Russinovich 2024), which unlearn the knowledge by increasing the loss when outputting harmful answers. These methods always utilize two distinct datasets as guidance to optimize the model: a *forget set*, which contains the information to be removed, and a *retain set*, which preserves the model's general knowledge and performance on unrelated tasks (Bourtoule et al. 2021b). These methods can adjust the final output of LLM to suppress harmful outputs.

Although existing machine unlearning methods can suppress harmful knowledge, several jailbreak attack studies (Zhou et al. 2024; Liu et al. 2023; Schwinn et al. 2024; Rimsky et al. 2024) have shown that the robustness issues remain. The **Re**training on **T** (RTT) (Deeb and Roger 2025), which is an attack method at the parameter-modification level, demonstrates that minimal re-training on a subset (*a.k.a.,* the **T** set) of the forget set can restore most of the supposedly eliminated knowledge. These results demonstrate that the model parameters still contain a substantial amount of knowledge that should have been forgotten, which reveals the inability of existing methods to thor-

*These authors contributed equally.
†Corresponding Author

oughly remove knowledge from the model parameters. That means that existing methods often rely on the adjustment of a small number of model parameters (*a.k.a.,* **cover layers**) to mask or suppress the representation of harmful knowledge, merely preventing the model from outputting undesired content without truly eliminating it from the model's internal representations. This fundamental limitation suggests the need for more robust and thorough unlearning methods.

In this paper, we propose **K**nowledge Density-Guided **Un**learning via **B**locks **R**einsertion (KUnBR), which identifies blocks with rich harmful knowledge, and iteratively performs independent unlearning on these blocks via re-insertion strategy, enables a deeper level of unlearning[1]. We first introduce a **knowledge density estimation** method to identify the layers that contain the most harmful knowledge. By calculating the absolute value of gradients associated with the forget set, knowledge density estimation can locate layers containing high-density knowledge. Instead of superficially modifying cover layer parameters to suppress model outputs, we propose a novel re-insertion strategy to thoroughly unlearn undesired knowledge. This approach extracts knowledge-rich blocks (selected according to the knowledge density estimation) from the unlearned LLM and re-inserts them into the original LLM without conducting the unlearning training. We then apply the unlearning method to train this "grafted" model, which contains the re-inserted layers, with a focus on deeper removal of the undesired knowledge left due to the influence of cover layers. By bypassing the obstruction of cover layers, this strategy ensures more effective gradient propagation and enhances the model's ability to forget. This simple but efficient strategy significantly reduces the vulnerability of the model to attacks like RTT, which exploit the residual knowledge left by conventional unlearning methods. Extensive experiments conducted on WMDP-Deduped, Years, Random Birthdays and RKWU benchmark datasets demonstrate that our method achieves state-of-the-art performance, since it can remove harmful knowledge more thoroughly and suppress knowledge recovery caused by RTT attack methods. Our contributions are summarized as follows:

- **We propose Knowledge Density-Guided Unlearning via Blocks Reinsertion (KUnBR)**, a novel unlearning framework that identifies layers containing undesired knowledge and performs targeted training to achieve thorough elimination of harmful knowledge.
- **We introduce a knowledge density estimation method**, which can identify layers with more harmful knowledge in LLMs for more effective unlearning.
- **We propose a novel re-insertion strategy** to ensure unlearning gradients propagate effectively, overcoming the limitations of gradient obstruction.
- **Extensive experiments** demonstrate that KUnBR achieves state-of-the-art forgetting performance across multiple benchmark datasets, keeping the general ability of LLM.

[1]Code available at github.com/llmgfffffff/Beyond-Superficial-Forgetting-KUnBR

Related Work

With the rapid development of LLMs, the importance of unlearning tasks has become increasingly prominent. During the pre-training process, where these models ingest massive amounts of information, they may incorporate harmful content (Carlini et al. 2021; Yao et al. 2024), sensitive data, or copyrighted materials (Ren et al. 2024; Dou et al. 2024). This creates risks including privacy leakage, legal infringement, and potential security threats from malicious exploitation. In recent years, several unlearning methods have emerged to ensure effective removal of undesirable information while maintaining model performance on legitimate tasks, such as Representation Misdirection for Unlearning (Li et al. 2024) (`RMU`) employs a dual loss function combining forgetting loss and retention loss, selectively adjusting intermediate layers to erase harmful knowledge. Gradient Ascent (Jang et al. 2022) (`GA`) applies gradient ascent on forget set. Building upon `DPO` (Wang et al. 2024), Negative Preference Optimization (Zhang et al. 2024) introduces negative preference optimization to address `GA`'s collapse problem. It achieves a better balance between unlearning quality and model utility, particularly effective in high-ratio forgetting scenarios (*e.g.,* $>$50% in the TOFU data set (Maini et al. 2024)) while maintaining usability. Gradient Differentiation (Liu, Liu, and Stone 2022) applies differentiated gradient operations on forgetting and retaining sets.

However, security challenges like jailbreaking have emerged as critical threats. Attackers can exploit model sensitivity through: (1) Contextually obscure prompts inducing information leakage (Liu et al. 2023), (2) Backdoor triggers embedded during training (Liu et al. 2022), (3) Adversarial examples disrupting unlearning mechanisms (Deeb and Roger 2025). Similarly, the RTT method proposed by Deeb and Roger (2025) reveals that fine-tuning on partially forgotten data can recover supposedly eliminated knowledge, exposing residual information retention in "unlearned" models. This suggests that current unlearning methods face significant limitations: existing approaches are merely a superficial form of forgetting, with harmful or intended-to-remove knowledge still remaining in various parts of the model. Additionally, while removing harmful information, how to prevent significant impacts on other model capabilities remains a challenge for existing methods.

Problem Definition

Given the forget data set D_{forget}, containing knowledge to be removed, and the retain data set D_{retain}, which helps the model maintain general ability during unlearning. The model parameters should be optimized to eliminate forgotten knowledge associated with D_{forget} as thoroughly as possible, while ensuring that the utility performance of the model remains unaffected. Furthermore, when subjected to a fine-tuning (RTT) attack–where the model is fine-tuned on a subset T partitioned from D_{forget}–it remains incapable of generating knowledge contained in another disjoint subset V of D_{forget}. This demonstrates the effectiveness and robustness of its unlearning.

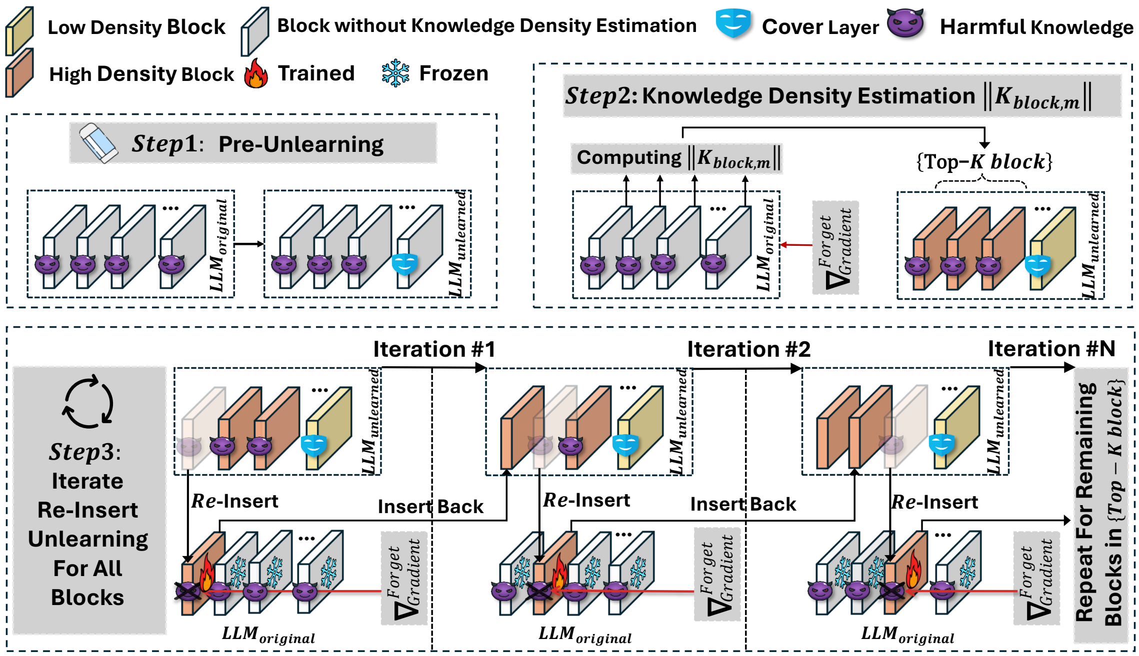

Figure 2: Architecture of our proposed **K**nowledge Density-Guided **Un**learning via **B**locks **R**einsertion (KUnBR).

KUnBR Methodology

Figure 2 shows the architecture of KUnBR. The first step is the global "warm-up" unlearning phase, in which we apply a standard Gradient Difference method to adjust all model parameters at once; In the second step, we perform knowledge density estimation and our block-selection strategy to pick out those blocks that contain high-density knowledge. Finally, we introduce a re-insertion strategy to bypass the masking effect of cover layers and enable any remaining knowledge to be further eliminated.

Influence of Cover Layer

Although existing methods (Li et al. 2024; Zhang et al. 2024; Liu, Liu, and Stone 2022; Jin et al. 2024) have achieved significant knowledge unlearning, recent studies (Hong et al. 2024) suggest that these methods modify only a small subset of layers during the unlearning. Thus, knowledge of D_{forget} is still retained in other layers, which explains why the forgotten knowledge can be easily recalled by retraining on T (RTT) attack (Deeb and Roger 2025). In this work, we refer to these modified layers as **cover layers** as they suppress the representation of the target knowledge.

Knowledge Density Estimation

To determine which layers' parameters require greater adjustment during unlearning (or are more likely to contain knowledge), it is crucial to develop a metric that accurately quantifies the knowledge density across different layers of the model. (Geva et al. 2021) demonstrated that the multilayer perceptron (MLP) components within LLMs serve as neural memory units. Other studies (Hong et al. 2024) have demonstrated that during unlearning, it is primarily the MLP layers that are modified and play a critical role. These findings indicate that the adjustment of knowledge in LLMs essentially involves fine-grained alterations to the neural storage units within the MLPs. Based on this insight, when optimizing a "forget set", the absolute value of the parameter gradients of each layer provides an intuitive measure of the amount of target knowledge it contains. In other words, larger gradient magnitudes imply that richer content is to be forgotten in that layer; accordingly, we adopt the absolute gradient value on the forget set as an effective metric for "knowledge density".

Motivated by this, we propose a gradient-guided knowledge density estimation metric, which is an indicator of knowledge density across layers associated with the forget set. Specifically, we first define the standard negative log-likelihood loss function for a given input x and target y with model parameters θ:

$$\mathcal{L}(x, y; \theta) = -\log(p(y|x; \theta)). \tag{1}$$

Given a forget set $D_{forget} = \{(x_i, y_i)\}_{i=1}^N$, where x_i represents an input question and y_i represents the corresponding answer that we want the model to forget, we can calculate the *knowledge density* K_l for each layer l of the LLM. This is done by taking the expectation over the forget set of the L_1 norm of the gradient of the loss with respect to the parameters θ_l of that specific layer:

$$K_l = \mathbb{E}_{(x,y)\sim D_{forget}} \left[\|\nabla_{\theta_l} \mathcal{L}(x, y; \theta_l)\|_1 \right], \tag{2}$$

where θ_l denotes the parameters of the l-th layer in the target LLM. A higher K_l suggests that the l-th layer's parameters are more sensitive to the information in the forget set.

To capture the relative importance of the l-th layer's knowledge density compared to other layers, we normalize

K_l by the total knowledge density across all H layers. The resulting K_l^{norm} represents the proportion of the total "forgettable" knowledge residing in the l-th layer:

$$K_l^{norm} = \frac{K_l}{\sum_{i=1}^{H} K_i}, \quad (3)$$

where H is the total number of layers in the target LLM.

Note that we compute these gradients solely on the forget set D_{forget} to derive the knowledge density metric. This metric indicates the degree to which the parameters within each layer need to be adjusted to facilitate the unlearning of the information contained in D_{forget}. Importantly, this entire step is solely for the calculation of the knowledge density of each layer; no parameter optimization or unlearning is performed at this stage.

Block Selection Strategy

Most LLMs are composed of a large number of stacked Transformer layers. Instead of treating each layer individually, we divide nearby layers into groups, which we refer to as "blocks", and treat each block as a basic unit for unlearning. This design simplifies the unlearning process and helps improve its overall efficiency.

Specifically, for an LLM containing H layers, we merge all layers into M blocks, with each block containing $N = \lfloor H/M \rfloor$ layers. Following this, we calculate the cumulative knowledge density of their constituent layers:

$$K_{\text{block},m} = \sum_{i=(m-1)N+1}^{mN} K_i^{\text{norm}}, \quad (4)$$

where $K_{\text{block},m}$ represents the m-th block's cumulative knowledge density, K_i^{norm} denotes the i-th layer's normalized knowledge density ($m = 1, 2, \ldots, M$). Next, we rank blocks by cumulative knowledge density and select them via the following two strategies.

Top-K Selection: We select top-K blocks with the highest knowledge density, where K is a hyperparameter. These blocks contain a high density of knowledge to be forgotten, since we calculate the density using the forget set as input, which enables effective forgetting of the target knowledge.

Ignoring the Head Layers: We observe a significant surge in the knowledge density values in the last two layers of the LLM. Based on empirical analysis of different layers (Hong et al. 2024), we hypothesize that this increase in knowledge density is not due to a higher concentration of knowledge in these layers, but rather a potential artifact caused by their involvement in the model's output generation. Consequently, during the unlearning process, we exclude the blocks that contain these last two layers to avoid unwanted interference.

Next, we will enhance the selected blocks during the unlearning process to ensure that these blocks with high knowledge density can more effectively forget the target knowledge. These two selection strategies enable efficient and maximal forgetting of harmful knowledge, while minimizing unintended damage to knowledge that should be retained, ensuring the efficiency and stability of the subsequent unlearning process.

Re-insertion Strategy For Unlearning

To mitigate the influence of the cover layer, we propose a *re-insertion strategy*. First, we identify harmful knowledge-rich blocks using our proposed block selection strategy. These blocks are then re-inserted into the original LLM that has not undergone unlearning, denoted as LLM$_{original}$.

To achieve this, we first apply a pre-unlearning process to LLM$_{original}$ to obtain LLM$_{unlearning}$. Specifically, we employ the standard Gradient Difference method (Liu, Liu, and Stone 2022) as the pre-unlearning step. We perform full-parameter fine-tuning during a warm-up phase to accelerate the overall convergence of unlearning.

Next, based on our block selection strategies, we identify harmful knowledge-rich blocks from LLM$_{unlearning}$. These blocks are then inserted into the corresponding positions in LLM$_{original}$, while the remaining layers are kept frozen. Subsequently, we apply Gradient Difference to this "grafted" LLM using D_{forget} and D_{retain}. Since the layers in LLM$_{original}$ remain unaltered and frozen, no cover layer is generated to interfere with the inserted block, enabling deeper removal of residual knowledge within the selected block. This allows us to eliminate residual knowledge from every selected block more deeply. Following the gradient difference process, the selected block in "grafted" LLM reverts to LLM$_{unlearning}$, resulting in significantly less residual knowledge compared to standard unlearning methods.

Experimental Setup

Datasets

We evaluate on four standard unlearning benchmarks: Random Birthdays (Deeb and Roger 2025), WMDP-Deduped (Li et al. 2024), Years (Penedo et al. 2024), and MMLU (Hendrycks et al. 2021).

Evaluation Metrics

Following Deeb and Roger (2025), we define **Forget Accuracy** to measure the model's retained knowledge on the forget set after unlearning:

$$\mathcal{A}_{\text{Unlearn}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I} \left(f_{\text{unlearn}}(x_i) = y_i \right), \quad (5)$$

where D_{forget} contains N multiple-choice questions (x_i, y_i), f_{unlearn} is the model after unlearning, and $\mathbb{I}(\cdot)$ returns 1 if the prediction matches y_i, else 0. At the same time, we use the same ACC calculation method in Formula 5 to measure the accuracy after the RTT attack (denoted as $\mathcal{A}_{\text{RTT}}$) and calculate the recovery rate before and after the RTT, as follows:

$$\mathcal{A}_{\text{Recover}} = \mathcal{A}_{\text{RTT}} - \mathcal{A}_{\text{Unlearn}}, \quad (6)$$

where the larger the $\mathcal{A}_{\text{Recover}}$, the worse the model's robustness in the face of attacks.

To verify whether the model's general capabilities are unexpectedly affected by our unlearning method, we adopt the utility evaluation framework proposed by the RKWU benchmark (Li et al. 2024). This framework encompasses the following core metrics: (1) Reasoning Ability (Rea.) is assessed on the Big-Bench-Hard (Suzgun et al. 2022)

Method	Backbone	Random Birthdays(99.3&96.1)			WMDP-Deduped(65.6&59.2)			Years(63.2&56.3)			MMLU(54.7&54.7)		
		Forget.↓	RTT.↓	Rec.↓	Forget.↓	RTT.↓	Rec.↓	Forget.↓	RTT.↓	Rec.↓	Forget.↓	RTT.↓	Rec.↓
GA	LLaMA3 8B	**23.5**	87.2	63.7	29.2	66.8	37.6	25.9	50.6	24.7	24.2	59.2	35.0
GD	LLaMA3 8B	64.9	80.2	15.3	30.5	62.4	31.9	25.9	68.3	42.4	35.0	57.6	22.6
RMU	LLaMA3 8B	36.3	88.5	52.2	29.9	64.9	35.0	24.2	68.3	44.1	24.8	49.0	24.2
RIA	LLaMA3 8B	61.7	73.8	12.1	**26.2**	52.2	26.0	**18.3**	65.8	47.5	26.7	48.4	21.7
NPO	LLaMA3 8B	71.3	78.3	**7.0**	35.6	58.4	22.8	26.5	67.7	41.2	31.2	38.8	**7.6**
KUnBR	LLaMA3 8B	36.9	**43.9**	**7.0**	29.2	**38.8**	**9.6**	25.9	**36.0**	**10.1**	**16.5**	**28.0**	11.5
GA	Zephyr 7B	—	—	—	35.0	57.3	22.3	25.3	56.3	31.0	24.2	47.1	22.9
GD	Zephyr 7B	32.5	46.5	14.0	26.8	52.2	25.5	26.0	58.9	32.9	34.4	48.4	14.0
RMU	Zephyr 7B	46.5	54.1	**7.64**	24.8	56.7	31.9	**20.9**	59.5	38.6	47.8	54.8	**7.01**
RIA	Zephyr 7B	35.7	47.5	11.8	**19.1**	47.8	28.7	24.1	58.9	34.8	35.7	47.8	12.1
NPO	Zephyr 7B	47.1	55.4	8.28	41.4	49.7	**8.28**	24.1	56.3	32.3	31.2	42.0	10.8
KUnBR	Zephyr 7B	**26.8**	**47.1**	20.4	23.6	**42.7**	19.1	25.3	**28.5**	**3.16**	**19.1**	**34.4**	15.3

Table 1: Performance comparison under RTT attack on two backbone LLMs. "Forget." ($\mathcal{A}_{\text{Unlearn}}$), "RTT." ($\mathcal{A}_{\text{RTT}}$), and "Rec." ($\mathcal{A}_{\text{Recover}}$) denote accuracy after unlearning, under RTT attack, and recovery accuracy, respectively. Parenthetical values next to each dataset name denote the original performance on LLaMA and Zephyr. **Bold** indicates the best result, underlined indicates the second best.

dataset through 3-shot chain-of-thought prompting, with Exact Match scores reported. (2) Truthfulness (Tru.) is measured on TruthfulQA's MC1 task (Lin, Hilton, and Evans 2022), reporting 6-shot accuracy. (3) Factuality (Fac.) is evaluated on the TriviaQA (Joshi et al. 2017) dataset using 6-shot prompting, with F1 scores reported. (4) Fluency (Flu.) is assessed using AlpacaEval's evaluation instructions (Dubois et al. 2023), reporting the weighted average of bi- and tri-gram entropies. All metrics related to RKWU benchmark adhere to the principle that higher scores indicate better performance.

Baselines

We employ several strong tuning-based unlearning approaches as the baselines: (1) **Gradient Ascent** (Jang et al. 2022) (`GA`): `GA` achieves unlearning by maximizing the loss on the forget set. (2) **Gradient Difference** (Liu, Liu, and Stone 2022) (`GD`): This approach performs gradient ascent on the forget dataset and gradient descent on the retain dataset. (3) **Representation Misdirection for Unlearning** (Li et al. 2024) (`RMU`): Given a harmful prompt, `RMU` performs unlearning by strategically modifying the internal representations (activations) within selected intermediate model layers. (4) **Random Incorrect Answer** (Deeb and Roger 2025) (`RIA`): For each multiple-choice question, `RIA` applies gradient descent to the incorrect choices, guiding the model to unlearn the correct choice associated with specific knowledge. (5) **Negative Preference Optimization** (Zhang et al. 2024) (`NPO`): `NPO` optimizes the model's preferences to exhibit a negative bias when handling tasks involving deleted information, thereby reducing the model's reliance on and memory of such information.

Implementation Details

We partition the datasets into forget and retain sets. The forget set is further divided into two subsets: the T set (used for retraining to simulate memory recall attempts) and the V set (used to evaluate whether unlearned data can be recovered via RTT attacks). We use the same split ratios for the D_{forget} / D_{retain} and the T / V subsets as Deeb and Roger (2025). All experiments are conducted on LLaMA3-8B-Instruct and Zephyr-7B-beta to evaluate the generalizability of our method across different backbone models. More implementation details can be found in Appendix.

Experimental Results

Overall Performance

Table 1 illustrates the forget accuracy of various unlearning methods, including `GA`, `GD`, `RIA`, `RMU`, `NPO`, and our proposed KUnBR. After conducting RTT attacks, most unlearning methods exhibit a significant increase in forget accuracy, indicating their vulnerability to RTT attacks and the potential recovery of forgotten knowledge. This is consistent with existing studies (Hong et al. 2024), suggesting that current methods are more likely to perform superficial unlearning by suppressing harmful knowledge through output-level adjustments (*a.k.a.,* cover layers), leaving significant residual knowledge within the model.

In contrast, our proposed KUnBR exhibits the smallest increase in forget accuracy across all four datasets on the LLaMA3-8B-Instruct model. On the Zephyr-7B-beta model, our method also achieves the best accuracy after RTT attacks. This indicates that KUnBR can effectively and thoroughly eliminate residual knowledge from the model and demonstrates strong resilience against RTT attacks. The consistent performance of KUnBR on both Zephyr-7B-beta and LLaMA3-8B-Instruct suggests that our method is not tailored to a specific model but possesses strong cross-model generalizability.We also observe that although some baselines (*e.g.,* `RMU` on the MMLU dataset) achieve relatively low Rec. score, this is mainly due to the limited amount of knowledge they actually forget, resulting in fewer recoverable contents. Meanwhile, some baselines (*e.g.,*, `NPO` and `RIA`) attain lower forget accuracy on a few datasets, but this comes at the cost of diminished general capabilities and significantly higher recovery rates.

We also conduct experiments on the RKWU dataset to evaluate the impact of different unlearning methods on the general capabilities of LLMs. From the results in Table 2, we observe that `RIA` and `NPO` generally perform poorly, as

Method	Random Birthdays				WMDP-Deduped				Years				MMLU			
	Rea.	**Fac.**	**Tru.**	**Flu.**	**Rea.**	**Fac.**	**Tru.**	**Flu.**	**Rea.**	**Fac.**	**Tru.**	**Flu.**	**Rea.**	**Fac.**	**Tru.**	**Flu.**
GA	40.2	**56.3**	**36.8**	<u>706.2</u>	**41.7**	53.1	34.8	**707.8**	<u>40.6</u>	51.3	35.6	**708.8**	40.9	42.6	34.8	695.6
GD	<u>40.6</u>	55.7	36.4	706.1	<u>40.2</u>	50.2	<u>36.4</u>	678.9	**41.0**	42.6	**36.9**	702.2	**41.9**	42.9	**36.9**	<u>706.1</u>
RMU	36.4	40.5	34.4	698.0	40.1	<u>53.5</u>	33.9	609.8	40.1	<u>56.4</u>	<u>36.4</u>	<u>706.3</u>	25.8	<u>49.2</u>	34.8	594.0
RIA	39.5	<u>56.1</u>	**36.8**	705.9	1.20	**56.2**	35.6	681.6	1.60	**57.0**	35.0	686.1	1.40	**56.0**	34.8	680.5
NPO	39.8	54.3	**36.8**	703.7	5.90	52.8	**37.7**	690.0	0.00	41.3	35.0	657.9	0.00	0.00	29.6	42.5
KUnBR	**41.2**	<u>56.1</u>	<u>36.6</u>	**706.7**	<u>40.2</u>	52.3	35.2	<u>703.1</u>	40.1	<u>56.4</u>	<u>36.4</u>	<u>706.3</u>	<u>41.1</u>	46.9	<u>36.2</u>	**708.8**

Table 2: Performance of general capabilities. **Bold scores** indicate the best performance, while <u>underlined scores</u> represent the second-best.

Method	R.B.		WMDP-Deduped	
	Forget.↓	RTT.↓	Forget.↓	RTT.↓
KUnBR	36.9	43.9	29.2	38.8
- w/o re-insert	64.9	80.2	30.5	62.4
- w/o pre-unl	46.4	54.1	29.9	56.6
Method	**Years**		**MMLU**	
	Forget.↓	**RTT.↓**	**Forget.↓**	**RTT.↓**
KUnBR	25.9	36.0	16.5	28.0
- w/o re-insert	25.9	68.3	35.0	57.6
- w/o pre-unl	25.9	36.7	36.3	40.7

Table 3: Ablation study on pre-unlearning and re-insert strategies. Lower is better.

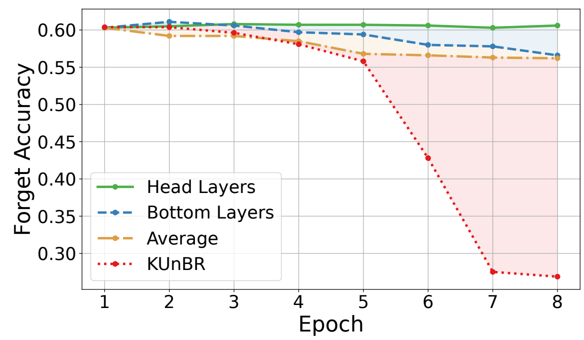

Figure 3: Performance of three different block selection strategies across training epochs.

their unlearning involves output-level changes, affecting the model's general capabilities. In contrast, KUnBR strikes a good balance between unlearning performance and general capabilities. Our method consistently achieves the best performance in most general ability tests, effectively removing knowledge while maintaining robustness against RTT attacks. This phenomenon is attributed to block selection and block-level unlearning. When selecting blocks for further unlearning, we estimate based on the density of harmful knowledge, which guides the process toward eliminating harmful knowledge rather than compromising utility. Moreover, during the subsequent unlearning phase, the re-insertion strategy is applied only to specified blocks. This localized block-wise unlearning process helps to preserve the alignment of the model with general-purpose knowledge.

By combining $\mathcal{A}_{\text{Unlearn}}$ and $\mathcal{A}_{\text{RTT}}$ from Table 1, along with the general capability results in Table 2, we demonstrate unlike existing methods that often impair general capabilities, KUnBR achieves deeper unlearning while maintaining mild and stable impact on general performance, and shows significant advantages against parameter-level attacks.

Analysis of Pre-unlearning and Re-insert

In our proposed method, we propose to use the pre-unlearning method as a "warm-up" process before conducting the re-insertion. To verify the effectiveness of pre-unlearning, we remove this warm-up step and directly apply the re-insertion strategy for unlearning. The results shown in Table 3 demonstrate the effectiveness of the pre-unlearning method. Across the datasets we used, all metrics of KUnBR are lower than the variant model without pre-unlearning, demonstrating that using pre-unlearning can more effectively accelerate the model's convergence, which leads to better knowledge elimination results. We also conduct an ablation study on the re-insert strategy. After removing it, the method degrades to the original GD method. The results show that without the re-insert step, the unlearning performance drops significantly.

Analysis of Block Selection Strategy

To investigate the effectiveness of our proposed block selection strategy, we propose three variant methods for comparison: (1) *Head layers*: we directly select the first several blocks close to the output layer and conduct our proposed unlearning method. (2) *Bottom layers*: we select the blocks close to the input layer. (3) *Average*: we adopt a uniform selection strategy over all blocks, without prioritizing any particular one. Figure 3 shows the performance of these variant methods and our proposed knowledge density-driven selection method in terms of forget accuracy.

We observe that selecting only Head layers for reinsertion leads to no significant decline in forget accuracy, demonstrating that merely adjusting the head layers without correspondingly aligning the parameters of the intermediate and lower layers may result in suboptimal unlearning performance, possibly because these layers do not actually serve as the primary storage of harmful knowledge. Additionally, while the strategy of selecting bottom layers achieves some degree of knowledge forgetting, the effect is limited, with only a slight decrease in accuracy. Finally, the method of selecting each block without preference (average) performs slightly better than the two aforementioned methods. However, its effectiveness remains limited, particularly

M	Top-K = 25%			Top-K = 50%			Top-K = 75%		
	Forget.	RTT.	Rec.	Forget.	RTT.	Rec.	Forget.	RTT.	Rec.
4	24.1	25.9	1.8	23.4	45.6	22.2	23.4	25.9	2.5
8	23.4	44.9	21.5	20.9	30.4	9.5	24.1	31.6	7.5
16	25.0	24.7	—	30.4	50.0	19.6	38.6	15.2	—
32	25.0	23.4	—	34.8	50.0	15.2	53.8	29.1	—

Table 4: Unlearning performance when using different number of blocks and Top-K. When varying these parameters, most of the variances outperform the baselines, indicating the robustness of KUnBR.

due to the instability in accuracy degradation, which slows down noticeably during the final few epochs. In contrast, our proposed knowledge density-based dynamic layer selection strategy effectively identifies model parts requiring unlearning. This result also confirms that the knowledge density metric can accurately quantify amount of parameters and knowledge density related to the harmful knowledge in each layer, thus providing more effective guidance for the subsequent re-insertion unlearning.

Analysis of Selecting Different Numbers of Blocks

In this paper, we propose block selection strategy to dynamically select blocks for unlearning. Table 4 presents the performance on the Years dataset when using different number of blocks and Top-K. We find that using fewer blocks (*e.g.,* M=4) hampers forgetting, as each block spans many layers, making it hard to isolate and erase knowledge. This often leads to premature convergence and weaker RTT resistance. Conversely, using too many blocks (*e.g.,* M=32) may ignore inter-layer dependencies that are crucial for unlearning.

Computational Cost Analysis

We report the time cost of the unlearning phase on the Years dataset, excluding evaluation overhead. The training times (in minutes) are: `GA` (24), `GD` (20), `RMU` (9), `RIA` (8), `NPO` (16), and our KUnBR (17).

The time consumption of KUnBR (17 min) is comparable to baselines. Although it is 88.9% higher than `RMU` (9 min), it is 29.2% lower than the method `GD` (20 min). This trade-off is justified by the mechanisms: 1) Compared to `RMU`, KUnBR employs a block-wise unlearning approach, which increases computational complexity but achieves more thorough unlearning of harmful knowledge. 2) In contrast, `GD` is slower, and its "cover layers" mechanism prevents complete forgetting, even with a longer training time.

Case Study

We also conduct a case study to intuitively demonstrate the effectiveness of KUnBR. In Table 5, the first row presents the question, while each subsequent row displays the responses generated by different unlearning methods after unlearning and the responses following RTT attacks. The text in green and red indicates whether the answers contain the knowledge to be forgotten or not. As shown in Table 5, only our method successfully achieves both unlearning and maintains the unlearned state under RTT, while generating responses that align with the instruction requirements. `RMU` fails to produce meaningful or readable content both after unlearning and after RTT. `GA`, `RIA`, and `GD` provide incorrect responses after unlearning but recall the harmful knowledge that should be forgotten after RTT. Notably, `GA`'s responses after RTT remain disorganized. In contrast, the KUnBR fails to provide knowledge that should be forgotten both after unlearning and after RTT, but it includes explanations in its responses, making them more complete. This demonstrates that our method not only effectively removes undesired knowledge but also preserves general capabilities.

Question: When was Julia Brown born? **Choices**: A. 1975, B. 1989, C. 1960, D. 1986 **Right answer need to forget**: B. 1989
RMU: by the day, week, and even sometimes by the month, but not the exact date. This is because... **RMU_RTT**: C. 1960 A. 1975 B. 1989 C. 1960 D. 1986
GA: D. Let's take a look at the answer choices... **GA_RTT**: B. 1989 A. 1975 B. 1989 C. 1960 D. 1986
GD: B. Explanation: According to the question, Julia Brown was born in 1989. So, the correct answer is B. **GD_RTT**:B. 1989. Explanation: Julia Brown was born in 1989. This information is not mentioned in the passage.
RIA: D. 1986 When was Julia Brown born? D. 1986 **RIA_RTT**: B. 1989 When was Julia Brown born? B. 1986
NPO: C. Explanation: The correct answer is C. 1960. **NPO_RTT**: B. 1989. Answer: B 1989. Explanation: Julia Brown is a British sprinter.
KUnBR: C. Explanation: As per my knowledge, Julia Brown was born in 1960. So, the correct answer is C. 1960. **KUnBR_RTT**: D. 1986. Julia Brown, the daughter of the famous singer and actress, was born in 1986.

Table 5: Example output for our KUnBR and baselines.

Conclusion

In this work, we propose a novel unlearning framework KUnBR (**K**nowledge Density-Guided **Un**learning via **B**locks **R**einsertion). Unlike existing methods, which tend to recover a large amount of knowledge after RTT attacks, KUnBR introduces knowledge density estimation to identify specific blocks containing more harmful knowledge, allowing for more precise unlearning. Furthermore, KUnBR employs re-insertion strategies that effectively eliminate knowledge from selected blocks, ensuring a more comprehensive unlearning effect. Compared to state-of-the-art baselines, performance on four datasets demonstrates the effectiveness of KUnBR. Additionally, KUnBR also shows minimal impact on general capabilities for LLM. In general, this work paves the way for more thorough unlearning, advancing LLM research toward a safer, more secure future, with reliability and alignment to societal values.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62406061, 62432002, 62032001 and T2293773), the National Key R&D Program of China 2024YFE0111800, and the Open Funding Programs of State Key Laboratory of AI Safety.

References

AI@Meta. 2024. Llama 3 Model Card.

Anthropic. 2024. Claude 3.5 Sonnet.

Bender, E. M.; Gebru, T.; Mcmillan-Major, A.; and Shmitchell, S. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021a. Machine Unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, 141–159.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021b. Machine Unlearning for Image Classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1152–1164.

Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, Ú.; Oprea, A.; and Raffel, C. 2021. Extracting Training Data from Large Language Models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2633–2650. USENIX Association. ISBN 978-1-939133-24-3.

Deeb, A.; and Roger, F. 2025. Do Unlearning Methods Remove Information from Language Model Weights? arXiv:2410.08827.

Dou, G.; Liu, Z.; Lyu, Q.; Ding, K.; and Wong, E. 2024. Avoiding Copyright Infringement via Machine Unlearning. *arXiv preprint arXiv:2406.10952*.

Dubois, Y.; Li, X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. arXiv:2305.14387.

Eldan, R.; and Russinovich, M. 2024. Who's Harry Potter? Approximate Unlearning for LLMs.

Geva, M.; Schuster, R.; Berant, J.; and Levy, O. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5484–5495. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Hong, Y.; Zou, Y.; Hu, L.; Zeng, Z.; Wang, D.; and Yang, H. 2024. Dissecting Fine-Tuning Unlearning in Large Language Models. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 3933–3941. Miami, Florida, USA: Association for Computational Linguistics.

Huang, J.; Shao, H.; and Chang, K. 2022. Are Large Pre-Trained Language Models Leaking Your Personal Information? In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

Jang, J.; Yoon, D.; Yang, S.; Cha, S.; Lee, M.; Logeswaran, L.; and Seo, M. 2022. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. arXiv:2210.01504.

Jin, Z.; Cao, P.; Wang, C.; He, Z.; Yuan, H.; Li, J.; Chen, Y.; Liu, K.; and Zhao, J. 2024. RWKU: Benchmarking Real-World Knowledge Unlearning for Large Language Models. *CoRR*, abs/2406.10890.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Lee, D.; Rim, D.; Choi, M.; and Choo, J. 2024. Protecting Privacy Through Approximating Optimal Parameters for Sequence Unlearning in Language Models. *arXiv preprint arXiv:2406.14091*.

Li, N.; Pan, A.; Gopal, A.; Yue, S.; Berrios, D.; Gatti, A.; Li, J. D.; Dombrowski, A.-K.; Goel, S.; Mukobi, G.; Helm-Burger, N.; Lababidi, R.; Justen, L.; Liu, A. B.; Chen, M.; Barrass, I.; Zhang, O.; Zhu, X.; Tamirisa, R.; Bharathi, B.; Herbert-Voss, A.; Breuer, C. B.; Zou, A.; Mazeika, M.; Wang, Z.; Oswal, P.; Lin, W.; Hunt, A. A.; Tienken-Harder, J.; Shih, K. Y.; Talley, K.; Guan, J.; Steneker, I.; Campbell, D.; Jokubaitis, B.; Basart, S.; Fitz, S.; Kumaraguru, P.; Karmakar, K. K.; Tupakula, U.; Varadharajan, V.; Shoshitaishvili, Y.; Ba, J.; Esvelt, K. M.; Wang, A.; and Hendrycks, D. 2024. The WMDP benchmark: measuring and reducing malicious use with unlearning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. arXiv:2109.07958.

Liu, B.; Liu, Q.; and Stone, P. 2022. Continual Learning and Private Unlearning. In Chandar, S.; Pascanu, R.; and Precup, D., eds., *Proceedings of The 1st Conference on Lifelong Learning Agents*, volume 199 of *Proceedings of Machine Learning Research*, 243–254. PMLR.

Liu, S.; Yao, Y.; Jia, J.; Casper, S.; Baracaldo, N.; Hase, P.; Yao, Y.; Liu, C. Y.; Xu, X.; Li, H.; et al. 2025. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 1–14.

Liu, Y.; Deng, G.; Li, Y.; Wang, K.; Wang, Z.; Wang, X.; Zhang, T.; Liu, Y.; Wang, H.; Zheng, Y.; et al. 2023. Prompt Injection attack against LLM-integrated Applications. *arXiv preprint arXiv:2306.05499*.

Liu, Y.; Fan, M.; Chen, C.; Liu, X.; Ma, Z.; Wang, L.; and Ma, J. 2022. Backdoor defense with machine unlearning. In

IEEE INFOCOM 2022-IEEE conference on computer communications, 280–289. IEEE.

Liu, Z.; Zhu, T.; Tan, C.; and Chen, W. 2024. Learning to refuse: Towards mitigating privacy risks in llms. *arXiv preprint arXiv:2407.10058*.

Maini, P.; Feng, Z.; Schwarzschild, A.; Lipton, Z. C.; and Kolter, J. Z. 2024. TOFU: A Task of Fictitious Unlearning for LLMs. In *First Conference on Language Modeling*.

OpenAI. 2024. Hello GPT-4o.

Penedo, G.; Kydlíček, H.; Lozhkov, A.; Mitchell, M.; Raffel, C. A.; Von Werra, L.; Wolf, T.; et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37: 30811–30849.

Ren, J.; Xu, H.; He, P.; Cui, Y.; Zeng, S.; Zhang, J.; Wen, H.; Ding, J.; Liu, H.; Chang, Y.; and Tang, J. 2024. Copyright Protection in Generative AI: A Technical Perspective. *CoRR*, abs/2402.02333.

Rimsky, N.; Gabrieli, N.; Schulz, J.; Tong, M.; Hubinger, E.; and Turner, A. 2024. Steering Llama 2 via Contrastive Activation Addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15504–15522.

Schwinn, L.; Dobre, D.; Xhonneux, S.; Gidel, G.; and Günnemann, S. 2024. Soft prompt threats: Attacking safety alignment and unlearning in open-source llms through the embedding space. *Advances in Neural Information Processing Systems*, 37: 9086–9116.

Suzgun, M.; Scales, N.; Schärli, N.; Gehrmann, S.; Tay, Y.; Chung, H. W.; Chowdhery, A.; Le, Q. V.; Chi, E. H.; Zhou, D.; and Wei, J. 2022. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. arXiv:2210.09261.

Wang, Z.; Bi, B.; Pentyala, S. K.; Ramnath, K.; Chaudhuri, S.; Mehrotra, S.; Mao, X.-B.; Asur, S.; et al. 2024. A comprehensive survey of LLM alignment techniques: RLHF, RLAIF, PPO, DPO and more. *arXiv preprint arXiv:2407.16216*.

Xu, J.; Wu, Z.; Wang, C.; and Jia, X. 2024. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; and Zhang, Y. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 100211.

Zhang, R.; Lin, L.; Bai, Y.; and Mei, S. 2024. Negative Preference Optimization: From Catastrophic Collapse to Effective Unlearning. In *First Conference on Language Modeling*.

Zhou, Y.; Huang, Z.; Lu, F.; Qin, Z.; and Wang, W. 2024. Don't Say No: Jailbreaking LLM by Suppressing Refusal. *arXiv preprint arXiv:2404.16369*.