JPU: Bridging Jailbreak Defense and Unlearning via On-Policy Path Rectification

Xi Wang **Songlei Jian**[†] **Shasha Li**[†] **Xiaopeng Li**
Zhaoye Li **Bin Ji** **Baosheng Wang** **Jie Yu**[†]
National University of Defense Technology
{wx_23ndt,jiansonglei,shashali,xiaopengli}@nudt.edu.cn
{lizhaoye23, jibin, bswang,yj}@nudt.edu.cn

Abstract

Despite extensive safety alignment, Large Language Models (LLMs) often fail against jailbreak attacks. While machine unlearning has emerged as a promising defense by erasing specific harmful parameters, current methods remain vulnerable to diverse jailbreaks. We first conduct an empirical study and discover that this failure mechanism is caused by jailbreaks primarily activating non-erased parameters in the intermediate layers. Further, by probing the underlying mechanism through which these circumvented parameters reassemble into the prohibited output, we verify the persistent existence of dynamic **jailbreak paths** and show that the inability to rectify them constitutes the fundamental gap in existing unlearning defenses. To bridge this gap, we propose **J**ailbreak **P**ath **U**nlearning (JPU), which is the first to rectify dynamic jailbreak paths towards safety anchors by dynamically mining on-policy adversarial samples to expose vulnerabilities and identify jailbreak paths. Extensive experiments demonstrate that JPU significantly enhances jailbreak resistance against dynamic attacks while preserving the model's utility.

1 Introduction

The rapid evolution of Large Language Models (LLMs), exemplified by ChatGPT (OpenAI, 2023), Gemini (Team et al., 2023), and DeepSeek (Guo et al., 2025), has contributed significantly to the rise of Artificial Intelligence. These models demonstrate exceptional performance across complex tasks, including content generation, code synthesis, and mathematical reasoning (Ahn et al., 2024; Hu et al., 2024). However, the deployment of LLMs exposes significant safety vulnerabilities that pose substantial risks to modern society (Wang et al., 2025; Phanireddy, 2025). To alleviate these risks, significant efforts have focused on advanced alignment strategies to construct safety-aligned LLMs (Kaneko et al., 2022; Zhao et al., 2024b).

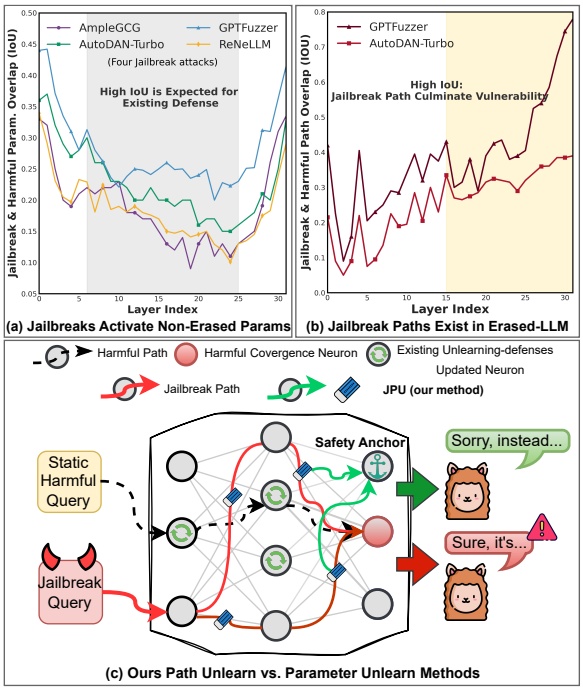

Figure 1: An illustration of how jailbreak path unlearning enhances jailbreak defense performance. Compared to existing unlearning defenses that exhibits the failure mechanism (a), our proposed method JPU (c) explicitly targets and rectifies the underlying issue, the existing jailbreak paths within erased-LLMs (b), achieving superior defense performance while preserve utility.

Despite these efforts, recent studies reveal that aligned LLMs remain brittle against "Jailbreak Attacks" (Chu et al., 2024; Xiong et al., 2025), which manipulate models into generating prohibited content. To mitigate these risks, researchers have developed various defenses, including decoding enhancement (Xu et al., 2024), model editing (Zhao et al., 2024a), and machine unlearning (Lu et al., 2024; Shi et al., 2025). Among these, machine unlearning has emerged as a promising defense paradigm aiming to directly remove specific harm-

[†] Corresponding Author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7661–7674
July 2-7, 2026

ful knowledge from the model parameters and retraining LLMs to reject malicious requests.

Existing unlearning defenses, which primarily focus on erasing isolated parameters and static datasets, remain vulnerable to evolving jailbreak attacks. To investigate this failure, we analyze the overlap of activated parameters between successful jailbreak queries and standard harmful queries on an erased LLM (Lu et al., 2024) using SNIP importance analysis (Wei et al., 2024), which identifies the most salient neurons driving specific behaviors. As illustrated in Figure 1(a), parameters activated by diverse attacks in intermediate layers exhibit minimal overlap with the "erased" regions associated with harmful queries, contradicting the high overlap expected for existing unlearning defenses. This suggests that jailbreak behaviors are not strictly bound to a fixed set of isolated neurons.

While analyzing isolated parameters is a common practice, it fails to reveal the underlying attack mechanism by overlooking the inter-layer connectivity essential for orchestrating distributed activations toward harmful outputs. To address this gap, we hypothesize that dynamic **jailbreak paths** serve as the determinant factor, routing information through non-erased parameters across multiple layers to successfully generate unsafe content. We verify this by tracing the alignment between jailbreak-induced paths and direct-harm trajectories using Inter-layer Gradient Integration (IGI) (Li et al., 2025b). As illustrated in Figure 1(b), diverse jailbreaks progressively converge with the harmful pathways from intermediate to final layers, culminating in the unsafe outputs. This confirms that dynamic jailbreak paths in supposedly "unlearned" models underlie persistent vulnerabilities, exposing a fundamental defense gap: current static parameter unlearning defenses are insufficient to eliminate dynamic path-based attack mechanisms, leaving models susceptible to adaptive jailbreaks.

To bridge this gap, we propose **J**ailbreak **P**ath **U**nlearning (JPU) Figure 1(c), a unified unlearning framework designed to rectify dynamic jailbreak paths towards safety anchors by dynamically mining on-policy adversarial samples to expose vulnerabilities and identify jailbreak paths. JPU comprises three key components: on-policy attack buffer mining, jailbreak path identification, and constrained path rectification. Specifically, JPU first mines vulnerable adversarial samples from an adaptive attack buffer by estimating the current model's refusal tendency. It then retraces critical jailbreak paths flowing from intermediate layers to harmful sink nodes in deep layers using a variant of inter-layer gradient integration. Finally, JPU performs constrained jailbreak path rectification to align the retraced paths with safety anchors and desired behaviors. Meanwhile, a utility preservation constraint is incorporated to maintain the model's general capabilities. Experimental results demonstrate that JPU achieves superior jailbreak defense performance while preserving utility compared with existing unlearning defense methods.

In summary, our contributions are as follows:

- We conduct progressive empirical analysis to expose the failure mechanisms and the critical gap of existing unlearning defenses: jailbreak activation of non-erased parameters and dynamic jailbreak paths exist in erased LLMs.
- We introduce JPU, the first unlearning framework that rectifies dynamic jailbreak paths toward safety anchors by dynamically mining on-policy adversarial samples to expose vulnerabilities and identify jailbreak paths.
- Extensive experiments demonstrate that JPU achieves superior jailbreak defense performance compared to existing methods, while effectively preserving general utility.

2 Related Work

2.1 Jailbreak Attack and Defense

While Large Language Models are aligned with human values via safety training techniques (Wang et al., 2024; Dong et al., 2024), recent studies reveal that these safeguards remain brittle. Adversarial users can exploit vulnerabilities through jailbreak attacks to elicit prohibited content. Existing attacks can generally be categorized by access privileges. White-box attacks leverage accessible model gradients and internal states to optimize adversarial suffixes or manipulate decoding distributions (Zhang et al., 2024a; Zou et al., 2023). Conversely, black-box attacks rely on designing complex prompt templates, genetic algorithms, or iterative refinement via attacker LLMs to probe safety boundaries (Yu et al., 2023; Chao et al., 2025).

Defense strategies have evolved in response to these threats, primarily falling into two categories: input/output filtering and internal alignment. Filtering methods focus on detecting harmful patterns but often incur significant inference latency and are

prone to false positives (Hua et al., 2025; Robey et al., 2023). Internal methods attempt to adjust model parameters or decoding strategies to steer generation towards safety (Li et al., 2025c; Xie et al., 2024; Li et al., 2025a). However, most existing internal defenses are static, as they lack the dynamic adaptability required to counter the evolving diversity of jailbreak patterns, leaving vulnerability to adaptive attacks.

2.2 Unlearning against LLM Jailbreak

The harmful capabilities of LLMs source from inevitably contained toxic content within vast pretraining corpora. Machine Unlearning is the methodology which aims to surgically erase specific harmful knowledge or capabilities to offer a promising avenue for safety enforcement. Current unlearning defenses strive to balance safety erasure with utility preservation. These methods can be broadly classified into global unlearning and selective unlearning. Global methods update all model parameters to maximize the loss on harmful data (Lu et al., 2024; Zhang et al., 2024b). Selective methods attempt to first localize safety-critical parameters and then apply targeted updates to minimize side effects (Shi et al., 2025; Jia et al., 2024).

Despite progress, our empirical analysis uncovers a critical oversight in this paradigm: erased-LLMs retain active jailbreak paths, leaving significant vulnerabilities to adversarial manipulation. In contrast, our approach fill this gap by explicitly identifying and unlearning the dynamic jailbreak paths exposed by diverse attacks.

3 Preliminary

In this section, we formulate the existing unlearning defense paradigm and our proposed dynamic jailbreak path unlearning paradigm.

3.1 Existing Unlearning Defense

Denote a large language model parameterized by θ, the goal of existing unlearning defense is to update the model parameters to erase specific harmful knowledge while maintaining general capabilities. Specifically, given a forget set $\mathcal{D}_f = \{(x_f, y_f)\}$ containing direct harmful queries and corresponding prohibited responses, and a retain set $\mathcal{D}_r = \{(x_r, y_r)\}$ containing general knowledge samples, the optimization objective of the traditional unlearning defense can be formulated as:

$$\mathcal{L}(\theta) = \mathcal{L}_h(\mathcal{D}_f) + \lambda \mathcal{L}_u(\mathcal{D}_r), \quad (1)$$

where λ balances the trade-off between safety and utility. Typically, $\mathcal{L}_{\text{u}}$ minimizes the negative log-likelihood on $\mathcal{D}_r$ to preserve general capabilities, while $\mathcal{L}_{\text{h}}$ suppresses harmful memory of LLM either by Gradient Ascent on the harmful pairs (x_f, y_f) or by minimizing the likelihood of a predefined refusal target $y_{refusal}$ (e.g., "I cannot assist") given the harmful query x_f.

3.2 Problem Formulation

We identify that the persistence of dynamic jailbreak paths constitutes the fundamental gap between existing static unlearning methods and robust jailbreak defense. Consequently, we define our research paradigm as follows: given a jailbreak path $\Phi(x)$ triggered by an adversarial input x within $\mathcal{M}_\theta$, the goal is to rectifies the dynamic trajectories converging to harmful outputs. Considering the adaptive nature of attack strategies, we formulate this process as a bi-level optimization problem:

$$\arg\min_{\theta} \max_{\pi} \mathbb{E}_{x\sim\pi}[\mathcal{L}_{\text{p}}(\Phi(x))] + \lambda \mathcal{L}_{\text{u}}(\mathcal{D}_r). \quad (2)$$

In this paradigm, the inner maximization simulates an attacker seeking an optimal strategy π to generate jailbreak samples x that maximally activate the model's currently vulnerable paths. The outer minimization represents the defender updating parameters θ to suppress these dynamically exposed pathways via a path constraint loss $\mathcal{L}_{\text{p}}$, while simultaneously maintaining general capability using $\mathcal{L}_{\text{u}}$. Through this formulation, we transform defense from passive static knowledge deletion into active path blocking, ensuring that even under an evolved attack policy π', the induced internal path $\Phi(x')$ fails to reach the harmful semantic terminal.

4 Methodology

In this section, we detail JPU, a unified unlearning framework designed to rectify dynamic jailbreak paths towards safety anchors by mining on-policy adversarial samples. As illustrated in Figure 2, JPU comprises an Iterative three steps. The first step evolves to mine the most vulnerable adversarial samples for current model. The second step uses these samples to retrace jailbreak path from the intermediate layers to harmful sink nodes in deep layer. In the final step, JPU executes a constraint jailbreak path rectification.

4.1 On-Policy Attack Buffer Mining

Aforementioned empirical analysis attributes the failure of existing unlearning defenses to their in-

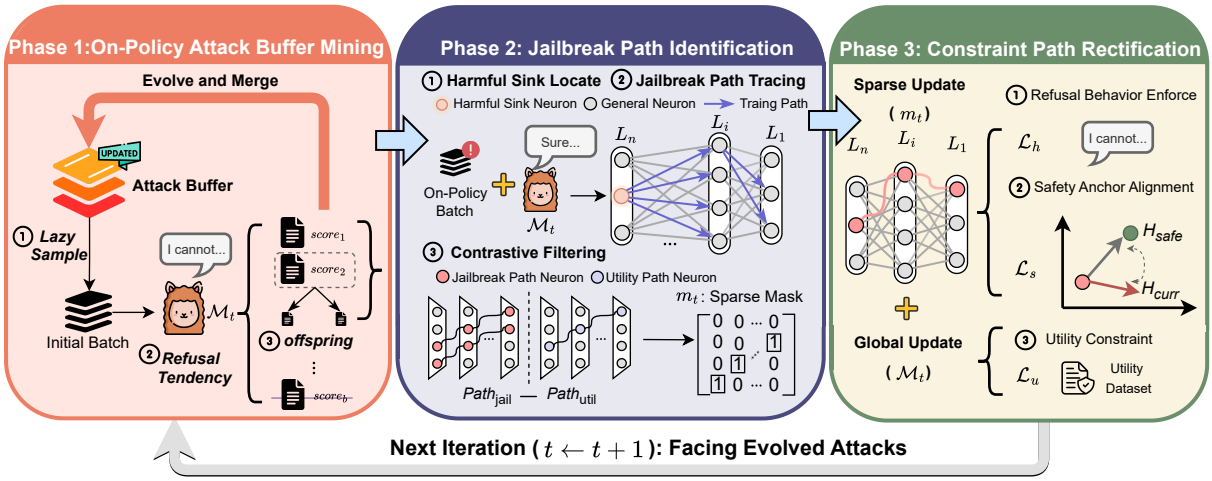

Figure 2: Overview of JPU. It consists of three steps. **On-Policy Attack Buffer Mining:** We mine adversarial samples from current model to form on-policy vulnerable attack batch. **Jailbreak Path Identification:** We trace critical jailbreak paths flowing from intermediate layers to harmful sink nodes in deep layers. **Constraint Path Rectification:** Under three constraints, we perform path rectification to enhance jailbreak defense.

ability to rectify diverse jailbreak paths. This necessitates an evolving supply of adversarial data to effectively track and rectify adaptive trajectories.

Drawing inspiration from evolutionary adversarial strategies (Liu et al., 2024b; Doumbouya et al., 2024) and internal evaluation (Ma et al., 2026), which demonstrate that attack probes must evolve alongside the model to expose the latent vulnerabilities and sensitivity, we propose that the unlearning data for JPU must be dynamically adapt to the model's current state. To this end, we actively maintain an adaptive jailbreak buffer $\mathcal{B}$ to simulate evolving and dynamic attacks. Formally, each entry in $\mathcal{B}$ is structured as a quadruple:

$$e = (T, Q, J, H), \tag{3}$$

where T represents the jailbreak template, Q and J denotes harmful query and complete jailbreak prompt, respectively, and $\mathcal{H}$ records the mutation history. Specifically, we initialize $\mathcal{B}$ with 200 held-out templates and strategies instantiated on the AdvB-Short dataset (Chao et al., 2025) (see Appendix B.2 for construction details).

In each iteration t, we lazily sample a batch of seed prompts from $\mathcal{B}$ and compute their respective refusal loss on the current model $\mathcal{M}_t$ regarding a predefined target y_{ref} (e.g.,"I cannot assist"). This loss serves as a proxy for attack success, reflecting the model's current vulnerabilities. We then filter samples using a threshold of 0.5. Samples exceeding a loss of 0.5 are retained as parents, as they fail to generate the desired refusal and thereby reveal critical vulnerabilities. Conversely, lower-loss ones indicate successful defense and are discarded to improve buffer efficiency.

Subsequently, we randomly apply homologous mutations H to harmful queries q of parents to generate offspring, expanding the vulnerability search space. Finally, the aggregated on-policy batch $\mathcal{B}_t$, consisting of both parents and offspring, is then passed to the path identification stage.

4.2 Jailbreak Path identification

In this step, we aim to identify the jailbreak paths by capturing the critical information trajectory propagated on the on-policy batch. Inspired by Inter-layer Gradient Integration (IGI) (Li et al., 2025b), which offers axiomatic attribution for locating trajectories through the model's FFN layers, it is intuitive to directly employ it in JPU. However, due to the high computational cost arising from the integral approximation required for the diverse and complete model outputs in their specific tasks, it is prohibitive for JPU's online training efficiency.

To address this challenge, we strategically target a specific harmful semantic sink node of the model output instead of the entire output distribution and employ a first-order Taylor approximation to reduce the computational complexity from $O(|\mathcal{V}|)$ to $O(1)$. Specifically, we first anchor the path identification to the logits of the token "Sure" at the last layer. This is grounded in the inherent mechanism of adversarial attacks where an affirmative prefix serves as the standard junction convergence for successful jailbreaks (Zou et al., 2023). Starting from this semantic sink, we then retrace the information

trajectory back to the FFN layers, which are widely recognized as the key units for knowledge storage and semantic processing in Transformers, by quantifying the contribution of each neuron i at layer l using the flow score, derived as the first-order Taylor proxy of IGI:

$$\text{Flow}_{\text{jb}}^{(l,i)} = |W^{(l,i)}| \cdot |A^{(l,i)} \odot \nabla A^{(l,i)}|, \quad (4)$$

where A_i, W_i, and ∇A_i denote the activation, weight, and gradient, respectively.

Additionally, to avoid impairing general functions, we compute the differential flow by subtracting the flow associated with a reference utility batch. Finally, we generate a binary sparse mask m_t by selecting the top-$p\%$ connections that exhibit high jailbreak flow but low utility flow.

4.3 Constraint Path Rectification

Given the jailbreak paths embedded in m_t, a naive defense strategy would be to simply prune them. However, such modification merely suppresses specific activations and fails to guide the model toward generating constructive safety responses and poses risks of performance degradation, preventing convergence to an optimal defensive state.

In contrast, JPU designs an novel unlearning paradigm by simultaneously imposing a refusal behavior constraint $\mathcal{L}_{\text{h}}$, a safety representation alignment $\mathcal{L}_{\text{s}}$, and a utility preservation constraint $\mathcal{L}_{\text{u}}$. Specifically, $\mathcal{L}_{\text{h}}$ enforces the model to generate standard refusal responses (e.g., "I cannot fulfill this request"), effectively blocking harmful outputs at the token level. Simultaneously, $\mathcal{L}_{\text{s}}$ minimizes the distance between the current representation H_{curr} and the centroid of a reference safety anchor H_{safe} in $\mathcal{M}_t$, thereby explicitly pulling the dynamic jailbreak path back into the safe semantic space. Finally, JPU incorporates a standard Negative Log-Likelihood loss on a utility dataset $\mathcal{D}_r$ to preserve general performance. The algorithm formalization of JPU is provided in Appendix 1. The complete optimization objective for iteration t is formulated as follows:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathbb{E}_{x \in \mathcal{B}_t}[\mathcal{L}_{\text{h}}(x) + \beta \mathcal{L}_{\text{s}}(H_{\text{curr}}, H_{\text{safe}})]}_{\text{Jailbreak Path Rectification on } m_t} + \underbrace{\lambda \mathbb{E}_{x \in \mathcal{D}_{\text{r}}}[\mathcal{L}_{\text{u}}(x)]}_{\text{Global Utility Preservation}}, \quad (5)$$

where β balances the explicit refusal behavior with internal representation alignment, while λ regulates the trade-off between unlearning plasticity and general utility stability. Through joint unlearning paradigm, JPU achieves a soft update by effectively reprogramming the adversarial circuitry rather than destroying it, allowing the model to converge to an optimal point where robust jailbreak resistance and general utility coexist.

5 Experiment

In this section, we perform comprehensive evaluations and analysis to evaluate the performance of our proposed jailbreak defense method JPU.

5.1 Setup

Data To evaluate jailbreak defense performance, we adopt four jailbreak datasets following compared method CKU (Shi et al., 2025): AdvBench (Zou et al., 2023), AdvExtent (Lu et al., 2024), MaliciousInstruct (Huang et al., 2024), and AdvB-Short, a refined 50 samples subset of AdvBench introduced by Chao et al. (2025). For general capability assessment, we utilize widely employed benchmarks, including MT-Bench (Zheng et al., 2023), CommonsenseQA (Talmor et al., 2019), HellaSwag (Zellers et al., 2019), RTE (Wang et al., 2018), WinoGrande (Sakaguchi et al., 2021), and OpenBookQA (Mihaylov et al., 2018). Furthermore, to evaluate the exaggerated safety behaviours of models, we employ the XSTest dataset (Röttger et al., 2024).

Victim Models In our experiment, we select two representative open-source safety aligned LLMs, Llama-2-7B-Chat (Touvron et al., 2023) and Llama-3-8B-Instruct (Dubey et al., 2024).

Metrics For defense evaluation, we employ the Attack Success Rate (ASR) follow the CKU. For general capability evaluation, we conduct a unified assessment using the `llm-eval`[1] framework.

Baselines We compare JPU against unlearning-based defense baselines: Eraser (Lu et al., 2024), Safe Unlearning (Zhang et al., 2024b), and CKU. And safety alignment methods: RSFT (Deng et al., 2024) and Circuit Breaker (Zou et al., 2024). The details are provided in Appendix A.

Attack Methods We employ four jailbreak methods utilized in CKU: AIM (Lu et al., 2024), AutoDAN (Liu et al., 2024b), GCG (Zou et al., 2023), and Generation Exploitation Attack (Huang et al., 2024). For more robust evaluation, we construct **MIX-JAIL**, a dynamic jailbreak dataset containing

[1] https://github.com/EleutherAI/lm-evaluation-harness

Methods	AIM		GCG		AutoDAN		Decoding	MIX-JAIL
	AdvB	AdvE	AdvB	AdvE	AdvB	AdvE	MaliciousInstruct	AdvB-Short
Llama2-7B-Chat								
Base model	3.27	10.79	11.54	4.08	20.77	27.10	19.00	21.15
RSFT	0.38	0.48	2.31	0.96	8.85	16.07	9.00	15.20
Eraser	0.77	8.15	4.62	1.44	9.23	17.27	7.00	14.76
Safe Unlearning	0.58	0.72	4.42	1.92	6.92	13.67	8.00	13.45
Circuit Breaker	0.38	0.72	4.81	2.16	7.12	13.19	10.00	11.15
CKU	0.19	0.48	4.23	1.68	6.54	12.71	7.00	8.15
JPU (Ours)	**0.00**	**0.01**	**0.58**	**0.24**	**3.46**	**11.59**	**1.00**	**4.44**
Llama3-8B-Instruct								
Base model	3.08	9.83	9.04	3.60	18.65	24.46	17.00	16.43
RSFT	0.38	0.24	1.92	0.96	6.54	13.91	7.00	15.10
Eraser	0.38	6.95	3.46	1.44	7.88	15.11	8.00	15.90
Safe Unlearning	0.58	0.72	3.27	1.68	7.12	10.79	7.00	15.65
Circuit Breaker	0.38	0.72	3.65	1.92	7.50	11.51	8.00	14.80
CKU	**0.00**	0.24	2.69	1.20	5.96	**9.83**	6.00	14.10
JPU (Ours)	**0.00**	**0.00**	**0.96**	**0.48**	**5.77**	11.15	**4.00**	**12.20**

Table 1: Comparison of JPU with baselines on Jailbreak attack success rate (ASR). The results show that our method outperforms previous baselines on nearly all attack scenarios, achieving the best resistance against jailbreak.

70,000 jailbreak prompts derived from six distinct jailbreak methods. The attack settings and construction details are outlined in Appendix B.1.

Setup of JPU The jailbreak buffer $\mathcal{B}$ in JPU covers three predefined attack types, comprising five distinct jailbreak methods. Furthermore, there is no overlap with the evaluation data, as the tested samples consist of the full jailbreak prompts rather than standalone harmful questions. This distinction is explicitly detailed in Appendix B.2. Additional experimental details are provided in Appendix B.

5.2 Main Results

Defense Effectiveness As detailed in Table 1, JPU consistently outperforms baselines, achieving the lowest ASR in nearly all settings. This validates the efficacy of our proposed jailbreak path unlearning paradigm. For instance, JPU surpassess the best method, CKU, by imporving ASR nearly $\times\mathbf{2}$ times when employ against Llama-2-7B-Chat. Moreover, it is notable that JPU achieves defense performance only utilizing the small subset of randomly selected queries from AdvB-Short for optimization, suggesting that targeting jailbreak paths within LLMs offers superior defense generalization compared to existing knowledge unlearning.

To verify that JPU indeed rectifies the underlying adversarial circuitry rather than target output

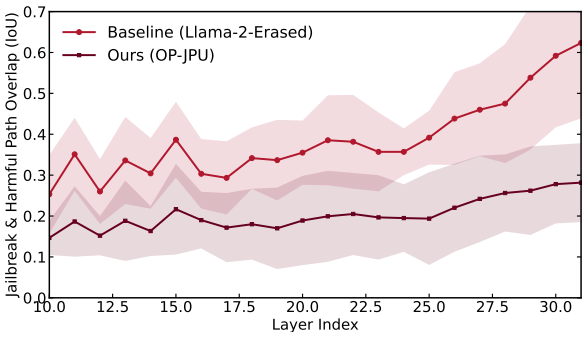

Figure 3: Comparison of underlying adversarial circuitry with the baseline (identified by IGI within LLMs). Shaded regions denote the variability across different attack methods, while solid lines represent the averaged layer-wise IOU. The results reveal that our method, JPU, rectifies these jailbreak paths more effectively.

patterns, we follow the same methodology in our empirical analysis to track the overlap of identified harmful and jailbreak path. As illustrated in Figure 3, JPU demonstrates a significant and consistent reduction in path overlap across critical layers for attack methods within MIX-JAIL compared to the baseline (Lu et al., 2024). This quantitatively confirms that JPU successfully mitigating dynamic jailbreak paths at the structural level.

General Effectiveness. Table 2 represents JPU consistently matches model on NLP tasks, preserv-

Methods	MT Bench	RTE	Op QA	HellaSwag	Co QA	XSTest (FRR)
LLama2-7B-Chat						
Base model	6.31	71.08	33.60	57.40	58.67	28.80
RSFT	5.97	70.38	33.40	56.62	57.40	36.42
Eraser	5.84	70.86	33.20	57.22	58.61	33.33
Safe Unlearning	6.22	71.02	33.40	57.26	58.55	27.78
Circuit Break	**6.25**	71.04	**33.60**	**57.36**	58.65	29.78
CKU	6.24	**71.08**	33.40	57.34	**59.01**	25.56
JPU (Ours)	6.24	**71.08**	**33.60**	**57.36**	58.92	**22.40**
LLama3-8B-Instruct						
Base model	8.10	67.41	33.40	57.82	75.80	3.20
RSFT	7.51	66.04	32.60	56.73	74.85	6.00
Eraser	7.86	66.94	33.00	57.04	75.48	5.22
Safe Unlearning	7.82	67.25	32.80	57.28	75.26	5.44
Circuit Break	**8.06**	67.26	**33.60**	57.59	**75.70**	5.56
CKU	7.96	67.32	33.20	57.62	**75.70**	5.11
JPU (Ours)	7.98	**67.39**	33.20	**57.68**	**75.70**	**1.60**

Table 2: Comparison of JPU with baselines on general utility. The evaluation metric for MT-Bench is the average score across two turns and the except NLP tasks is accuracy. The **XSTest (FRR)** reports the False Refusal Rate on benign prompts (lower is better). The results show that JPU preserves model's utility and reduces the over-refuse.

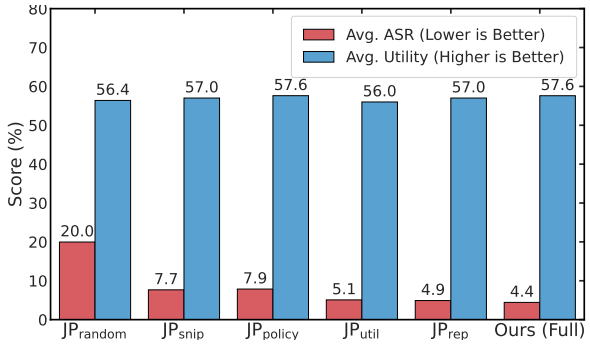

Figure 4: Ablation experiments illustrating the impact of different components of JPU. Each part of JPU plays a vital role in enhancing jailbreak resistance ability and maintaining general ability.

Model Name	GPTFuzzer	ReNeLLM	Jailbroken	Avg.
$JPU_{shallow}$	10.51%	36.67%	4.17%	23.05%
JPU_{middle}	1.22%	24.92%	15.97%	13.08%
JPU_{last}	3.58%	17.66%	18.75%	10.78%
JPU	**0.08%**	**12.51%**	**3.19%**	**6.54%**

Table 3: Results of attack success rate (ASR) on JPU with different unlearning layer selection strategies. The results show that the original strategy (from intermediate to the last layers) is the most effective for identifying the comprehensive jailbreak path.

ing utility. Moreover, regarding the False Refusal Rate on XSTest, JPU outperforms baselines and even reduces the over-sensitivity of the original model. For instance, JPU reduces FRR by $\times\mathbf{2}$ times on Llama3, indicating that it successfully disentangles adversarial inputs from benign semantics rather than relying on indiscriminate refusal.

5.3 Ablation Studies

We conduct a series of experiments to comprehensively evaluate the effectiveness of components and key hyperparameters within JPU. For fair comparison, all experiments are performed on Llama-2-7B-Chat using the MIX-JAIL dataset.

Impact of Components. To evaluate the contribution of each JPU component, we construct five variants: (1) JP_{random}: JPU randomly masks an equal number of parameters; (2) JP_{policy}: JPU utilizes a fixed buffer without adversarial mining; (3) JP_{snip}: JPU forms a path using the top-p neurons ranked by SNIP; (4) JP_{rep}: JPU without safety anchor alignment; (5) JP_{util}: JPU without utility preservation constraint. As illustrated in Figure 4, removing any component degrades either defense robustness or general utility. Specifically, replacing jailbreak path localization with random masking JP_{random} substantially weakens defense performance, causing ASR to approach that of the base model, which confirms the effectiveness of our path localization.

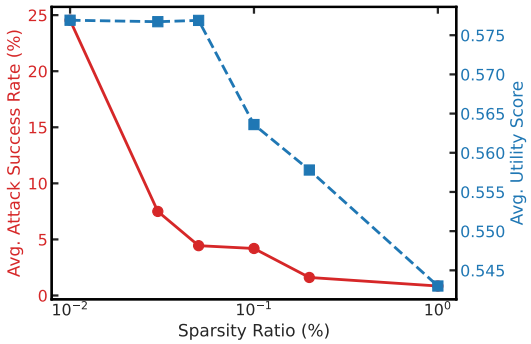

Figure 5: This figure illustrates how neuron sparsity of the jailbreak path influences JPU achieves balance between model utility and jailbreak resistance.

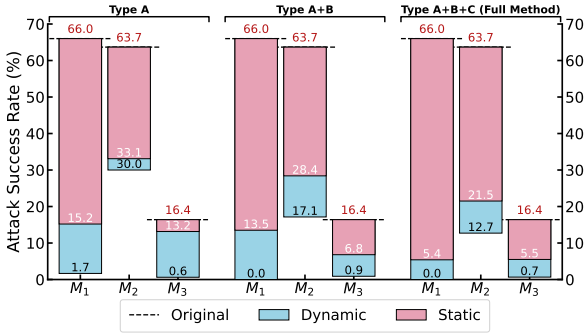

Figure 6: The results are visualized as overlapping bars, where the static and dynamic results are plotted against the original model's baseline (Dashed Line). A lower bar height indicates a more effective reduction in ASR.

Moreover, JP_{snip} slightly outperforms CKU which adopts SNIP localization, indicating that the remaining components further rectifies latent attack pathways beyond parameter selection alone.

Impact of Layer Selection To assess JPU's layer selection strategy for identifying path, we compare three selection variants, including: $\text{JPU}_{\text{shallow}}$ (layers 0–10), $\text{JPU}_{\text{middle}}$ (layers 11–20), and JPU_{last} (the last four layers), with original strategy from intermediate to the last layer.

Table 3 presents the results against attack methods sampled from MIX-JAIL. Our analysis yields two findings. First, intervening at the shallow and middle layers leads to a significant degradation in defense, revealing jailbreak semantics and paths have not ye fully formed in these layers. Second, JPU_{last} achieves closest performance to the full method, showing that disrupting the harmful aggregation of jailbreak paths and guiding them toward safe regions plays a vital role in JPU. However, it neglects signals propagated from intermediate layers, thereby narrowing the scope of defense and resulting in compromised performance.

Impact of Path Sparsity To examine the impact of path sparsity (defined as the top-$p\%$ masked neurons) on JPU, we evaluate its performance on Mix-JailExpert across $p \in \{0.01, 0.03, 0.05, 0.1, 0.2, 1.0\}$. Figure 5 highlights a utility-safety trade-off: extremely small p compromises defense effectiveness, while excessively large p degrades utility. Therefore, we adopt $p = 0.05$ as our standard configuration, as it strikes the optimal balance in our experiments.

5.4 Necessity of On-Policy Mining Strategy

To empirically validate the essential necessity of the on-policy mining for capturing dynamic attack features, we conduct ablation studies comparing training strategies: Static (Off-Policy) and Dynamic (On-Policy). Specifically, the static strategy relies on a fixed buffer without on-policy attack buffer mining component, while the dynamic strategy corresponds to the full JPU framework.

To ensure comprehensive robustness against variations in data scale and diversity, we evaluate both strategies under three progressive buffer initialization settings: Type $\in$ {A, A+B, A+B+C (full)}. We assess the overall defense performance against three representative attack methods (M_1-M_3) from MIX-JAIL dataset on Llama-2-7B-Chat. More information is detailed in C. As illustrated in Figure 6, the dynamic strategy consistently achieves a substantially lower ASR than the static baseline across all buffer settings. Even with the full diversity buffer, the static variant remains inferior to the on-policy strategy, indicating that simply scaling static data is insufficient for addressing dynamic threats. Instead, achieving an effective defense performance depends on on-policy adversarial mining, which actively identifies and mitigates evolving vulnerabilities that static datasets fail to capture.

6 Conclusion

In this paper, we introduce JPU, a novel framework designed to rectify dynamic jailbreak paths via on-policy rectification. Our research reveals a critical limitation in existing static unlearning defenses: adaptive attacks can activate the non-erased parameters in intermediate layers and eventually align with harmful semantics paths to trigger unsafe outputs. Our experimental results demonstrate that JPU not only achieves superior defense performance against diverse and dynamic jailbreak

attacks but also effectively preserves the general utility of LLMs. Moreover, post-hoc mechanism analysis confirms that our method successfully rectifies dynamic jailbreak paths and redirects these malicious paths toward safety regions. We hope that our work provides valuable insights into the dynamics of jailbreak attacks and inspires future research on robust unlearning-based defenses.

7 Limitations

The current implementation of our Attack Buffer within JPU is primarily designed to encompass prompt-based injection strategies. While JPU exhibits strong generalization capabilities against decoding-based attacks, exploring the integration of attack vectors beyond the prompt space—such as parameter-level manipulation or optimization-based attacks—could potentially yield even more robust defense performance. Integrating a broader spectrum of adversarial methodologies remains a promising direction to provide deeper insights and guidance for the design of future safety alignment strategies.

8 Acknowledgments

This work was supported in part by the National Key Research and Development Program under Grant (No. 2024YFB4506200); in part by the National Natural Science Foundation of China under Grant (No. 62421002); in part by the Science and Technology Innovation Program of Hunan Province under Grant (No. 2024RC1048); in part by the Innovation Research Foundation of NUDT under Grant (No. 24-ZZCX-JDZ-07); and in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118).

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2025. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42. IEEE.

Jiachi Chen, Qingyuan Zhong, Yanlin Wang, Kaiwen Ning, Yongkun Liu, Zenan Xu, Zhe Zhao, Ting Chen, and Zibin Zheng. 2024. Rmcbench: Benchmarking large language models' resistance to malicious code. In *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, pages 995–1006.

Junjie Chu, Yugeng Liu, Ziqing Yang, Xinyue Shen, Michael Backes, and Yang Zhang. 2024. Comprehensive assessment of jailbreak attacks against llms. *CoRR*.

Boyi Deng, Wenjie Wang, Fuli Feng, Yang Deng, Qifan Wang, and Xiangnan He. 2024. Attack prompt generation for red teaming and defending large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 2024. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2136–2153.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf a comprehensive practical alignment recipe of iterative preference learning. *Transactions on Machine Learning Research*, 2024.

Moussa Koulako Bala Doumbouya, Ananjan Nandi, Gabriel Poesia, Davide Ghilardi, Anna Goldie, Federico Bianchi, Dan Jurafsky, and Christopher D Manning. 2024. h4rm3l: A dynamic benchmark of composable jailbreak attacks for llm safety assessment. *CoRR*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jianhua Hu, Huixiang Gao, Qing Yuan, and Ganchang Shi. 2024. Dynamic content generation in large language models with real-time constraints.

Peichun Hua, Hao Li, Shanghao Shi, Zhiyuan Yu, and Ning Zhang. 2025. Rethinking jailbreak detection of large vision language models with representational contrastive scoring. *Preprint*, arXiv:2512.12069.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2024. Catastrophic jailbreak

of open-source llms via exploiting generation. In *The Twelfth International Conference on Learning Representations*.

Jinghan Jia, Jiancheng Liu, Yihua Zhang, Parikshit Ram, Nathalie Baracaldo, and Sijia Liu. 2024. Wagle: Strategic weight attribution for effective and modular unlearning in large language models. *Advances in Neural Information Processing Systems*, 37:55620–55646.

Masahiro Kaneko, Danushka Bollegala, and Naoaki Okazaki. 2022. Debiasing isn't enough!–on the effectiveness of debiasing mlms and their social biases in downstream tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1299–1310.

Junxian Li, Beining Xu, Simin Chen, Jiatong Li, Jingdi Lei, Haodong Zhao, and Di Zhang. 2025a. Iag: Input-aware backdoor attack on vlm-based visual grounding. *arXiv preprint arXiv:2508.09456*.

Kunhao Li, Wenhao Li, Di Wu, Lei Yang, Jun Bai, Ju Jia, and Jason Xue. 2025b. Cross-modal unlearning via influential neuron path editing in multimodal large language models. *arXiv preprint arXiv:2511.06793*.

Yu Li, Han Jiang, and Zhihua Wei. 2025c. DeTAM: Defending LLMs against jailbreak attacks via targeted attention modification. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Zeyi Liao and Huan Sun. 2024. Amplegcg: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed llms. *arXiv preprint arXiv:2404.07921*.

Xiaogeng Liu, Peiran Li, Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. 2024a. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms. *arXiv preprint arXiv:2410.05295*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024b. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang, and Cen Chen. 2024. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. *CoRR*.

Huijie Lv, Xiao Wang, Yuansen Zhang, Caishuang Huang, Shihan Dou, Junjie Ye, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Codechameleon: Personalized encryption framework for jailbreaking large language models. *arXiv preprint arXiv:2402.16717*.

Xiaoxu Ma, Xiangbo Zhang, and Zhenyu Weng. 2026. Stable and explainable personality trait evaluation in large language models with internal activations. *Preprint*, arXiv:2601.09833.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

OpenAI. 2023. ChatGPT, https://openai.com/chatgpt.

Sandeep Phanireddy. 2025. Llm security and guardrail defense techniques in web applications. *International Journal of Emerging Trends in Computer Science and Information Technology*, pages 221–224.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *Transactions on Machine Learning Research*.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5377–5400.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Zesheng Shi, Yucheng Zhou, Jing Li, Yuxin Jin, Yu Li, Daojing He, Fangming Liu, Saleh Alharbi, Jun Yu, and Min Zhang. 2025. Safety alignment via constrained knowledge unlearning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25515–25529.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Cheng Wang, Zeming Wei, Qin Liu, and Muhao Chen. 2025. False sense of security: Why probing-based malicious input detection fails to generalize. *arXiv preprint arXiv:2509.03888*.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Zixu James Zhu, Xiang-Bo Mao, Sitaram Asur, and Na Claire Cheng. 2024. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *CoRR*.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Proceedings of the 41st International Conference on Machine Learning*, pages 52588–52610.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. 2024. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 507–518.

Xiqiao Xiong, Ouxiang Li, Zhuo Liu, Moxin Li, Wentao Shi, Fengbin Zhu, Qifan Wang, and Fuli Feng. 2025. Trojail: Trajectory-level optimization for multi-turn large language model jailbreaks with process rewards. *arXiv preprint arXiv:2512.07761*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5587–5605.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Hangfan Zhang, Zhimeng Guo, Huaisheng Zhu, Bochuan Cao, Lu Lin, Jinyuan Jia, Jinghui Chen, and Dinghao Wu. 2024a. Jailbreak open-sourced large language models via enforced decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5475–5493.

Zhexin Zhang, Junxiao Yang, Pei Ke, Shiyao Cui, Chujie Zheng, Hongning Wang, and Minlie Huang. 2024b. Safe unlearning: A surprisingly effective and generalizable solution to defend against jailbreak attacks. *CoRR*.

Wei Zhao, Zhe Li, Yige Li, Ye Zhang, and Jun Sun. 2024a. Defending large language models against jailbreak attacks via layer-specific editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5094–5109.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Shuaiqiang Wang, Chong Meng, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2024b. Improving the robustness of large language models via consistency alignment. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8931–8941.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. 2024. Improving alignment and robustness with circuit breakers. *Advances in Neural Information Processing Systems*, 37:83345–83373.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Details of Baselines

In this section, we provide detailed descriptions of the baseline defense methods employed in our comparative experiments. These baselines encompass two primary categories: Unlearning-based defenses and Representation-based defenses.

A.1 Unlearning-based Defenses

- **Eraser** (Lu et al., 2024): It operates on the principle of Gradient Ascent. Specifically, Eraser maximizes the loss on harmful queries to decrease the likelihood of generating hazardous tokens, while simultaneously employing a retain loss on general domain datasets to prevent catastrophic forgetting. The optimization objective is to push the model's

distribution away from the harmful target sequences, erasing the knowledge required to answer malicious queries.

- **Safe Unlearning** (Zhang et al., 2024b): It treats safety training as a negative preference optimization problem. It constructs pairs of harmful query and response and minimizes the probability of generating the harmful response while steering the model towards a safe distribution or a specific refusal target. It emphasizes minimizing the side effects of unlearning, aiming to maintain the model's helpfulness on benign instructions while removing unsafe behaviors.

- **CKU (Confused Knowledge Unlearning)** (Shi et al., 2025): It represents a state-of-the-art approach to knowledge unlearning. Specifically, CKU first identifies safety-critical neurons in the multi-layer perceptron (MLP) components of LLMs that are strongly associated with safeguard knowledge. It then performs targeted unlearning by selectively modifying except these identified neurons, aiming to suppress unsafe behaviors while preserving the model's safety.

A.2 Representation-based Defenses

- **RSFT** (Deng et al., 2024): It implements an iterative fine-tuning strategy designed to harden Large Language Models (LLMs) against adversarial prompts. By simulating dynamic interactions between the attacker and the target model, RSFT progressively identifies safety vulnerabilities and refines the model's refusal boundaries, significantly enhancing its resilience to complex, multi-turn harmful instruction attacks.

- **Circuit Breaker (Representation Engineering)** (Zou et al., 2024): This is a representation-level defense that utilizes "circuit breakers" to reroute harmful internal activations. By intervening at the hidden state level rather than the output level, it provides a generalized defense against diverse attack vectors while preserving the model's original utility on benign tasks.

B Preparation of Experiments

B.1 Construction of MIX-JAIL

In this section, we present the construction details of our proposed challenging jailbreak dataset, **MIX-JAIL**. Specifically, we first collect six representative jailbreak attack methods that have emerged recently, covering diverse attack paradigms. These include Type A attacks such as CodeChameleon (4 jailbreak prompts); Type B attacks including GPTFuzzer (481 jailbreak prompts), ReNeLLM (475 jailbreak prompts), AutoDAN-Turbo (9 jailbreak prompts), and JailBroken (24 jailbreak prompts); as well as Type C attacks such as AmpleGCG (488 adversarial suffixes).

We then apply all collected jailbreak prompts and adversarial suffixes to the AdvB-Short dataset. After data filtering and separation, we construct a mixed jailbreak dataset consisting of approximately 70,000 jailbreak instances, which we refer to as **MIX-JAIL**.

B.2 Details of Attack Buffer Construction

The adversarial buffer $\mathcal{B}$ employed in our method integrates a diverse spectrum of existing prompt-based jailbreak methodologies. To ensure comprehensive coverage of the attack surface, we categorize these methods into three distinct types:

- **Type A (Code-based Encapsulation)**: Attacks that utilize coding scenarios or nested formats to bypass safety filters, represented by CodeChameleon (Lv et al., 2024) and RmcBench (Chen et al., 2024).

- **Type B (Mutation-based Optimization)**: Methods that employ fuzzing or evolutionary strategies to generate complex, evolving attack templates. Representative methods include GPTFuzzer (Yu et al., 2023), ReNeLLM (Ding et al., 2024), and AutoDAN-Turbo (Liu et al., 2024a).

- **Type C (Adversarial Suffix Optimization)**: Approaches that optimize fixed adversarial character sequences (suffixes) to induce affirmative responses, such as GCG (Zou et al., 2023) and AmpleGCG (Liao and Sun, 2024).

Buffer Instantiation. The collection process for our 200-sample buffer follows a protocol similar to the construction of the MIX-JAIL benchmark.

Specifically, we initialize the buffer by curating distinct jailbreak templates and mutation strategies from the aforementioned categories. We collect 66 templates for Type A (from CodeChameleon and RmcBench), 66 seeds for Type B (aggregated from GPTFuzzer, ReNeLLM, and AutoDAN-Turbo), and 66 suffixes for Type C (from AmpleGCG). Finally, these curated seeds are instantiated by applying them to harmful intents sampled from the AdvB-Short dataset, thereby forming the complete adversarial queries for the buffer $\mathcal{B}$.

B.3 Hyperparameter Settings

Regarding the critical coefficients in our joint unlearning objective, we empirically set the representation alignment weight $\beta = 4.0$ and the utility preservation weight $\lambda = 1.0$. Specifically, the value of $\beta = 4.0$ is chosen to prioritize the rectification of internal adversarial circuitry, enforcing a strong alignment between the model's hidden states and the safety direction. The utility coefficient $\lambda = 1.0$ is selected to maintain a balanced trade-off, providing sufficient regularization to preserve general language capabilities without overly constraining the plasticity required for unlearning jailbreak paths. These hyperparameters were determined based on a grid search on a held-out validation set.

C On-Policy Mining Ablation Details

We selected `codeChameleon`, `AutoDAN-Turbo`, and `GPTFuzzer` from the MIX-JAIL attack dataset to represent models M_1, M_2, M_3, respectively. Subsequently, we conducted ablation experiments on the On-Policy module.

Algorithm 1 Training Procedure of JPU

Require: Initial LLM $\mathcal{M}_\theta$; Initial Jailbreak Buffer $\mathcal{B}$ (containing templates & mutation strategies); Utility Dataset $\mathcal{D}_r$; Reference Safety Anchor H_{safe}; Refusal Target y_{ref} (e.g., "I cannot assist"); Semantic Sink Node y_{sink} (e.g., token "Sure"). Jailbreak threshold τ; Masking ratio p; Loss weights β, λ; Learning rate η.

Output: Defended Model $\mathcal{M}_\theta^*$.

while not converged **do**

  // Stage 1: On-Policy Attack Buffer Mining

  Sample a batch of seed prompts S_{seed} from Buffer $\mathcal{B}$.

  Compute refusal loss $\mathcal{L}_{\text{ref}}(x)$ on current $\mathcal{M}_\theta$ for $x \in S_{\text{seed}}$ w.r.t y_{ref}.

  Identify **Parents** (High-loss samples): $\mathcal{P} \leftarrow \{x \in S_{\text{seed}} \mid \mathcal{L}_{\text{ref}}(x) > \tau\}$.

  Discard low-loss samples $S_{\text{seed}} \setminus \mathcal{P}$.

  Generate **Offspring** $\mathcal{O}$ by applying homologous mutations to $\mathcal{P}$.

  Aggregate On-Policy Adversarial Batch: $\mathcal{B}_t \leftarrow \mathcal{P} \cup \mathcal{O}$.

  Update Buffer $\mathcal{B}$ with new effective samples from $\mathcal{B}_t$.

  // Stage 2: Jailbreak Path Identification

  For $\mathcal{B}_t$, target the semantic sink node y_{sink} at the last layer.

  Compute **Jailbreak Flow Score** via first-order Taylor approximation:

$$\text{Flow}_{\text{jb}}^{(l,i)} = |W^{(l,i)}| \cdot |A^{(l,i)} \odot \nabla A^{(l,i)}|$$

  Compute **Utility Flow Score** $\text{Flow}_{\text{util}}$ on a batch from $\mathcal{D}_r$.

  Calculate Differential Flow: $S^{(l,i)} = \text{Flow}_{\text{jb}}^{(l,i)} - \text{Flow}_{\text{util}}^{(l,i)}$.

  Generate **Sparse Mask** $\mathcal{M}_t$: Set top-$p\%$ connections with highest $S^{(l,i)}$ to 1, others to 0.

  // Stage 3: Constraint Path Rectification

  Compute Rectification Loss on $\mathcal{B}_t$:

$$\mathcal{L}_{\text{rect}} = \mathbb{E}_{x \in \mathcal{B}_t}[\mathcal{L}_{\text{h}}(x) + \beta \mathcal{L}_{\text{s}}(H_{\text{curr}}, H_{\text{safe}})]$$

  Compute Utility Loss on $\mathcal{D}_r$: $\mathcal{L}_{\text{util}} = \mathbb{E}_{x \in \mathcal{D}_r}[\mathcal{L}_{\text{u}}(x)]$.

  Compute Gradients and Apply Soft Update:

$$\theta \leftarrow \theta - \eta \left((\nabla_\theta \mathcal{L}_{\text{rect}} \odot \mathcal{M}_t) + \lambda \nabla_\theta \mathcal{L}_{\text{util}} \right)$$

end whilereturn $\mathcal{M}_\theta$