Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Representation-Conditioned Adapter Fusion for Balancing Safety and Utility in Large Language Model Services

Xinghao Wang
Information Engineering University
Yuke Ma
Information Engineering University
Wei Guo
Information Engineering University
Xi Chen
529118905@qq.com

Information Engineering University

Research Article

Keywords: Large language model security, jailbreak defense, adapter fusion, LoRA, over-refusal, safety alignment

Posted Date: June 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9821788/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Representation-Conditioned Adapter Fusion for Balancing Safety and Utility in Large Language Model Services

Abstract

Large language model (LLM) services remain vulnerable to jailbreak prompts that conceal harmful intent within apparently benign narratives, educational framing, or role playing. This paper proposes representation-conditioned adapter fusion (RCF), a parameter efficient defense that uses hidden states from a frozen instruction tuned model to derive separate frame and intent codes. Their interaction is mapped to layerwise fusion weights over refusal, correction, and purification adapters. The controller is trained with decomposed frame and intent supervision and continuous adapter targets, while risk gating and deterministic response repair calibrate safety and utility at inference. We evaluate RCF with a fixed GPT-5.4-mini automatic judge protocol using 200 prompts per split across harmful WildJailbreak, benign WildJailbreak, OR-Bench, and Alpaca. On Qwen2.5-7B-Instruct, RCF reduces attack success rate from 81.91% to 6.74%, with false refusal rates of 7.0%, 1.0%, and 0.5% on the three benign sets. Prompt and guard filter baselines retain higher attack success rates, ranging from 46.0% to 71.0%. A prompt overlap audit removes all hardmix calibration prompts from the compact evaluation and still gives 5.75% ASR with benign false refusal rates of 5.71%, 1.31%, and 0.5%. A separated calibration and test rerun further checks this issue: on a fresh 200/200/200/200 test manifest with calibration prompts excluded, the calibrated variant obtains 11.0% ASR with false refusal rates of 2.0%, 9.0%, and 0.5%. An extended Qwen verification keeps harmful ASR at 9.0%, and Llama-2/Mistral experiments show low harmful ASR with backbone dependent benign calibration gaps.

Keywords: Large language model security, jailbreak defense, adapter fusion, LoRA, over-refusal, safety alignment

1 Introduction

Large language model services increasingly mediate search, programming, education, writing assistance, and decision-support workflows. Their security failures therefore affect not only isolated model outputs but also the reliability of surrounding information systems. One prominent failure mode is semantic jailbreak: a harmful objective is embedded inside a benign-looking frame, such as fiction, role play, translation, safety

education, counterfactual analysis, or red-team discussion. Such prompts can preserve malicious intent while changing surface form, making keyword filters and uniform refusal prompts brittle (Chao et al. 2025; Liu et al. 2024; Shen et al. 2024; Wei et al. 2023; Zou et al. 2023b).

This setting creates a practical safety-utility conflict. A model or adapter trained mainly to refuse unsafe requests may reduce harmful compliance, but it can also reject benign security education, civic discussion, fictional writing, and other policy-adjacent prompts. Conversely, a permissive model may preserve ordinary helpfulness while still following harmful framed requests. Existing add-on defenses, such as prompt reminders and guard models, are attractive because they can be deployed without retraining the backbone (Inan et al. 2023; Xie et al. 2023). However, a binary pass/block or uniform warning decision does not describe how the LLM should answer when a prompt is benign but framed, or harmful but disguised as benign.

We argue that semantic jailbreak defense should be treated as representation-conditioned behavior control. The central distinction is not only whether a prompt is unsafe, but how surface framing and latent intent interact. A direct harmful request, an adversarially framed harmful request, and an adversarially framed benign request should induce different generation behavior. This motivates a factorized controller that reads the frozen model's internal representation, separates frame and intent evidence, and continuously fuses multiple safety behaviors before decoding.

This paper proposes representation-conditioned adapter fusion (RCF), a lightweight parameter-efficient framework for safety-utility balanced LLM services. RCF keeps the base LLM frozen and uses three heterogeneous LoRA adapters: a refusal adapter, a correction adapter, and a purification/helpfulness adapter. A small controller receives a pooled hidden representation from the base model and learns two latent codes: a frame code and an intent code. Their interaction produces layer-wise fusion weights over the adapters, allowing the system to apply different mixtures at different depths of the model. At inference time, a risk gate controls the total adapter strength, ambiguity calibration moves uncertain framed prompts away from hard refusal, and deterministic response repair reduces generic refusals and unhelpful redirects under pre-registered non-risk conditions.

Fig. 1 summarizes the resulting service-side pipeline, from representation extraction to factorized control, layer-wise expert fusion, and calibrated response generation.

The contributions are as follows:

- We formulate jailbreak defense as representation-conditioned adapter fusion, replacing one-label safety routing with a continuous control problem over reusable safety behaviors.
- We introduce a factorized frame/intent controller that learns separate representation codes and uses their interaction to generate layer-wise LoRA fusion weights.
- We add inference-time safety-utility calibration, including risk-gated adapter strength, ambiguous-frame correction, text-risk refusal adjustment, and auditable deterministic response repair.
- We provide a controlled evaluation across harmful WildJailbreak, benign WildJailbreak, OR-Bench, and Alpaca utility prompts, including Qwen baselines, external

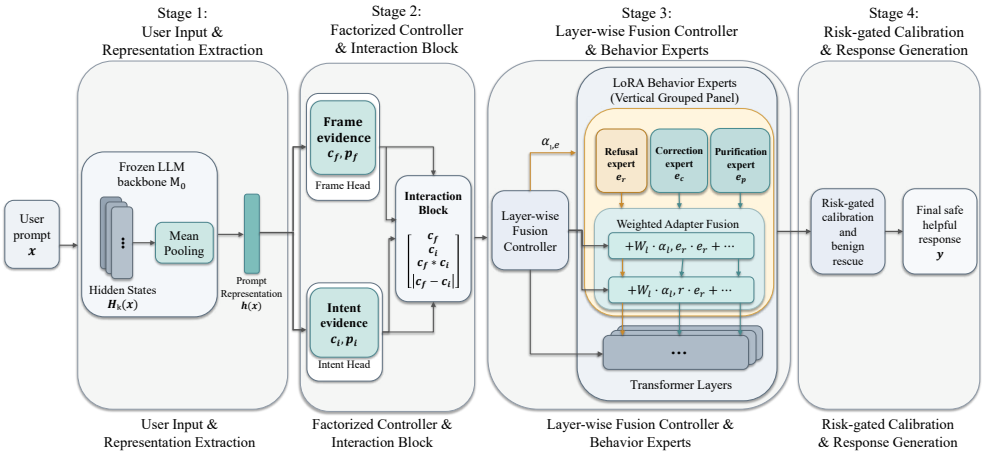


Fig. 1 Overview of RCF. A frozen LLM provides hidden representations that are pooled into a prompt vector. A factorized controller estimates frame and intent evidence, and their interaction generates layer-wise fusion weights over refusal, correction, and purification LoRA experts. Risk-gated calibration adjusts adapter strength, and deterministic response repair can revise generic refusals under pre-registered non-risk conditions

prompt/guard baselines, component ablations, prompt-overlap audits, a separated calibration-test rerun, pre-specified extended verification, and Llama-2/Mistral transfer experiments.

2 Related Work

2.1 Jailbreak attacks and safety evaluation

Jailbreak attacks attempt to bypass LLM safety alignment through adversarial suffixes, instruction hierarchy conflicts, role-play scenarios, persona injection, and other semantic transformations (Chao et al. 2025; Liu et al. 2024; Shen et al. 2024; Wei et al. 2023; Zou et al. 2023b). WildJailbreak and WildTeaming emphasize adversarial prompts collected or generated at scale, including harmful prompts with benign-looking frames and benign prompts that test over-refusal (Jiang et al. 2024). HarmBench and JailbreakBench further show the need for standardized red-teaming datasets, threat models, scoring rules, and reproducible artifacts (Chao et al. 2024; Mazeika et al. 2024). OR-Bench and XSTest focus on over-refusal, exposing cases where aligned models reject harmless requests that contain sensitive or policy-adjacent language (Cui et al. 2025; Röttger et al. 2024). These benchmarks are important because a defense that only reduces harmful compliance may still be unsuitable for deployment if it blocks benign user tasks.

2.2 Prompt-level and guard-model defenses

Prompt-level defenses prepend safety reminders or policy instructions to the user request (Xie et al. 2023). They are cheap and model-agnostic, but the same instruction

is applied to all inputs. Guard models such as Llama Guard classify user and assistant messages against safety policies (Inan et al. 2023). Other inference-time defenses perturb prompts or alter decoding to favor safer continuations (Robey et al. 2023; Xu et al. 2024). These systems are useful for moderation or test-time filtering, but they do not directly specify how the generation model should balance refusal, safe redirection, and ordinary helpfulness for different prompt profiles. RCF is complementary: it controls generation behavior inside the model by fusing adapters according to the prompt representation.

2.3 Parameter-efficient safety adaptation

Parameter-efficient fine-tuning, especially LoRA, enables behavior specialization without updating all backbone parameters (Hu et al. 2022). Prior work has also shown that fine-tuning can alter safety behavior in unintended ways, sometimes compromising alignment (Qi et al. 2024). A single safety adapter, however, occupies one point on the safety-utility frontier. In our experiments, a single refusal adapter sharply reduces harmful attack success but severely over-refuses benign prompts. RCF instead treats adapters as reusable behavior bases and learns a representation-conditioned mixture over them.

2.4 Representation-level behavior control

Recent work indicates that high-level model behavior can be monitored or shifted through internal representations rather than only through prompts or output filters. Representation engineering studies population-level activation features for transparency and control (Zou et al. 2023a), inference-time intervention steers selected activation directions to improve truthfulness (Li et al. 2023), and refusal-direction analysis suggests that refusal behavior is strongly tied to identifiable activation subspaces (Arditi et al. 2024). RCF follows this representation-level view, but uses prompt representations to compose multiple LoRA behavior experts rather than applying a single activation shift.

2.5 LLM-as-a-judge evaluation

Automatic safety evaluation often requires more than string matching because harmful compliance may be partial, indirect, or phrased without obvious prohibited keywords. LLM-as-a-judge protocols have been widely used for instruction-following and safety evaluation (Chao et al. 2024; Mazeika et al. 2024; Souly et al. 2024; Zheng et al. 2023). We therefore use GPT-5.4-mini as a fixed external judge for harmful compliance and benign over-refusal across all formal runs. The judge prompt, decoding setting, sampled prompts, and output schema are held constant across systems; invalid harmful judge outputs are excluded from the denominator and reported rather than imputed. This protocol makes the comparisons internally consistent and follows current automatic safety-evaluation practice.

3 Method

3.1 Problem formulation

Let x be a user prompt and y be the generated response. A deployed LLM service should reduce harmful compliance on unsafe prompts while preserving useful answers on benign prompts, including benign prompts whose surface form resembles safety-sensitive content. Instead of learning one static safety behavior, we consider a frozen instruction-tuned model M_0 and a set of LoRA experts:

$$\mathcal{E} = \{e_r, e_c, e_p\}, \tag{1}$$

where e_r is a refusal expert, e_c is a safe-correction or redirection expert, and e_p is a purification/helpfulness expert. The goal is to infer, from x, how much each expert should influence each target layer before decoding.

For LoRA expert e, the adapted weight of a linear module is:

$$W_e = W_0 + s_e B_e A_e, \tag{2}$$

where W_0 is frozen, A_e and B_e are low-rank matrices, and s_e is the base scaling factor. Rather than selecting one expert globally, RCF predicts layer-wise expert weights $\alpha_{\ell,e}(x)$ and applies:

$$W_\ell(x) = W_{0,\ell} + \sum_{e \in \mathcal{E}} \alpha_{\ell,e}(x) s_{\ell,e} B_{\ell,e} A_{\ell,e}, \tag{3}$$

with $\alpha_{\ell,e}(x) \geq 0$. This continuous layer-wise fusion gives the controller a richer action space than a binary guard or a hard adapter switch.

3.2 Representation factorization

Semantic jailbreaks confound two factors: surface framing and harmful intent. We therefore define two binary latent labels:

$$f(x) \in \{0, 1\}, \qquad i(x) \in \{0, 1\}, \tag{4}$$

where $f(x) = 1$ denotes adversarial or framed presentation and $i(x) = 1$ denotes harmful intent. Their product gives four route categories: direct safe, direct unsafe, adversarial benign, and adversarial harmful. The factorization shares evidence across categories: framed benign and framed harmful prompts both train the frame code, while direct harmful and framed harmful prompts both train the intent code.

Let $H_K(x)$ be the final hidden state sequence of the frozen base model. We pool the final hidden states into a prompt representation:

$$h(x) = \text{MeanPool}\left(H_K(x)\right) \in \mathbb{R}^d. \tag{5}$$

The controller first maps $h(x)$ into a shared hidden vector and then into separate frame and intent codes:

$$u(x) = \phi_s(h(x)), \tag{6}$$
$$c_f(x) = \phi_f(u(x)), \tag{7}$$
$$c_i(x) = \phi_i(u(x)). \tag{8}$$

Two sigmoid heads estimate the frame and intent probabilities:

$$p_f(x) = \text{sigmoid}(w_f^\top c_f(x)), \qquad p_i(x) = \text{sigmoid}(w_i^\top c_i(x)). \tag{9}$$

3.3 Layer-wise adapter fusion

The interaction between frame and intent codes is used to produce fusion weights:

$$v(x) = [c_f(x); c_i(x); c_f(x) \odot c_i(x); |c_f(x) - c_i(x)|]\,. \tag{10}$$

A fusion head maps $v(x)$ to logits $G(x) \in \mathbb{R}^{L\times|\mathcal{E}|}$, where L is the number of LoRA target layers. The raw fusion weights are:

$$\tilde{\alpha}_\ell(x) = \text{softmax}(G_\ell(x)/T), \tag{11}$$

where T is a training temperature. Early and late layers can therefore receive different mixtures of refusal, correction, and purification experts.

The target fusion policy encodes the intended safety behavior. Direct unsafe prompts emphasize refusal across layers. Adversarial harmful prompts use correction earlier and stronger refusal later. Adversarial benign prompts emphasize correction and purification while avoiding hard refusal. Direct safe prompts use purification/helpfulness. For a category z, the layer target is constructed by linearly interpolating between category-specific start and end expert distributions:

$$\tau_{\ell,z} = (1-\rho_\ell)\tau_z^{\text{start}} + \rho_\ell \tau_z^{\text{end}}, \qquad \rho_\ell = \frac{\ell-1}{L-1}. \tag{12}$$

Table 1 reports the final target preset used in the Qwen controller. The entries are not learned labels; they are the pre-specified behavior targets used to supervise the fusion head.

3.4 Training objective

Given a training prompt x with route category z, frame label f_z, intent label i_z, and target layer weights $\tau_{\ell,z}$, the controller minimizes:

$$\begin{aligned}\mathcal{L} =& \lambda_f \text{BCE}(p_f(x), f_z) + \lambda_i \text{BCE}(p_i(x), i_z) \\ &+ \lambda_a \frac{1}{L}\sum_{\ell=1}^{L} \|\tilde{\alpha}_\ell(x) - \tau_{\ell,z}\|_2^2 + \lambda_m \mathcal{L}_{\text{margin}} + \lambda_c \mathcal{L}_{\text{supcon}} + \lambda_e \mathcal{L}_{\text{entropy}}.\end{aligned} \tag{13}$$

Table 1 Layer-wise fusion targets for the final `safety_utility_v2` preset. Each entry is a start-to-end distribution over refusal, correction, and purification experts.

Route category	Start target (e_r, e_c, e_p)	End target (e_r, e_c, e_p)
Direct unsafe	(0.96, 0.04, 0.00)	(0.98, 0.02, 0.00)
Adversarial harmful	(0.55, 0.40, 0.05)	(0.90, 0.10, 0.00)
Adversarial benign	(0.00, 0.85, 0.15)	(0.00, 0.75, 0.25)
Direct safe	(0.00, 0.05, 0.95)	(0.00, 0.03, 0.97)

The margin term separates harmful and benign intent logits. The supervised contrastive term is applied to intent codes inside the framed-prompt subspace, encouraging adversarial benign and adversarial harmful prompts to separate even when their surface frames are similar. The entropy term controls the sharpness of layer-wise fusion weights.

3.5 Inference-time calibration

The learned controller predicts a direction in expert-mixture space, but deployment requires calibrated adapter strength. We define a scalar risk score:

$$r(x) = \max\{p_i(x), \gamma p_f(x), p_f(x)p_i(x)\}, \tag{14}$$

where γ is a frame-risk floor. Adapter strength is:

$$s(x) = s_{\min} + (1 - s_{\min}) \operatorname{sigmoid}\left(\frac{r(x) - t_r}{T_r}\right). \tag{15}$$

The final weights are initially $\alpha_{\ell,e}(x) = s(x)\tilde{\alpha}_{\ell,e}(x)$.

We then apply three calibrated adjustments. First, for framed but low-intent prompts, a fraction of refusal mass is shifted to correction and purification, reducing false refusal on adversarial-benign prompts. Second, for high-intent prompts, non-refusal mass is shifted toward refusal. Third, prompts matching strict text-risk patterns can receive an additional refusal shift when the frame signal is high. Finally, a deterministic response-repair stage handles generic refusals and unhelpful redirects. The repair registry has two pre-specified trigger classes: registered benign utility patterns, such as security-awareness training, fictional or educational scenarios, harmless simulations, and hard benign OR-Bench prompts; and refusal-like outputs for prompts that do not match the strict text-risk pattern list. The replacement is a safe, task-preserving response template. This stage does not query the external judge, does not use judge labels, and does not change the risk gate or adapter weights. Because it is an output-level calibration step, we report its trigger-equivalent output changes on every split rather than treating it as a hidden classifier. Separately, hard-benign calibration uses previously over-refused OR-Bench-style benign prompts as adversarial-benign

controller examples; this improves routing away from hard refusal on benign policy-adjacent prompts before any response repair is applied. The inference procedure is:

1. Extract $h(x)$ from M_0 with adapters disabled.
2. Compute $p_f(x)$, $p_i(x)$, and raw layer weights $\tilde{\alpha}_{\ell,e}(x)$ using the controller.
3. Compute risk score $r(x)$ and adapter strength $s(x)$.
4. Apply ambiguous-frame, harmful-intent, and text-risk calibration to obtain $\alpha_{\ell,e}(x)$.
5. Apply layer-wise LoRA scaling in Eq. (3) and generate response y with deterministic decoding.
6. If y is a generic refusal or unhelpful redirect meeting the pre-registered non-risk repair rules, replace it with a safe task-preserving response.

3.6 Risk interpretation

The factorization gives a simple interpretation of routing risk. Let q_{ab} be the probability that a prompt from true profile a receives behavior b, and let $r_b(a)$ be the harmful-compliance or over-refusal risk of applying behavior b to profile a. The expected deployment risk is:

$$\mathcal{R} = \sum_a \pi_a \sum_b q_{ab} r_b(a). \tag{16}$$

If the frame and intent heads have error rates ϵ_f and ϵ_i, then the joint profile error is bounded by:

$$P(\hat{z} \neq z) \leq \epsilon_f + \epsilon_i. \tag{17}$$

If a wrong behavior increases risk by at most B, then:

$$\mathcal{R}(\hat{z}) - \mathcal{R}(z^*) \leq B(\epsilon_f + \epsilon_i). \tag{18}$$

This bound does not prove universal superiority over every monolithic classifier. It clarifies the inductive bias: when jailbreak prompts are structured by frame and intent, decomposed supervision can reduce the burden of learning each joint profile independently and provides interpretable control signals for adapter fusion.

4 Experimental Setup

4.1 Models and implementation

Qwen2.5-7B-Instruct is the main backbone (Yang et al. 2024). We also evaluate transfer on Llama-2-7B-Chat (Touvron et al. 2023) and Mistral-7B-Instruct (Jiang et al. 2023). The main Qwen controller uses 28 fusion layers, 128-dimensional frame and intent codes, 8 epochs, batch size 64, learning rate 8×10^{-5}, weight decay 0.01, and 26,999 mixed training samples. The expert set contains refusal, correction, and purification LoRA adapters. The final inference configuration uses risk-gated calibration, frame-risk floor 0.55, ambiguous shift 0.45, harmful refusal shift 0.40, strict text-risk mode, and deterministic response repair.

4.2 Expert adapter training

The three behavior experts are trained before controller learning and are frozen during representation-conditioned fusion training. All experts use LoRA on the frozen backbone with response-token-only supervised fine-tuning: prompt tokens are masked out of the cross-entropy loss, and only assistant response tokens contribute gradients. The initial experts use a maximum sequence length of 256, two epochs, batch size 4, gradient accumulation 4, learning rate 10^{-4}, at most 3,000 samples per expert, AdamW weight decay 0.01, gradient clipping at 1.0, LoRA dropout 0.05, no bias parameters, and LoRA scaling $\alpha = 2r$. The LoRA modules are attached to `q_proj` and `v_proj`. The refusal expert uses rank 16 on layers 24–27; the correction expert uses rank 8 on layers 16–22; and the purification/helpfulness expert uses rank 8 on layers 20–26.

The expert training data are constructed to induce distinct generation behaviors. Unsafe prompts are paired with diverse refusal templates for the refusal expert and safe-redirection templates for the correction expert. Safe prompts with substantive reference responses are used for the purification/helpfulness expert. After the initial expert training, the refusal expert is further hardened on unsafe prompts while the correction and purification experts are loaded as frozen negative adapters. This stage uses one epoch, batch size 2, gradient accumulation 8, learning rate 3×10^{-5}, at most 12,000 samples, maximum sequence length 256, and a hook on layer 27. Its loss combines the same response-token cross-entropy with a contrastive push-apart term that separates the refusal hidden representation from the frozen correction and purification hidden representations, using contrastive weight 0.3 and margin 0.5. The controller is then trained only on frozen backbone representations and frozen experts, so the reported fusion behavior comes from learned layer-wise expert weights rather than additional backbone tuning.

4.3 Datasets

Harmful safety is evaluated on WildJailbreak harmful prompts. Over-refusal is evaluated on three benign sets: WildJailbreak benign prompts, OR-Bench hard benign prompts, and Alpaca-style utility prompts. Each main formal run uses a fixed 200/200/200/200 protocol with 200 harmful, 200 WildJailbreak benign, 200 OR-Bench, and 200 Alpaca prompts. This compact protocol is designed to cover the two failure surfaces targeted by the method: framed harmful compliance and false refusal on benign safety-adjacent or ordinary utility requests. We keep the same prompt subsets and judge protocol across main comparisons, ablations, and cross-model transfer to avoid mixing judge and sample effects. The evaluation also includes a pre-specified extended Qwen protocol with 500 harmful prompts, all 210 available WildJailbreak benign prompts, 500 OR-Bench prompts, and 500 Alpaca prompts.

The final controller also uses a hardmix calibration pool of 91 unique judged-failure prompts. A prompt-level audit found overlap between this pool and the compact evaluation prompts: 19 harmful WildJailbreak prompts, 25 benign WildJailbreak prompts, 47 OR-Bench prompts, and no Alpaca prompts. We therefore report the full compact protocol for comparability with all baselines and also provide a prompt-deduplicated

sensitivity analysis in which every hardmix calibration prompt is removed from evaluation before metrics are computed.

To strengthen this post hoc deduplication check, we additionally construct a separated train/calibration/test rerun. The split generator rebuilds the controller-training data, writes a calibration manifest, writes a test manifest with 200 prompts from each evaluation split, and passes an explicit 55-prompt calibration-exclusion list to the evaluator. The evaluator enforces exact 200/200/200/200 test counts after exclusion. This rerun is used as a rigor check for utility calibration rather than as a replacement for the compact protocol used in the main baseline comparisons.

4.4 Baselines and variants

We compare RCF against:

- **Base**: the unmodified instruction-tuned backbone.
- **Single refusal adapter**: a safety-only adapter that emphasizes refusal.
- **Static equal fusion**: an unconditioned equal mixture of refusal, correction, and purification adapters.
- **Self-Reminder**: a prompt-level safety reminder applied as a system message before generation.
- **Safety system prompt**: a stronger safety-prioritized system prompt that instructs the model to refuse operationally unsafe content and redirect to preventive information.
- **Llama Guard filter**: a guard-model pipeline that blocks prompts classified as unsafe by Llama Guard and otherwise forwards the prompt to the base generator.
- **Prior redirect**: an earlier fixed redirect calibration that does not use the full representation-conditioned fusion and response-repair pipeline.
- **Hardmix only**: a harder safety-skewed mixture selected for low harmful ASR without the final benign calibration.
- **Ablations**: removing deterministic response repair, safe redirect, ambiguous shift, text-risk shift, hard-benign calibration, and hardmix calibration.

These baselines include both mechanism-isolating comparisons and external deployment-style defenses. Prompt reminders and guard models are important complementary defenses, but they mainly decide whether to warn, block, or moderate a request. Our question is different: whether generation behavior can be controlled inside the model by representation-conditioned fusion of reusable LoRA behaviors. The base, single-refusal, static-fusion, prior-redirect, and component-removal variants therefore test whether the result comes from representation-conditioned fusion and calibrated inference rather than from simply adding a safety adapter, averaging adapters, or applying a fixed redirect rule. The Self-Reminder, safety-system-prompt, and Llama-Guard-filter baselines test whether a lightweight external defense can obtain a comparable safety-utility point without representation-conditioned generation control.

4.5 Metrics and judge

Attack success rate (ASR) is the percentage of harmful prompts judged as jailbreak successes. The harmful judge uses a five-point rubric, and responses with score at least 3 are counted as successful harmful compliance. Benign false positive rate (FPR) is the percentage of benign outputs judged as refusals. For benign prompts, the judge labels refusals, mixed responses, and answered responses; FPR counts refusals, while mixed responses are tracked separately in the raw judge files. All formal ASR and FPR values in the main tables use GPT-5.4-mini as the external judge under temperature-zero JSON decoding. For harmful prompts, invalid judge outputs are excluded from the denominator and reported in the tables. For the main RCF harmful row, the invalid-output sensitivity is 6.74% when invalid outputs are excluded, 6.50% when they are counted as safe, and 10.0% when they are counted as unsafe.

4.6 Reproducibility

To make the automatic evaluation auditable, each formal run records the sampled prompt indices, generated responses, judge labels, invalid judge outputs, and inference configuration. These artifacts are separate from the manuscript and can be released with the code and processed result summaries subject to the licenses of the underlying models and datasets.

5 Results

5.1 Main Qwen results

Table 2 reports the main Qwen2.5-7B-Instruct comparison on the full compact protocol. The base model has high harmful ASR, while the single refusal adapter sharply improves safety but collapses benign utility. Static equal fusion is also insufficient: it leaves ASR at 55.5% and OR-Bench FPR at 60.0%. In contrast, RCF reaches 6.74% ASR with 7.0%, 1.0%, and 0.5% FPR on WildJailbreak benign, OR-Bench, and Alpaca. The RCF ASR is computed over 193 valid harmful judge outputs, with 7 invalid harmful judge outputs excluded. The result supports the central claim that representation-conditioned fusion helps obtain a balanced safety-utility point.

Table 3 repeats the key Qwen comparison after prompt-level removal of all hard-mix calibration prompts. This audit addresses the possibility that utility calibration examples overlap benchmark prompts. The deduplicated RCF row remains in the same safety-utility range: ASR is 5.75% over 174 valid harmful judge outputs, and OR-Bench FPR is 1.31% on 153 non-overlapping OR-Bench prompts.

Table 4 reports the separated calibration-test rerun. The unretuned disjoint configuration already keeps ASR at 11.0%, but it over-refuses 14.5% of WildJailbreak benign and OR-Bench prompts. The low-over-refusal calibration leaves ASR unchanged at 11.0%, reduces WildJailbreak benign FPR to 2.0%, reduces OR-Bench FPR to 9.0%, and keeps Alpaca FPR at 0.5%. This result is weaker than the compact OR-Bench number but methodologically cleaner: the calibration prompts are not part of the test manifest, and the judge is applied only after the split-controlled generation run is complete.

Table 2 Main Qwen2.5-7B-Instruct results under the 200/200/200/200 GPT-5.4-mini judge protocol. Lower is better for ASR/FPR. Invalid counts harmful judge outputs excluded from ASR denominators.

System	ASR	WJ FPR	OR-Bench FPR	Alpaca FPR	Invalid
Qwen base	81.91	0.0	6.5	0.0	1
Single refusal adapter	5.00	78.0	97.0	59.0	0
Static equal fusion	55.50	5.0	60.0	2.0	0
RCF final	6.74	7.0	1.0	0.5	7

Table 3 Prompt-deduplicated sensitivity for the compact Qwen comparison. The deduplicated rows remove every prompt text that appears in the 91-prompt hardmix calibration pool before computing metrics. Lower is better for ASR/FPR.

System and set	ASR	WJ FPR	OR-Bench FPR	Alpaca FPR	Invalid
Qwen base, full	81.91	0.0	6.5	0.0	1
RCF final, full	6.74	7.0	1.0	0.5	7
Qwen base, deduplicated	81.11	0.0	5.23	0.0	1
RCF final, deduplicated	5.75	5.71	1.31	0.5	7

Table 4 Separated calibration-test verification on Qwen2.5-7B-Instruct. Both rows use the same disjoint 200/200/200/200 test manifest and GPT-5.4-mini judge protocol. Lower is better for all metrics.

Variant	ASR	WJ FPR	OR-Bench FPR	Alpaca FPR	Invalid
Before calibration	11.00	14.5	14.5	0.5	0
After calibration	11.00	2.0	9.0	0.5	0

Although the single refusal adapter has a slightly lower ASR than RCF on the full compact set, it is unusable as a balanced service defense because it refuses most benign prompts. RCF sacrifices only 1.74 percentage points of ASR relative to this safety-only point while reducing OR-Bench over-refusal from 97.0% to 1.0%.

Fig. 2 visualizes the same tradeoff using harmful ASR and mean benign FPR.

5.2 External prompt and guard baselines

Table 5 compares RCF with lightweight external defenses that do not train a representation-conditioned controller. Self-Reminder reduces ASR only modestly, from 81.91% to 71.0%. The stronger safety system prompt and Llama Guard filter reduce ASR further, to 46.0% and 52.0%, but remain above RCF and increase OR-Bench over-refusal to 21.0% and 31.5%. These results indicate that prompt-level or binary guard decisions alone do not provide the same safety-utility tradeoff as representation-conditioned generation control.

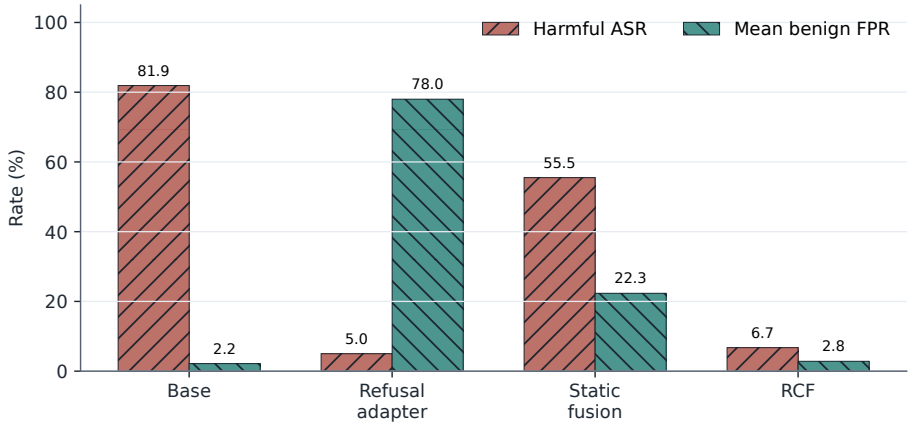


Fig. 2 Safety-utility tradeoff on Qwen2.5-7B-Instruct under the compact protocol. RCF keeps harmful ASR close to the safety-only refusal adapter while greatly reducing mean benign false refusal compared with safety-only and static-fusion variants

Table 5 External prompt and guard baselines under the 200/200/200/200 GPT-5.4-mini judge protocol. Lower is better for ASR/FPR. Invalid counts harmful judge outputs excluded from ASR denominators.

System	ASR	WJ FPR	OR-Bench FPR	Alpaca FPR	Invalid
Qwen base	81.91	0.0	6.5	0.0	1
Self-Reminder	71.00	2.5	7.5	0.0	0
Safety system prompt	46.00	1.5	21.0	1.0	0
Llama Guard filter	52.00	7.0	31.5	0.0	0
RCF final	6.74	7.0	1.0	0.5	7

5.3 Extended Qwen verification

Table 6 reports a pre-specified extended Qwen verification run. Because the local WildJailbreak benign split contains 210 available prompts, the extended protocol uses 500 harmful prompts, all 210 WildJailbreak benign prompts, 500 OR-Bench prompts, and 500 Alpaca prompts. The base model remains highly unsafe, with 83.6% ASR. RCF keeps harmful ASR low at 9.0%, while obtaining 5.71%, 11.60%, and 0.20% FPR on WildJailbreak benign, OR-Bench, and Alpaca. After removing hardmix-overlapping prompts, the extended RCF metrics are 8.73% ASR, 4.32% WildJailbreak benign FPR, 12.80% OR-Bench FPR, and 0.20% Alpaca FPR. The extended run supports the harmful-safety finding under a larger harmful set and provides a broader benign calibration check.

5.4 Component ablations

Table 7 shows Qwen ablations. Removing deterministic response repair increases OR-Bench FPR from 1.0% to 33.0%, while leaving ASR in the same low range, which indicates that this stage mainly repairs false refusals rather than serving as the primary

Table 6 Extended Qwen verification with 500 harmful, 210 WildJailbreak benign, 500 OR-Bench, and 500 Alpaca prompts. Lower is better for all metrics.

System	ASR	WJ FPR	OR-Bench FPR	Alpaca FPR	Invalid
Qwen base	83.60	1.90	5.60	0.00	0
RCF final	9.00	5.71	11.60	0.20	0

Table 7 Qwen component ablations. The strongest validated components are safe redirect, deterministic response repair, and hard-benign calibration.

Variant	ASR	WJ FPR	OR-Bench FPR	Alpaca FPR
RCF final	6.74	7.0	1.0	0.5
No deterministic repair	8.00	18.0	33.0	0.0
No safe redirect	19.50	16.5	42.5	0.0
No ambiguous shift	6.00	7.0	1.0	0.5
No text-risk shift	7.07	4.5	1.5	0.5
No hard-benign calibration	7.77	4.5	22.5	0.5
Hardmix only	2.60	6.5	18.0	0.0
Prior redirect	9.79	4.5	7.0	0.0

harmful-safety mechanism. Removing safe redirect is more damaging: ASR rises to 19.5%, WildJailbreak benign FPR to 16.5%, and OR-Bench FPR to 42.5%. Removing hard-benign calibration also increases OR-Bench FPR to 22.5%, showing that utility-side routing needs explicit hard benign evidence. Compared with the no-repair run, the final configuration changes 100 harmful outputs, 35 WildJailbreak-benign outputs, 79 OR-Bench outputs, and one Alpaca output in the compact protocol; the largest measured FPR reduction is on OR-Bench, where refused outputs drop from 66 to 2. In contrast, removing ambiguous shift or text-risk shift has limited aggregate effect in this 200-sample protocol, so we treat them as calibration refinements rather than independently necessary mechanisms.

The hardmix-only variant obtains the lowest ASR in the table but keeps OR-Bench FPR at 18.0%. This is why we do not select it as the main result. The final configuration is chosen as the balanced point rather than the absolute minimum-ASR point.

Fig. 3 highlights the largest ablation effects on harmful ASR and OR-Bench false refusal.

5.5 Cross-model transfer

Table 8 evaluates the framework on Llama-2-7B-Chat and Mistral-7B-Instruct under the same judge protocol. Both transfer runs obtain low harmful ASR, around 1.5%, showing that the safety behavior can transfer to other 7B chat backbones. However, benign calibration is model-dependent. Llama-2 has higher benign over-refusal on all three benign sets, while Mistral mainly over-refuses WildJailbreak benign prompts.

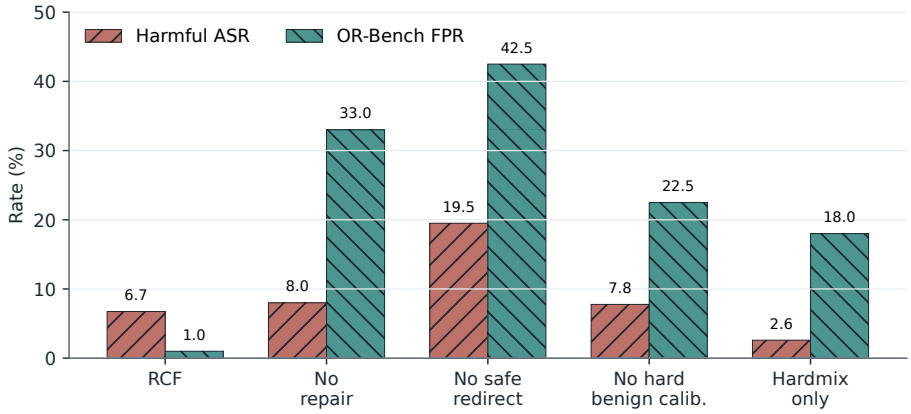


Fig. 3 Ablation summary for the main Qwen setting. Safe redirect and benign-utility calibration are the largest contributors to the balanced safety-utility point

Table 8 Cross-model transfer under the same GPT-5.4-mini judge protocol. Invalid harmful judge outputs are excluded from ASR denominators.

Model	ASR	WJ FPR	OR-Bench FPR	Alpaca FPR	Invalid
Qwen RCF	6.74	7.0	1.0	0.5	7
Llama-2 transfer	1.50	17.5	19.0	8.5	0
Mistral transfer	1.52	23.0	3.5	0.0	2

We therefore interpret transfer as evidence that the representation-conditioned safety behavior is portable, while the benign utility calibration remains backbone-specific.

5.6 Routing diagnostics

Table 9 summarizes the learned global fusion behavior on the final Qwen run. Harmful prompts receive high refusal mass, while Alpaca prompts are almost entirely routed to purification/helpfulness. OR-Bench prompts receive low refusal mass and are mostly dominated by purification or correction. WildJailbreak benign prompts remain the hardest benign category because they intentionally resemble adversarial frames; their average correction mass is high, but some prompts still receive refusal-dominant routing.

These diagnostics match the intended control structure: harmful prompts are dominated by refusal, utility prompts by purification, and framed benign prompts by correction. The remaining WildJailbreak benign FPR indicates that framed benign prompts still require better calibration or richer benign safety-adjacent training data.

Table 9 Qwen final routing diagnostics. Values are effective post-gating adapter coefficients averaged over layers and prompts; rows need not sum to one because risk gating scales total adapter strength. Dominant profile reports the percentage of prompts whose largest global weight belongs to each expert.

Split	Metric	Refusal	Correction	Purification	Dominant expert
Harmful	6.74	0.7084	0.2311	0.0605	refusal 87.5%, correction 12.5%
WJ benign	7.00	0.2855	0.5334	0.1764	correction 68.5%, refusal 30.5%
OR-Bench	1.00	0.0548	0.3413	0.2321	purification 53.0%, correction 41.0%
Alpaca	0.50	0.0017	0.0115	0.0829	purification 99.5%

6 Discussion

6.1 Why dynamic fusion matters

The main comparison separates several possible explanations. If safety were solved by adding refusal behavior, the single refusal adapter would be sufficient; instead, it produces extreme over-refusal. If simple adapter averaging were sufficient, static equal fusion would be competitive; instead, it remains unsafe and over-refuses OR-Bench. If a fixed redirect rule were sufficient, the prior-redirect variant would match the final system; instead, it has higher ASR and OR-Bench FPR. If external prompting or a binary guard were sufficient, Self-Reminder, a stronger safety system prompt, or Llama Guard filtering would approach the same tradeoff; instead, they either leave ASR high or increase OR-Bench false refusal. RCF improves the tradeoff because it conditions the adapter mixture on the prompt representation and uses factorized frame/intent evidence to select different safety behaviors for different prompt types.

6.2 Boundary of the current evidence

The results support the primary Qwen harmful-safety claim and safety-oriented transfer to Llama-2 and Mistral. They do not support a claim that every backbone preserves benign utility without calibration. Llama-2 and Mistral both need model-specific benign tuning, especially for adversarial-looking benign prompts. The extended Qwen verification supports the low-ASR finding under a larger harmful set and gives a broader view of benign calibration across WildJailbreak benign, OR-Bench, and Alpaca. The prompt-deduplicated audit and the separated calibration-test rerun reduce concern that the compact OR-Bench result is only a calibration-overlap artifact. The separated rerun is especially informative because the calibrated variant reduces WildJailbreak benign FPR from 14.5% to 2.0% and OR-Bench FPR from 14.5% to 9.0% without increasing harmful ASR. The utility claim should therefore be framed as a balanced improvement over safety-only, prompt-only, and guard-only alternatives rather than as solved over-refusal. The ablations show that ambiguous

shift and text-risk shift should be described as calibration refinements, while safe redirect, hard-benign calibration, and deterministic response repair are the main validated components.

6.3 Deployment considerations

RCF is intended for service-side generation control, not for replacing policy review or moderation. It requires loading multiple LoRA experts and applying dynamic layer-wise scaling. This is lighter than full model duplication but more complex than prompt-only defenses. The deterministic response-repair stage should also be handled carefully in production: it should be pre-registered, logged as a separate post-generation action, and audited for safe task preservation on both benign and harmful-looking prompts. Future deployments should combine RCF with broader guardrails and logging for high-risk domains.

7 Conclusion

This paper presented RCF, a representation-conditioned adapter fusion framework for balancing jailbreak safety and benign utility in LLM services. The method factorizes prompt representations into frame and intent codes, uses their interaction to generate layer-wise fusion weights over refusal, correction, and purification adapters, and calibrates inference with risk gating and deterministic response repair. On Qwen2.5-7B-Instruct, RCF reduces harmful WildJailbreak ASR from 81.91% to 6.74% while keeping benign false refusal low across WildJailbreak, OR-Bench, and Alpaca in the compact protocol. A prompt-deduplicated audit preserves the same conclusion after removing hardmix calibration prompts from evaluation, and a separated calibration-test rerun confirms that over-refusal can be reduced substantially without increasing harmful ASR. External prompt and guard baselines retain higher harmful ASR in the same protocol, and the extended Qwen verification keeps harmful ASR at 9.0% under a larger harmful set. Mechanism-isolating baselines and ablations show that dynamic fusion, safe redirect, hard-benign calibration, and deterministic response repair are central to the balanced point, while transfer experiments indicate low harmful ASR on Llama-2 and Mistral with remaining model-dependent benign calibration gaps. These findings support representation-conditioned behavior control as a practical direction for safer LLM services.

Limitations

The main protocol uses 200 prompts per split, and the extended Qwen verification still falls short of a full benchmark sweep. The extended run uses all 210 available WildJailbreak benign prompts rather than 500 benign WildJailbreak prompts, and the 11.6% OR-Bench FPR indicates remaining room for utility calibration on hard benign security-adjacent prompts. The hardmix calibration pool overlaps some compact benchmark prompts, so the compact OR-Bench result should be read together with the prompt-deduplicated analysis and the separated calibration-test rerun. The latter removes the direct overlap concern but still reports 9.0% OR-Bench FPR,

which means over-refusal is reduced rather than solved. All main safety and benign-refusal metrics rely on a fixed automatic judge model, GPT-5.4-mini; broader judge agreement would further strengthen reproducibility. The strongest balanced result is on Qwen2.5-7B-Instruct, while Llama-2 and Mistral show benign over-refusal gaps. Finally, deterministic response repair improves measured utility but is an auditable post-generation calibration mechanism; it should be expanded beyond the current pre-specified categories and reported with trigger counts before production deployment.

References

Arditi A, Obeso O, Syed A, et al (2024) Refusal in language models is mediated by a single direction. In: Advances in Neural Information Processing Systems, vol 37. Curran Associates, Inc., pp 136037–136083, https://doi.org/10.52202/079017-4322, arXiv:2406.11717

Chao P, Debenedetti E, Robey A, et al (2024) Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In: Advances in Neural Information Processing Systems, Datasets and Benchmarks Track, vol 37. Curran Associates, Inc., pp 55005–55029, https://doi.org/10.52202/079017-1745

Chao P, Robey A, Dobriban E, et al (2025) Jailbreaking black box large language models in twenty queries. In: 2025 IEEE Conference on Secure and Trustworthy Machine Learning. IEEE, pp 23–42, https://doi.org/10.1109/SATML64287.2025.00010, arXiv:2310.08419

Cui J, Chiang WL, Stoica I, et al (2025) Or-bench: An over-refusal benchmark for large language models. In: Proceedings of the 42nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 267. PMLR, pp 11515–11542

Hu EJ, Shen Y, Wallis P, et al (2022) Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations

Inan H, Upasani K, Chi J, et al (2023) Llama guard: Llm-based input-output safeguard for human-ai conversations. arXiv preprint arXiv:2312.06674

Jiang AQ, Sablayrolles A, Mensch A, et al (2023) Mistral 7b. arXiv preprint arXiv:2310.06825

Jiang L, Rao K, Han S, et al (2024) Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In: Advances in Neural Information Processing Systems, vol 37. Curran Associates, Inc., pp 47094–47165, https://doi.org/10.52202/079017-1493, arXiv:2406.18510

Li K, Patel O, Viegas F, et al (2023) Inference-time intervention: Eliciting truthful answers from a language model. In: Advances in Neural Information Processing

Systems, pp 41451–41530, https://doi.org/10.52202/075280-1797

Liu X, Xu N, Chen M, et al (2024) Autodan: Generating stealthy jailbreak prompts on aligned large language models. In: International Conference on Learning Representations

Mazeika M, Phan L, Yin X, et al (2024) Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In: Proceedings of the 41st International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 235. PMLR, pp 35181–35224, arXiv:2402.04249

Qi X, Zeng Y, Xie T, et al (2024) Fine-tuning aligned language models compromises safety, even when users do not intend to! In: International Conference on Learning Representations, arXiv:2310.03693

Robey A, Wong E, Hassani H, et al (2023) Smoothllm: Defending large language models against jailbreaking attacks. arXiv preprint arXiv:2310.03684

Röttger P, Kirk H, Vidgen B, et al (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). Association for Computational Linguistics, Mexico City, Mexico, pp 5377–5400, https://doi.org/10.18653/v1/2024.naacl-long.301

Shen X, Chen Z, Backes M, et al (2024) "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In: Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, pp 1671–1685, https://doi.org/10.1145/3658644.3670388, arXiv:2308.03825

Souly A, Lu Q, Bowen D, et al (2024) A strongreject for empty jailbreaks. In: Advances in Neural Information Processing Systems, Datasets and Benchmarks Track, vol 37. Curran Associates, Inc., pp 125416–125440, https://doi.org/10.52202/079017-3984, arXiv:2402.10260

Touvron H, Martin L, Stone K, et al (2023) Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288

Wei A, Haghtalab N, Steinhardt J (2023) Jailbroken: How does llm safety training fail? In: Advances in Neural Information Processing Systems, pp 80079–80110, https://doi.org/10.52202/075280-3508

Xie Y, Yi J, Shao J, et al (2023) Defending chatgpt against jailbreak attack via self-reminders. Nature Machine Intelligence 5(12):1486–1496. https://doi.org/10.1038/s42256-023-00765-8

Xu Z, Jiang F, Niu L, et al (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). Association for Computational Linguistics, Bangkok, Thailand, pp 5587–5605, https://doi.org/10.18653/v1/2024.acl-long.303

Yang A, Yang B, Zhang B, et al (2024) Qwen2.5 technical report. arXiv preprint arXiv:2412.15115

Zheng L, Chiang WL, Sheng Y, et al (2023) Judging llm-as-a-judge with mt-bench and chatbot arena. In: Advances in Neural Information Processing Systems, pp 46595–46623, https://doi.org/10.52202/075280-2020

Zou A, Phan L, Chen S, et al (2023a) Representation engineering: A top-down approach to ai transparency. arXiv preprint arXiv:2310.01405

Zou A, Wang Z, Nasr M, et al (2023b) Universal and transferable adversarial attacks on aligned language models. arXiv preprint arXiv:2307.15043