WALKSAFE: Risk-aware Graph Random Walk with Bi-GRPO for LLM Safety

Shilong Pan[1], Zhiliang Tian[1]*, Wanlong Yu[1], Zhen Huang[1], Qingyu Qiu[1], Zihan Chen[1], Zhonghao Sun[1], Minlie Huang[2], Dongsheng Li[1]*

[1]College of Computer Science and Technology, National University of Defense Technology
[2]Department of Computer Science, Tsinghua University
panshilong18@nudt.edu.cn

Abstract

Large language models (LLMs) may generate harmful outputs on malicious inputs. Existing safety methods, including prompt engineering and model editing, rely on handcrafted templates or target-driven parameter modifications, limiting their generalizability in unseen harmful scenarios. Post-training aims to ensure LLM safety in general domains via supervised fine-tuning (SFT) or reinforcement learning (RL) on diverse malicious inputs. SFT needs annotated refusal samples while RL learns to refuse risk by exploring diverse harmful inputs. However, these methods tend to harshly refuse over any possible risks, sacrificing potentially useful information and degrading model utility. We argue that realistic malicious inputs often mix both harmful and helpful semantics (i.e., entities and relations), and LLMs should identify and remove only harmful relations while preserving useful ones. Thus, the original malicious user inputs can shift into safe queries, to which LLMs can respond safely and helpfully. In this paper, we propose WALKSAFE, a graph-based risk-aware training framework that enables LLMs to identify potential risks of key semantics (entities and relations) in user inputs via graph structure. By filtering harmful relations, LLMs can respond to safe input queries and then generate their corresponding safe and helpful responses. First, we model all entities and relations in the inputs with a graph structure. Second, we adopt a risk-aware random walk on the graph to quantify potential risk under multiple entities and relations. Then, we reconstruct safe queries by filtering harmful relations to promote the LLM to answer safely and helpfully rather than with direct refusals. Finally, we propose Bi-GRPO to post-train LLMs. As vanilla GRPO conducts only the intra-group comparison, Bi-GRPO performs both intra-group and inter-group comparisons between different response groups. The extra inter-group rewards encourage the model to distinguish harmful and safe semantics, and thus prefer safe and helpful responses. Experiments on three LLMs show that our models obtain SOTA results.

Introduction

Large language models (LLMs) have been widely adopted across domains (Hurst et al. 2024; Guo et al. 2025), but their output safety remains a critical concern (Zhou et al. 2024; Xu et al. 2025). Recent studies have shown that LLMs are vulnerable to various jailbreak attacks that bypass LLMs' safety mechanisms to elicit harmful, biased, or sensitive content (Yi et al. 2024; Shen et al. 2024). To mitigate these concerns, researchers have proposed various safety mechanisms to detect the risk and mitigate it, aimed at improving the safety of generated responses.

Existing methods to improve LLM output safety generally fall into three categories: prompt engineering, model editing, and post-training alignment. Prompt engineering (Zheng et al. 2024a; Pan et al. 2024) wraps inputs with designed safety templates to steer LLMs toward helpful and harmless responses. However, such prompting methods require carefully crafted safety templates. These methods limit their generalizability in various scenarios and thus are vulnerable to adversarial prompt attacks that intentionally disable the safety templates (Liu et al. 2024; Li et al. 2024a). Model editing (Jiang et al. 2025; Pan et al. 2025) partially adjusts internal parameters or representations to mitigate harmful outputs under target attacks. However, they typically operate on limited components in LLMs for specialized attacks, in which each editing approach addresses one specific attack type, resulting in a limited generalizability to unseen attacks (Li et al. 2024b; Youssef et al. 2025; Liu et al. 2025) and a possible worse degradation in overall performance, including incoherent outputs and poor instruction following (Huang et al. 2024; Gu et al. 2024). Both prompt engineering and model editing approaches rely on designs tailored to specific harmful scenarios, making them hard to generalize to various unseen cases.

To address the above issues, researchers have adopted post-training to improve model safety in general domains, which trains models with datasets covering diverse malicious scenarios. Post-training typically includes supervised fine-tuning (SFT) and reinforcement learning (RL). SFT-based methods rely on annotated datasets with malicious inputs and safe (refusal) responses, but require costly manual data collection (Yuan et al. 2024; Yu et al. 2024). RL-based methods explore various harmful inputs and optimize models with rewards favoring refusals towards harmful inputs, which avoid using supervised datasets (Bai et al. 2022; Guan et al. 2024; Li et al. 2025a). However, these methods often encourage refusing the entire input over any possible risks, ignoring potential helpful responses.

In RL-based post-training, rather than direct refusals for

*Corresponding Author

any possible harmful inputs, we argue that inputs contain various key semantics (i.e., entities and relations) and not all information is harmful. Therefore, to achieve both helpfulness and safety, rebuilding a harmless input query (erasing harmful parts from original inputs) and feeding the safety parts to LLMs still makes sense. By removing harmful semantics and keeping safe ones, LLMs can reconstruct safe queries and generate safe and helpful responses.

In this paper, we propose WALKSAFE, a graph-based risk-aware post-training framework, which utilizes a risk-aware random walk on an entity-relation graph (ERG) to quantify the risk of multiple entities and relations. By selectively removing harmful relations and retaining safe ones, WALKSAFE proposes a Bi-GRPO (bi-group relative policy optimization) to post-train LLMs towards safer and more helpful responses. Specifically, to model all relations among entities, WALKSAFE extracts (entity, relation, entity) triples from inputs to construct an ERG, where entities serve as vertices and relations as edges. Then, to quantify risk among multiple entities and relations (subgraphs), WALKSAFE performs a risk-aware random walk (RRW), where edges with higher risk scores have lower walk probabilities. Probabilistic walks on the ERG help to quantify and identify high-risk edges. We selectively remove risk edges to keep a safe query as LLMs' input. With both original and safe queries, WALKSAFE introduces Bi-GRPO training. While vanilla GRPO compares responses within a single query, Bi-GRPO performs both intra-group and inter-group comparisons between responses from harmful and safe queries, guiding the model to prefer safe and helpful responses. We evaluate WALKSAFE on three LLM backbones across safety and helpfulness metrics, and the results show that WALKSAFE outperforms all baselines.

Our main contributions are threefold as follows: (1) We propose a graph-based risk-aware training framework that filters harmful and safe content by performing risk-aware random walks over the entity-relation graph, enabling the model to generate safe and helpful responses even with harmful inputs. (2) We propose Bi-GRPO that not only ranks responses within the same group (as vanilla GRPO does), but also jointly incorporates cross-group rewards between responses from original and safe queries, comparatively enhancing the safety and helpfulness of model outputs. (3) Extensive experiments demonstrate that WALKSAFE achieves SOTA performance on three LLM backbones.

Related Work

Safety Approaches for LLMs

Safety approaches for LLMs can be broadly categorized into (1) Prompt Engineering, (2) Model Editing, and (3) Post-training Alignment.

Prompt Engineering. The PTST method (Lyu et al. 2024) incorporated customized safety templates during inference and demonstrated strong defensive capabilities in standardized domains. The RePD framework (Wang, Liu, and Xiao 2024) employed a retrieval-based prompt decomposition strategy combined with a one-shot learning paradigm to defend against jailbreak attacks. Zheng et al. (2024b) proposed the DRO method to optimize safety prompts and enhance the security of LLMs by rejecting harmful queries. PARDEN (Zhang, Zhang, and Foerster 2024) enabled the detection of potential jailbreak behaviors by prompting the model to repeat its own outputs. AttentionDefense (Siska and Sankaran 2025) leveraged the attention weights of system prompts to detect and defend against jailbreak attacks.

Model Editing. The LED framework (Zhao et al. 2024) employed a hierarchical editing mechanism to achieve fine-grained defense and to effectively resist templatic attacks. Wang et al. (2025) developed the DELMAN dynamic editing framework, which defended against semantic paraphrasing and suffix attacks by directly editing critical parameter layers of LLMs. Zhang et al. (2025a) proposed JBShield, which extracted attack-related activation vectors and injected steering vectors to defend against jailbreak attacks such as code obfuscation. Wang et al. (2024b) introduced the SELFDEFEND framework, which employed a shadow model to detect and block harmful prompts. DINM (Wang et al. 2024a) precisely removed the toxic regions of LLMs through targeted knowledge editing.

Post-training Alignment. Reinforcement Learning from Human Feedback (RLHF) (Bai et al. 2022; Dai et al. 2023; Tan et al. 2025) remains a foundational technique for aligning LLMs with safe and helpful responses. Su, Kempe, and Ullrich (2024) introduced Enhanced RLHF (E-RLHF), a method that aligned harmful prompts to safety-paraphrased references. GRPO (Li et al. 2025b; Yang et al. 2025) combined multi-label reward regression with policy optimization, providing a lightweight yet effective alternative to PPO-based RLHF, DPO (Zhao et al. 2025; Zhang et al. 2025b), and achieving multi-dimensional safety alignment for LLMs.Xu et al. (2024) proposed SafeDecoding, which incorporated safety signals into the generation process and effectively reduced the success rates of various jailbreak attacks. STAIR (Zhang et al. 2025c) combined safety-aware reasoning and tree search to enable self-training, thus enhancing model safety. DeRTa (Yuan et al. 2025) employed a two-pronged training strategy, incorporating harmful response prefixes and applying reinforced transition optimization at each token position.

Red Teaming for LLMs

Red teaming methods aim to proactively discover safety vulnerabilities in LLMs by querying LLMs with adversarial inputs, which can be broadly categorized into (1) Manual Red Teaming and (2) Automated Red Teaming.

Manual Red Teaming. Ribeiro et al. (2020) and Röttger et al. (2020) systematically tested model weaknesses in dialogue and hate speech scenarios by constructing test templates. Xu et al. (2021) introduced adversarial dialogue agents that simulated high-risk conversations to improve the safety of the model through realistic stress tests.

Automated Red Teaming. APRT (Jiang et al. 2024) systematized and parameterized red teaming tasks, making them learnable. Xiong, Chen, and Ho (2025) proposed the CoP framework, which constructed complex adversarial prompts to enable comprehensive safety evaluations. HARM (Zhang et al. 2024) generated large-scale test cases

based on fine-grained risk categorization to identify vulnerabilities and support model safety alignment. Lees et al. (2022) introduced the perspective API, a fast and multilingual toxicity classifier covering six fine-grained risk types, which are widely adopted as an up-to-date automated red-teaming tool.

Methodology

Overview

Our proposed method (Fig. 1) consists of three parts: (1) Entity-relation graph for structural modeling, which extracts entities and relations from input texts to construct a structured entity-relation graph (ERG); (2) Risk-aware random walk for subgraph risk quantification, which performs risk-aware random walks (RRW) on the ERG and its subgraphs (i.e., multi-hop entities and relations) for risk quantification, to extract safe subgraphs by removing risky relations; (3) Bi-GRPO training with cross-group rewards, which ranks responses from both original and safe input queries (i.e., reconstructed from the safe subgraphs) to optimize for better safe and helpful responses.

Our method adopts a multi-stage training framework: firstly extracting all entities and relations to build an ERG (§3.2 entity-relation graph for structural modeling), then quantifying risk of subgraphs in ERG to rebuild safe input queries by removing harmful relations (§3.3 risk-aware random walk for subgraph risk quantification), and finally optimizing LLMs with reward ranking between responses from the original and the reconstructed safe input queries (§3.4 Bi-GRPO training with cross-group rewards).

Entity-Relation Graph for Structural Modeling

To enable structured modeling of entities and relations, we construct an entity-relation graph (ERG) from the user inputs, in which an edge represents a relation, connecting two entities (vertices). The motivation is that existing risk assessments in LLMs lack identification of hidden harmful or safe relations among entities in inputs, resulting in unsafe outputs when regarding harmful relations as safe ones, or over-refusals when misclassifying safe relations. Therefore, we use the structured ERG to explicitly model all relations among entities for better quantification of hidden risk.

Specifically, we use the deepseek-r1 (a low-overhead alternative for human extraction) to extract all (entity, relation, entity) triples from each input. We regard an entity as a vertex v and a relation as an edge e. Therefore, we can use all triples $t_k = (v_i, e_{ij}, v_j)$ to construct a global graph as $G = (V, E, \mathcal{T})$, where $V = \{v_i | v_i \in t_k, t_k \in \mathcal{T}\}$ denotes the set of entity vertices, $E = \{e_i | e_i \in t_k, t_k \in \mathcal{T}\}$ denotes the set of relation edges, and $\mathcal{T} = \{t_k = (v_i, e_{ij}, v_j)\}$ is the set of all triples. Each user input corresponds to a local subgraph in the global graph $\mathcal{G} \subseteq G$, which models its own entities and relations.

Finally, we obtain the ERG that models all relations among entities of all inputs, where each input corresponds to a subgraph used for risk quantification in the following section.

Risk-aware Random Walk for Subgraph Risk Quantification

To further quantify the risk of a subgraph (entities with their relations), we propose a risk-aware random walk, which uses risk-based probabilistic walks among multiple entities and their relations to capture critical high-risk relations in the subgraph. The motivation is to quantify the risk of multiple connected entities and relations (i.e., subgraph) rather than a single entity-relation-entity triple, because risk can arise not from individual triple, but from combined ones (e.g., (someone, consults, hydrogen) and (hydrogen, can be, explosive) are individually benign, but together they imply someone's risky intent for making explosive devices). Therefore, we can selectively remove harmful content while retaining safe content to get safer queries, thus leading to safer and more helpful model responses.

Risk-aware Random Walk. To quantify risk among multiple entities, we propose a risk-aware random walk (RRW) algorithm on the ERG, where we integrate risk scores into walk probabilities.

Specifically, the RRW consists of 3 steps as follows:

(1) Edge risk scoring. We compute an isolated risk score $r_{e_{ij}}$ for each edge in the ERG based on its corresponding triple $t_k = (v_i, e_{ij}, v_j)$. Let $\mathrm{emb}(t_k)$ denote the feature embedding of the triple and w, b be the weights and biases of the perspective model (an up-to-date risk detection model and a low-overhead alternative for human annotation) (Lees et al. 2022). We scale the result into $[0, 1]$ using the logistic function σ: $r_{e_{ij}} = \sigma\big(w^\top \mathrm{emb}(t_k) + b\big)$. A higher score indicates a higher risk.

(2) Walk probability initialization. We initialize a walk probability $p_{e_{ij}}$ for each edge $e_{ij} \in E$ based on its risk score $r_{e_{ij}}$ defined as $p_{e_{ij}} = 1 - r_{e_{ij}}$. Thus, edges with higher risk have lower walking probability.

(3) Risk quantification via random walk. We perform multiple rounds of random walks on the entire ERG according to p_e, computing the walk frequency $f_{\mathrm{global}}(e)$ for each edge. We then get the global ERG expected risk as

$$r_{\mathrm{global}} = \mathbb{E}_{e \in E}\big[f_{\mathrm{global}}(e) \cdot r_e\big]. \tag{1}$$

Similarly to above, we apply RRW to the subgraph of each input to obtain walk frequency $f_{\mathrm{local}}(e)$ and local subgraph expected risk r_{local}. The global walk estimates the risk of the entire ERG built on all inputs in the dataset, while the local walk evaluates the risk of a subgraph for a single input.

Iterative Safe Subgraph Extraction. To iteratively isolate a safe subgraph from the original subgraph for each input x_{orig}, we propose an extraction strategy considering both the local risk of the subgraph and the global risk across the ERG obtained in the above subsection. Specifically, we iteratively extract a safe subgraph as the following four steps: (1) For each edge e, we compute a walk score w_e by multiplying the walk frequency $f_{\mathrm{local}}(e)$ and the risk score r_e as $w_e = f_{\mathrm{local}}(e) \cdot r_e$, which quantify the risk of an edge considering its walking frequency. (2) Then, we remove the edge with the largest walk score, which represents the most

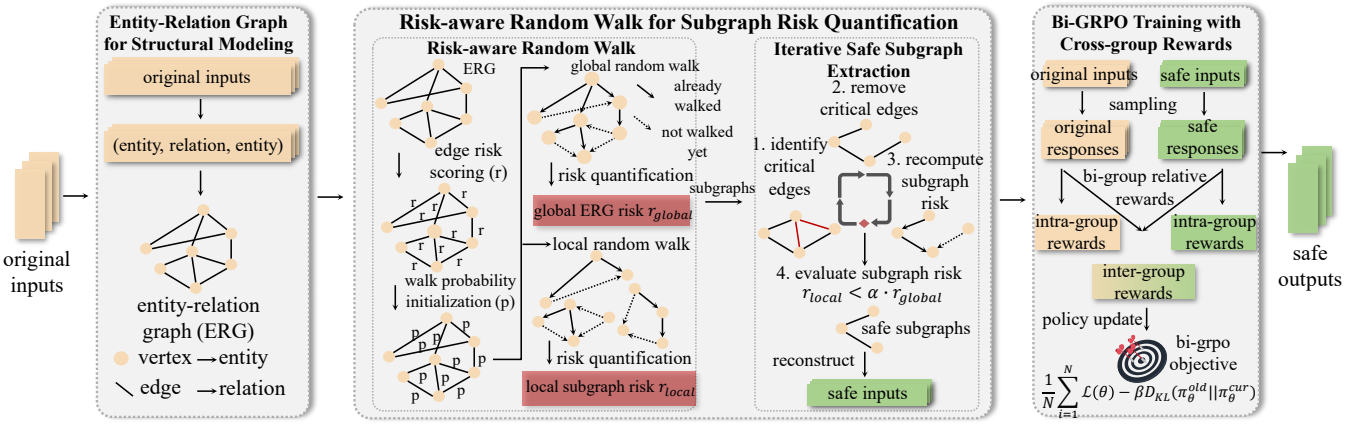

Figure 1: Overview of WALKSAFE. Given original inputs, WALKSAFE follows a three-stage process (gray background) from left to right: (1) ERG construction (left): extracts entities and relations from the original queries to build a global ERG, where each query corresponds to a subgraph. (2) RRW for risk quantification (middle): scores edge risk and assigns walk probabilities and performs global/local walks to iteratively remove risky edges, extracting a safe subgraph and reconstructing a safe query. (3) Bi-GRPO training (right): samples responses from original and safe input queries, computes intra/inter-group rewards, and updates the policy with bi-group relative optimization. Finally, LLMs trained with WALKSAFE produce safe and helpful outputs.

critical high-risk edge during RRW. (3) We recompute the r_{local} as Eq. 1 over the updated subgraph. (4) We terminate the iteration when it meets $r_{\text{local}} < \alpha \cdot r_{\text{global}}$, otherwise the iteration continues from step (1). α is a scaling factor.

Finally, the resulting subgraph contains safe entities and relations while removing harmful ones. Therefore, we can rebuild a safer input query x_{safe} based on the safe subgraph, which serves for the following training.

Bi-GRPO Training with Cross-group Rewards

To further enhance our model's helpfulness and harmlessness during training, we propose a bi-group relative policy optimization (Bi-GRPO), which augments vanilla GRPO by adding cross-group reward ranking between responses from the original input and the safe input queries obtained in the above section. The motivation is that vanilla GRPO generates multiple responses as one group for the same input, and rewards these responses based on comparisons within only one group via reinforcement learning (RL), which fails to receive possibly higher rewards from another response group (e.g, responses from the safe query). Therefore, we extend a cross-group reward comparison in RL between two response groups from the original input and the safe input query, capturing potentially higher rewards across groups to optimize the model for safer and more helpful responses.

Initialization by Supervised Fine-tuning. To initialize the model with safe-generation capability, following the steps in the above sections, we perform supervised fine-tuning (SFT) on our multi-stage safety reasoning dataset constructed following methods in §3.2 entity-relation graph for structural modeling and §3.3 risk-aware random walk for subgraph risk quantification. Specifically, we construct each sample in our dataset, consisting of four reasoning steps as follows:

(1) ERG construction. Following methods in §3.2 entity-relation graph for structural modeling, given an original dataset $\mathcal{D}_{\text{orig}} = \{(x^i_{\text{orig}}, y^i_{\text{orig}})\}^N_{i=1}$, we extract entities and relations from each input x^i_{orig}. We use all entities and relations to construct a global ERG G and also construct a local subgraph g^i_{orig} for each input with entities and relations from itself. Let $\mathcal{G}_{\text{orig}}$ donate the original set of subgraphs g^i_{orig} as $\mathcal{G}_{\text{orig}} = \{g^i_{\text{orig}}\}^N_{i=1}$.

(2) Safe subgraph extraction. Following methods in §3.3 risk-aware random walk for subgraph risk quantification, given the ERG G and the original subgraph set $\mathcal{G}_{\text{orig}}$, we apply RRW to quantify subgraph risk and extract safe subgraphs $\mathcal{G}_{\text{safe}} = \{g^i_{\text{safe}}\}^N_{i=1}$.

(3) Safe input query reconstruction. Given the safe subgraph $g^i_{\text{safe}} \in \mathcal{G}_{\text{safe}}$, we reconstruct a new input query x^i_{safe}, removing detected high-risk entities and relations while preserving safe ones.

(4) Distill ground-truth answer. Given the safe input query, we distill the answer y^i_{safe} from deepseek-r1, which is the ground truth for model training.

Above all, we combine the four steps above to get a safety reasoning answer $\mathcal{Y}^i_{\text{safe}} = (g^i_{\text{orig}}, g^i_{\text{safe}}, x^i_{\text{safe}}, y^i_{\text{safe}})$, and thus get our multi-stage safety reasoning dataset $\mathcal{D}_{\text{safe}} = \{(x^i_{\text{orig}}, \mathcal{Y}^i_{\text{orig}})\}^N_{i=1}$.

Then, we train our model on the $\mathcal{D}_{\text{safe}}$ by SFT objective: $\mathcal{L}_{\text{sft}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(\mathcal{Y}^i_{\text{safe}} \mid x^i_{\text{orig}})$, where θ denotes the model parameters, and $p_\theta(\cdot)$ represents the model's conditional probability distribution over output tokens.

Bi-Group Relative Policy Optimization. To jointly optimize helpfulness and harmlessness for responses from re-

constructed safe input queries, we extend vanilla GRPO into Bi-Group Relative Policy Optimization (Bi-GRPO), combining intra-group rewards and inter-group rewards. Intra-group rewards compare responses from the same input, while inter-group rewards compare responses between the original x_{orig} and the safe input query x_{safe}.

Specifically, the Bi-GRPO approach consists of 3 steps:

Step 1: Sampling. At each iteration, we first sample M responses $\{y^j_{\text{orig}}\}_{j=1}^M$ from the original input query x_{orig}. Then, we sample another M responses $\{y^j_{\text{safe}}\}_{j=1}^M$ from the safe input query x_{safe}.

Step 2: Bi-group relative reward computation. For each sampled response $y^j \in \{y^j_{\text{orig}}, y^j_{\text{safe}}\}$, we compute 3 scalar reward components: (1) harmlessness reward R^j_{ha}: a score from an existing safety-customized model[1], which was trained to evaluate the safety of a text, measuring harmful, biased, sensitive content. The higher reward means the higher safety. (2) helpfulness reward R^j_{he}: a score from an existing model[2] customized in detection of helpful response. The higher reward means the higher helpfulness. (3) format reward R^j_{fmt}: a score adherence to the desired output format (e.g., structural reasoning steps). The higher score means the better match with the desired output format. We combine these into one overall reward: $R = R^j_{\text{ha}} + R^j_{\text{he}} + R^j_{\text{fmt}}$, Since we get two groups of response candidates $\{y^j_{\text{orig}}\}_{j=1}^M$ and $\{y^j_{\text{safe}}\}_{j=1}^M$ in the previous sampling step, we calculate the overall reward R for each of them to get two group rewards $\{R^j_{\text{orig}}\}_{j=1}^M$ and $\{R^j_{\text{safe}}\}_{j=1}^M$, respectively.

We define intra-group rewards the same as the previous overall rewards $\{R^j_{\text{intra}}\}_{j=1}^M \in \{\{R^j_{\text{orig}}\}_{j=1}^M, \{R^j_{\text{safe}}\}_{j=1}^M\}$ We define inter-group rewards as the difference between rewards from the original and safe group: $\{R^j_{\text{inter}} | R^j_{\text{inter}} = R^j_{\text{safe}} - R^j_{\text{orig}}\}_{j=1}^M$.

Following vanilla GRPO, we calculate relative rewards for each response in the intra-group A^j_{intra} and inter-group A^j_{inter}, which is the difference between the absolute reward $R^j_{\{\text{intra,inter}\}}$ and the mean reward $\mu_{\{\text{intra,inter}\}}$ divided by standard deviation $\sigma_{\{\text{intra,inter}\}}$:

$$A^j_{\text{intra}} = \frac{R^j_{\text{intra}} - \mu_{\text{intra}}}{\sigma_{\text{intra}}}, A^j_{\text{inter}} = \frac{R^j_{\text{inter}} - \mu_{\text{inter}}}{\sigma_{\text{inter}}}. \quad (2)$$

Rather than one intra-group reward A^j_{intra} for a response within one group in vanilla GRPO, we extend an extra inter-group reward A^j_{inter} that encourages the model to prefer and reward higher toward safer and more helpful responses by comparison across different groups.

Finally, we obtain a bi-group relative rewards $\{A^j\}_{j=1}^M$ merged with intra-group rewards and inter-group rewards, in which each relative reward A^j is a combination of A^j_{intra} and A^j_{inter} as $A^j = (A^j_{\text{intra}} + A^j_{\text{inter}})/2$.

[1]https://huggingface.co/google/shieldgemma-2b

[2]https://huggingface.co/Ray2333/gpt2-large-helpful-reward_model

Vanilla GRPO performs intra-group comparisons among responses within the same group from a single query. In contrast, our Bi-GRPO performs both intra-group and inter-group comparisons between two response groups from harmful and safe queries, respectively. The inter-group comparisons help the model to distinguish safe and harmful semantics under harmful and safe queries, thus assigning higher rewards to safe and helpful responses.

Step 3: Policy update. Based on bi-group relative rewards $\{A^j\}_{j=1}^M$ for the M responses $\{y^j\}_{j=1}^M$ of input x^i, we calculate the loss for x^i under the current policy π_θ: $\mathcal{L}^i(\theta) = -\frac{1}{M}\sum_{j=1}^M \log \pi^\theta(y^j_{\text{safe}}|x^i) \cdot A^j$. $\mathcal{L}^i$ is the per-input loss only for one input x^i and its M responses $\{y^j\}_{i=1}^M$.

To build the overall training objective $\mathcal{J}_{\text{Bi-GRPO}}$, we aggregate the loss across all training inputs $\{x^i\}_{i=1}^N$ by averaging their individual losses $\mathcal{L}^i$ and adding a punishment of over-update with KL divergence $\mathbb{D}_{\text{KL}}(\pi_\theta^{\tau-1}||\pi_\theta^\tau)$ between the old policy π_θ^{old} at the last iteration and the current policy π_θ^{cur}:

$$\mathcal{J}_{\text{Bi-GRPO}} = \frac{1}{N}\sum_{i=1}^N \mathcal{L}^i(\theta) - \beta\mathbb{D}_{\text{KL}}(\pi_\theta^{\text{old}}||\pi_\theta^{\text{cur}}). \quad (3)$$

We can update the policy model π_θ by maximizing the objective during iterative training.

Experiments

Experimental Settings

Models. We use three open-source LLMs, including Qwen2.5-3B-Instruct, Llama-3.2-3B-Instruct, and Qwen3-4B.

Datasets. We use STAIR-SFT (Zhang et al. 2025c) and TET Luong et al. (2024) datasets to evaluate our methods.

Baselines. Our baselines comprise: (1) three commercial closed-source models, including `GPT-4o` (Hurst et al. 2024), `o1` (Jaech et al. 2024), and `o4-mini` (OpenAI 2025); (2) two existing safety methods, including `STAIR` (Zhang et al. 2025c) and `DeRTa` (Yuan et al. 2025); (3) three original versions of our training models, including `Qwen2.5-3B-Instruct`, `Llama-3.2-3B-Instruct`, and `Qwen3-4B`.

Metrics. Following Luong et al. (2024), we evaluate model responses on the TET benchmark with two major dimensions: (1) harmlessness, using perspective API (Lees et al. 2022) scores across six distinct risk types: toxicity, severe toxicity, identity attack, insult, profanity, and threat, where a lower harmlessness score indicates better safety; and (2) helpfulness, using the helpfulness detection model as in Yang et al. (2024), where a higher helpfulness score indicates better helpfulness.

Overall Performance

Tab. 1 presents the harmlessness scores and helpfulness scores across nine baselines and three models trained with our WALKSAFE. The overall results show that our models consistently achieve better performance both on harmlessness and helpfulness, demonstrating the effectiveness of

Model	Harmlessness ↓							Helpfulness ↑
	Toxicity	S-Toxicity	Id Attack	Insult	Profanity	Threat	Avg.	
GPT-4o	17.85	6.33	4.52	10.03	18.33	4.62	10.28	0.00
o1	20.43	6.26	4.02	17.38	17.40	5.38	11.81	-0.45
o4-mini	21.80	7.18	5.58	20.79	18.55	5.89	13.30	-0.50
Qwen3-4B								
Vanilla	39.05	17.66	17.01	26.85	36.39	11.41	24.73	0.17
Ours	24.96	11.74	13.97	9.35	26.39	7.34	15.62	1.00
Qwen2.5-3B-Instruct								
Vanilla	27.82	12.07	10.04	14.34	25.93	6.84	16.17	0.04
STAIR	14.13	1.16	8.92	5.10	10.68	2.53	7.09	0.10
Ours	12.95	1.94	7.25	5.26	10.93	2.49	6.80	**2.58**
Llama3.2-3B-Instruct								
Vanilla	20.32	3.65	6.72	15.91	14.43	6.30	11.22	-0.12
STAIR	12.43	0.65	8.46	4.71	7.65	2.51	6.07	-0.49
DeRTa	12.63	4.63	4.61	5.66	12.14	3.46	7.19	-0.09
Ours	**6.63**	**0.52**	**2.25**	**2.92**	**5.58**	**1.61**	**3.25**	1.41

Table 1: Comparison of six types of harmless score and helpful score across 9 baseline models and 3 of our models. A lower harmlessness score indicates better safety, and a higher helpfulness score indicates better helpfulness. Bolded numbers indicate best performance. Note that baseline comparisons vary across LLM backbones due to the limited open-sourced models released by STAIR and DeRTa. Our improvements are significant under the ks-test with $p < 0.01$.

our methods in improving LLM safety and helpfulness. Notably, the average harmlessness improves by more than 71% on Llama3.2 (vanilla vs. Ours), achieving the best safety across all baselines, including o1, o4-mini, and other safety methods. Additionally, rather than other baselines struggling with low helpfulness under malicious inputs, our models achieve far higher helpfulness than all baselines, generating more helpful and safe answers instead of simple helpless refusals. Moreover, WALKSAFE shows consistent improvement across all model sizes and types, which confirms the generalizability of our approach.

Ablation Study

As shown in Tab. 2, we conduct ablation studies to validate the effectiveness of each major part in our proposed WALKSAFE. W/o ERG (§3.2 entity-relation graph for structural modeling), indicating the removal of the entity-relation graph while instructing the model itself to evaluate risk, causes a substantial increase in average harmlessness and a decrease in helpfulness. It shows that the structural ERG is critical to model entities and relations for risk quantification. W/o RRW (§3.3 risk-aware random walk for subgraph risk quantification), replacing our risk-aware random walk with an empirical threshold to filter risky edges, degrades the safety and helpfulness (-103.5% and -48.2% respectively at most on Llama3.2). It confirms the effectiveness of RRW to quantify the risk of subgraphs considering multiple entities and relations. W/o Bi-GRPO (§3.4 Bi-GRPO training with cross-group rewards) applies the vanilla GRPO instead of our Bi-GRPO during training. It increases the average risk by 10.0% and decreases the average helpfulness by 68.2% across 3 models, which validates the effectiveness of Bi-GRPO in improving LLM safety and helpfulness.

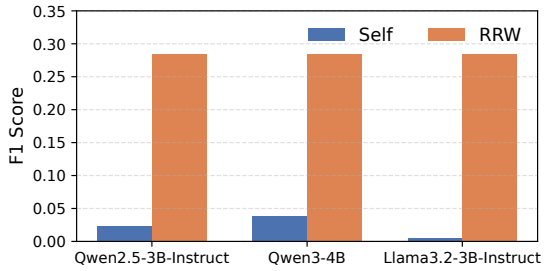

Figure 2: F1 score of extracted safe entities and relations by RRW (orange bar) and the models themselves (blue bar). The y-axis indicates the F1 score, and the x-axis indicates the model backbones. Higher F1 scores indicate greater ability to correctly identify harmful (precision) and avoid missing them (recall).

Analysis Study of RRW

To verify the motivation of our RRW algorithm that can better quantify and filter the hidden risk among multiple entities and relations, we conduct an analysis comparing risk-filtered entities and relations from (1) RRW, (2) the model itself, and (3) a human-level model (deepseek-r1) as a reference.

Specifically, based on all extracted entities and relations in ERG (§3.2 entity-relation graph for structural modeling), we apply RRW, the model itself, and deepseek-r1 to filter high-risk relations and retain safe ones. We evaluate the results from RRW and the model by computing precision and recall against the deepseek-r1 results as references. Higher precision and recall indicate the method's greater ability to correctly identify harmful (precision) and avoid missing them (recall). Based on the precision and recall values, we calculate F1 scores for RRW and the model itself.

Model Backbone	Ablation	Harmlessness ↓							Helpfulness ↑
		Toxicity	S-Toxicity	Id Attack	Insult	Profanity	Threat	Avg.	
Qwen2.5-3B-Instruct	full model	**12.95**	**1.94**	**7.25**	**5.26**	**10.93**	**2.49**	**6.80**	**2.58**
	w/o ERG	24.09	8.64	12.10	11.08	22.93	5.33	14.03	1.40
	w/o RRW	19.41	7.27	7.87	10.46	18.28	4.35	11.27	2.10
	w/o Bi-GRPO	14.15	2.56	7.60	5.44	12.16	2.80	7.45	1.02
Qwen3-4B	full model	**24.96**	**11.74**	**13.97**	**9.35**	**26.84**	**7.34**	**15.63**	**1.00**
	w/o ERG	28.94	19.04	19.95	13.31	32.58	9.16	20.45	0.65
	w/o RRW	27.21	15.34	16.48	12.56	28.65	9.22	18.24	0.85
	w/o Bi-GRPO	26.57	12.57	16.53	12.00	31.96	8.74	18.06	0.31
Llama3.2-3B-Instruct	full model	**6.63**	**0.52**	**2.25**	**2.92**	**5.58**	**1.61**	**3.25**	**1.41**
	w/o ERG	11.42	2.11	7.52	5.21	11.05	2.91	6.70	0.88
	w/o RRW	11.79	3.50	4.96	5.39	10.85	3.20	6.62	0.73
	w/o Bi-GRPO	6.97	0.67	2.48	3.00	5.68	1.67	3.41	0.35

Table 2: Ablation study on different components of our method across three model backbones. A lower harmlessness score indicates better safety, and a higher helpfulness score indicates better helpfulness. Bolded numbers indicate best performance in the same model group.

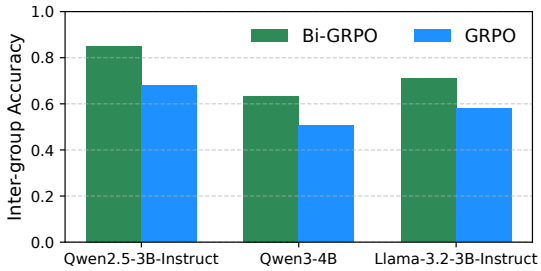

Figure 3: Inter-group ranking accuracy (y-axis) of Bi-GRPO (green bar) and GRPO (blue bar) across model backbones (x-axis). Higher accuracy indicates a stronger preference for harmlessness and helpfulness.

As shown in Fig. 2, blue bars (Self) represent the results from models themselves, and orange bars (RRW) represent RRW results. The orange bars' results across three models are the same because the RRW is model-independent. The results show that RRW consistently outperforms the models on the F1 score across three models, indicating that RRW can better quantify the risk of entities and relations.

Analysis Study of Bi-GRPO

To verify the motivation of Bi-GRPO, we compare Bi-GRPO with vanilla GRPO through an inter-group analysis, in which we score responses from the original inputs and safe input queries (§3.4 Bi-GRPO training with cross-group rewards) to validate whether Bi-GRPO actually provides a preference towards more harmless and helpful responses than GRPO during training.

Specifically, we used two models optimized with GRPO and Bi-GRPO respectively to generate responses for both original input x_{orig} and safe input query x_{safe}, yielding four groups of responses: $y_{\text{orig}}^{\text{grpo}}$, $y_{\text{safe}}^{\text{grpo}}$, $y_{\text{orig}}^{\text{bi-grpo}}$, and $y_{\text{safe}}^{\text{bi-grpo}}$. Following the main experiments, we use the TET dataset to generate responses. We evaluate each response by harmlessness-helpfulness score as $s_{\text{hh}} = s_{\text{ha}} + s_{\text{he}}$, where s_{ha} is the negative average score across six dimensions from the perspective API (same as harmful evaluation in our main experiments), and s_{he} is the helpfulness score from an open-source helpful reward model (same as helpful reward model in Bi-GRPO training). We define an inter-group ranking accuracy $\text{Acc}_{\text{inter}}$ as the proportion of cases where the safe response outperforms the original response: $\text{Acc}_{\text{inter}} = \frac{1}{N}\sum_{i=1}^{N} 1[s_{\text{hh}}(y_{\text{safe},i}) > s_{\text{hh}}(y_{\text{orig},i})]$. It measures whether responses to safe input queries are actually safer and more helpful than the original ones during training with Bi-GRPO or GRPO.

As shown in Fig. 3, Bi-GRPO (green) consistently outperforms GRPO (blue) on $\text{Acc}_{\text{inter}}$ across all model backbones, demonstrating its effectiveness in generating safer and more helpful responses.

Conclusion

In this paper, we propose WALKSAFE, a graph-based risk-aware training framework that enables LLMs to quantify potential risk among multiple entities and relations and to generate safe and helpful responses from risk-filtered input queries. WALKSAFE first (1) constructs an entity-relation graph from inputs, then (2) performs a risk-aware random walk on the graph to quantify risks among multiple entities and relations and reconstructs safe input queries by removing high-risk edges, and finally (3) adopts the Bi-GRPO that leverages cross-group rewards between responses from original and safe input queries to enhance harmlessness and helpfulness during training. Extensive experiments on three LLM backbones demonstrate that WALKSAFE consistently outperforms existing baselines.

Acknowledgements

This work is supported by the following foundations: the National Natural Science Foundation of China un-

der Grant No.62025208 and 62421002, the National Science Foundation for Distinguished Young Scholars under Grant No.62325604, No.62125604, the Young Elite Scientist Sponsorship Program by CAST under Grant No.YESS20230367, and the National Natural Science Foundation of China (NSFC) under Grant No.62306330, No.62376284.

References

Bai, Y.; Kadavath, S.; Askell, A.; et al. 2022. Training a helpful and harmless assistant with rlhf. *arXiv preprint arXiv:2204.05862*.

Dai, J.; Pan, X.; Sun, R.; Ji, J.; Xu, X.; Liu, M.; Wang, Y.; and Yang, Y. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*.

Gu, J.-C.; Xu, H.-X.; Ma, J.-Y.; Lu, P.; Ling, Z.-H.; Chang, K.-W.; and Peng, N. 2024. Model editing harms general abilities of large language models: Regularization to the rescue. *arXiv preprint arXiv:2401.04700*.

Guan, M. Y.; Joglekar, M.; Wallace, E.; Jain, S.; Barak, B.; Helyar, A.; Dias, R.; Vallone, A.; Ren, H.; Wei, J.; et al. 2024. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*.

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

Huang, X.; Liu, J.; Wang, Y.; and Liu, K. 2024. Reasons and solutions for the decline in model performance after editing. *Advances in Neural Information Processing Systems*, 37: 68833–68853.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Jiang, B.; Jing, Y.; Shen, T.; Wu, T.; Yang, Q.; and Xiong, D. 2024. Automated progressive red teaming. *arXiv preprint arXiv:2407.03876*.

Jiang, H.; Zhao, Z.; Fang, J.; Ma, H.; Wang, R.; Deng, Y.; Wang, X.; and He, X. 2025. Mitigating Safety Fallback in Editing-based Backdoor Injection on LLMs. *arXiv preprint arXiv:2506.13285*.

Lees, A.; Tran, V. Q.; Tay, Y.; Sorensen, J.; Gupta, J.; Metzler, D.; and Vasserman, L. 2022. A new generation of perspective api: Efficient multilingual character-level transformers. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 3197–3207.

Li, W.; Yang, W.; Hou, Y.; Liu, L.; Liu, Y.; and Li, X. 2025a. SARATR-X: Toward Building a Foundation Model for SAR Target Recognition. *IEEE Transactions on Image Processing*, 34(1): 869–884.

Li, X.; Li, Z.; Kosuga, Y.; and Bian, V. 2025b. Optimizing Safe and Aligned Language Generation: A Multi-Objective GRPO Approach. *arXiv preprint arXiv:2503.21819*.

Li, X.; Zhou, Z.; Zhu, J.; Yao, J.; Liu, T.; and Han, B. 2024a. DeepInception: Hypnotize Large Language Model to Be Jailbreaker. arXiv:2311.03191.

Li, Y.; Li, T.; Chen, K.; Zhang, J.; Liu, S.; Wang, W.; Zhang, T.; and Liu, Y. 2024b. Badedit: Backdooring large language models by model editing. *arXiv preprint arXiv:2403.13355*.

Liu, L.; Sun, S.; Zhi, S.; Shi, F.; Liu, Z.; Heikkilä, J.; and Liu, Y. 2025. A Causal Adjustment Module for Debiasing Scene Graph Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5): 4024–4043.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Luong, T. S.; Le, T.-T.; Van, L. N.; and Nguyen, T. H. 2024. Realistic evaluation of toxicity in large language models. *arXiv preprint arXiv:2405.10659*.

Lyu, K.; Zhao, H.; Gu, X.; Yu, D.; Goyal, A.; and Arora, S. 2024. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *Advances in Neural Information Processing Systems*, 37: 118603–118631.

OpenAI. 2025. OpenAI o3 and o4-mini System Card.

Pan, S.; Tian, Z.; Ding, L.; Zheng, H.; Huang, Z.; Wen, Z.; and Li, D. 2024. POMP: Probability-driven Meta-graph Prompter for LLMs in Low-resource Unsupervised Neural Machine Translation. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *ACL 2024*, 9976–9992. Bangkok, Thailand: Association for Computational Linguistics.

Pan, S.; Tian, Z.; Huang, Z.; Yu, W.; Wen, Z.; Liu, X.; Lu, K.; Huang, M.; and Li, D. 2025. AGD: Adversarial Game Defense Against Jailbreak Attacks in Large Language Models. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *ACL 2025*, 17391–17406. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Ribeiro, M. T.; Wu, T.; Guestrin, C.; and Singh, S. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. *arXiv preprint arXiv:2005.04118*.

Röttger, P.; Vidgen, B.; Nguyen, D.; Waseem, Z.; Margetts, H.; and Pierrehumbert, J. B. 2020. HateCheck: Functional tests for hate speech detection models. *arXiv preprint arXiv:2012.15606*.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2024. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 1671–1685.

Siska, C.; and Sankaran, A. 2025. AttentionDefense: Leveraging System Prompt Attention for Explainable Defense Against Novel Jailbreaks. *arXiv preprint arXiv:2504.12321*.

Su, J.; Kempe, J.; and Ullrich, K. 2024. Mission impossible: A statistical perspective on jailbreaking llms. *Advances in Neural Information Processing Systems*, 37: 38267–38306.

Tan, Y.; Jiang, Y.; Li, Y.; Liu, J.; Bu, X.; Su, W.; Yue, X.; Zhu, X.; and Zheng, B. 2025. Equilibrate rlhf: Towards balancing helpfulness-safety trade-off in large language models. *arXiv preprint arXiv:2502.11555*.

Wang, M.; Zhang, N.; Xu, Z.; Xi, Z.; Deng, S.; Yao, Y.; Zhang, Q.; Yang, L.; Wang, J.; and Chen, H. 2024a. Detoxifying large language models via knowledge editing. *arXiv preprint arXiv:2403.14472*.

Wang, P.; Liu, X.; and Xiao, C. 2024. Repd: Defending jailbreak attack through a retrieval-based prompt decomposition process. *arXiv preprint arXiv:2410.08660*.

Wang, X.; Wu, D.; Ji, Z.; Li, Z.; Ma, P.; Wang, S.; Li, Y.; Liu, Y.; Liu, N.; and Rahmel, J. 2024b. Selfdefend: Llms can defend themselves against jailbreaking in a practical manner. *arXiv preprint arXiv:2406.05498*.

Wang, Y.; Weng, F.; Yang, S.; Qin, Z.; Huang, M.; and Wang, W. 2025. DELMAN: Dynamic Defense Against Large Language Model Jailbreaking with Model Editing. *arXiv preprint arXiv:2502.11647*.

Xiong, C.; Chen, P.-Y.; and Ho, T.-Y. 2025. CoP: Agentic Red-teaming for Large Language Models using Composition of Principles. *arXiv preprint arXiv:2506.00781*.

Xu, J.; Ju, D.; Li, M.; Boureau, Y.-L.; Weston, J.; and Dinan, E. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2950–2968.

Xu, Q.; Tian, Z.; Wu, H.; Huang, Z.; Liang, D.; Song, Y.; Pan, Z.; Liu, F.; and Li, D. 2025. Disguise While Defensing: Avoid Refusal Responses in LLM's Defense via a Multi-agent Attacker-Disguiser Game. *IEEE Transactions on Information Forensics and Security*, 1–1.

Xu, Z.; Jiang, F.; Niu, L.; Jia, J.; Lin, B. Y.; and Poovendran, R. 2024. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. *arXiv preprint arXiv:2402.08983*.

Yang, R.; Pan, X.; Luo, F.; Qiu, S.; Zhong, H.; Yu, D.; and Chen, J. 2024. Rewards-in-Context: Multi-objective Alignment of Foundation Models with Dynamic Preference Adjustment. *International Conference on Machine Learning*.

Yang, Y.; Dan, S.; Li, S.; Roth, D.; and Lee, I. 2025. MR. Guard: Multilingual Reasoning Guardrail using Curriculum Learning. *arXiv e-prints*, arXiv–2504.

Yi, S.; Liu, Y.; Sun, Z.; Cong, T.; He, X.; Song, J.; Xu, K.; and Li, Q. 2024. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Youssef, P.; Zhao, Z.; Braun, D.; Schlötterer, J.; and Seifert, C. 2025. Position: Editing large language models poses serious safety risks. *arXiv preprint arXiv:2502.02958*.

Yu, L.; Do, V.; Hambardzumyan, K.; and Cancedda, N. 2024. Robust LLM safeguarding via refusal feature adversarial training. *arXiv preprint arXiv:2409.20089*.

Yuan, Y.; Jiao, W.; Wang, W.; Huang, J.-t.; Xu, J.; Liang, T.; He, P.; and Tu, Z. 2024. Refuse whenever you feel unsafe: Improving safety in llms via decoupled refusal training. *arXiv preprint arXiv:2407.09121*.

Yuan, Y.; Jiao, W.; Wang, W.; tse Huang, J.; Xu, J.; Liang, T.; He, P.; and Tu, Z. 2025. Refuse Whenever You Feel Unsafe: Improving Safety in LLMs via Decoupled Refusal Training. arXiv:2407.09121.

Zhang, J.; Zhou, Y.; Liu, Y.; Li, Z.; and Hu, S. 2024. Holistic automated red teaming for large language models through top-down test case generation and multi-turn interaction. *arXiv preprint arXiv:2409.16783*.

Zhang, S.; Zhai, Y.; Guo, K.; Hu, H.; Guo, S.; Fang, Z.; Zhao, L.; Shen, C.; Wang, C.; and Wang, Q. 2025a. Jbshield: Defending large language models from jailbreak attacks through activated concept analysis and manipulation. *arXiv preprint arXiv:2502.07557*.

Zhang, Y.; Liu, T.; Zhao, Z.; Meng, G.; and Chen, K. 2025b. Align in depth: Defending jailbreak attacks via progressive answer detoxification. *arXiv preprint arXiv:2503.11185*.

Zhang, Y.; Zhang, S.; Huang, Y.; Xia, Z.; Fang, Z.; Yang, X.; Duan, R.; Yan, D.; Dong, Y.; and Zhu, J. 2025c. Stair: Improving safety alignment with introspective reasoning. *arXiv preprint arXiv:2502.02384*.

Zhang, Z.; Zhang, Q.; and Foerster, J. 2024. Parden, can you repeat that? defending against jailbreaks via repetition. *arXiv preprint arXiv:2405.07932*.

Zhao, W.; Li, Z.; Li, Y.; Zhang, Y.; and Sun, J. 2024. Defending large language models against jailbreak attacks via layer-specific editing. *arXiv preprint arXiv:2405.18166*.

Zhao, X.; Cai, W.; Shi, T.; Huang, D.; Lin, L.; Mei, S.; and Song, D. 2025. Improving llm safety alignment with dual-objective optimization. *arXiv preprint arXiv:2503.03710*.

Zheng, C.; Yin, F.; Zhou, H.; Meng, F.; Zhou, J.; Chang, K.-W.; Huang, M.; and Peng, N. 2024a. On Prompt-Driven Safeguarding for Large Language Models. arXiv:2401.18018.

Zheng, C.; Yin, F.; Zhou, H.; Meng, F.; Zhou, J.; Chang, K.-W.; Huang, M.; and Peng, N. 2024b. On prompt-driven safeguarding for large language models. *arXiv preprint arXiv:2401.18018*.

Zhou, Z.; Yu, H.; Zhang, X.; Xu, R.; Huang, F.; and Li, Y. 2024. How alignment and jailbreak work: Explain llm safety through intermediate hidden states. *arXiv preprint arXiv:2406.05644*.