SPARD: Defending Harmful Fine-Tuning Attack via Safety Projection with Relevance–Diversity Data Selection

Shuhao Chen [1 2] **Weisen Jiang** [3] **Yeqi Gong** [4] **Shengda Luo** [5] **Chengxiang Zhuo** [4] **Zang Li** [4] **James T. Kwok** [2] **Yu Zhang** [1]

Abstract

Fine-tuning large language models often undermines their safety alignment, a problem further amplified by harmful fine-tuning attacks in which adversarial data removes safeguards and induces unsafe behaviors. We propose **SPARD**, a novel defense framework that integrates **S**afety-**P**rojected **A**lternating optimization with **R**elevance-**D**iversity aware data selection. SPARD optimizes alternatively between utility updates and explicit safety projections with a set of safe data to enforce safety constraints. To curate safe data, we propose a Relevance–Diversity Determinantal Point Process to select compact safe data, balancing task relevance and safety coverage. Experiments on GSM8K and OpenBookQA under four harmful fine-tuning attacks demonstrate that SPARD consistently achieves the lowest average attack success rates, substantially outperforming state-of-the-art defense methods, while maintaining high task accuracy. Code is available at https://github.com/shuhao02/SPARD.

1. Introduction

Large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023; An et al., 2024; Meta, 2024; Jiang et al., 2024) have shown strong capabilities across a wide range of tasks, making them increasingly popular in real-world applications (Jiang et al., 2023; Wei et al., 2024; Chen et al., 2024). Fine-tuning-as-a-service has become a common way for users to adapt LLMs to specific downstream domains via service providers. However, fine-tuning can inadvertently undermine safety alignment, causing models to forget their safeguards (Qi et al., 2024; Yang et al., 2023; Lermen et al., 2023). This problem becomes more severe when fine-tuning data contains malicious or adversarial content, as in harmful fine-tuning attacks (Liu et al., 2023; Zou et al., 2023; Huang et al., 2024b; Jiang & Pan, 2025), which can effectively strip away safety mechanisms and cause the model to produce unsafe outputs.

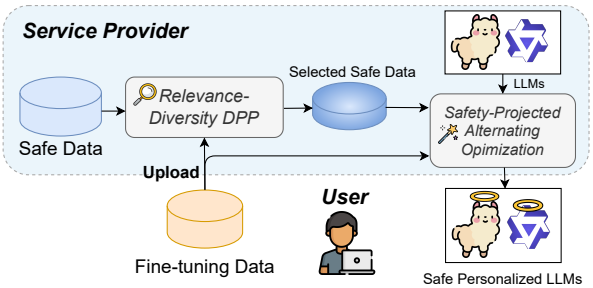


Figure 1. Illustration of SPARD.

Recently, many defense methods have been proposed to counter the harmful fine-tuning attacks. For example, PTST (Lyu et al., 2024) and SafeLoRA (Hsu et al., 2024) mitigate harmful updates by re-injecting safety prompts or constraining LoRA adapters, but they either rely on carefully crafted prompt templates or require structural constraints that limit general applicability. Other approaches exploit safe data as an implicit safeguard. For instance, SafeInstr (Bianchi et al., 2024) mixes a small fraction of safe examples into fine-tuning data to counter harmful behaviors, while Lisa (Huang et al., 2024c) uses a bi-state optimization with safe samples and applies a proximal term to constrain the drift of each state. However, those methods suffer from two key drawbacks: (i) they treat safe data merely as a soft regularizer, which provides only weak control and makes it difficult to balance safety with downstream utility; and (ii) they typically select safe samples randomly, overlooking a fact that more relevant safe data can provide stronger corrective signals against harmful fine-tuning (as shown in Section 3.2). This motivates the need for a more principled defense and data selection method to robustly withstand harmful fine-tuning.

[1]Department of Computer Science and Engineering, Southern University of Science and Technology [2]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology [3]Department of Computer Science and Engineering, The Chinese University of Hong Kong [4]Platform and Content Group, Tencent [5]Chinese Medicine Guangdong Laboratory. Correspondence to: Yu Zhang <yu.zhang.ust@gmail.com>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

To address these challenges, we propose **SPARD**, a novel framework that defends aligned LLMs against harmful fine-tuning attacks by combining **S**afety-**P**rojected **A**lternating optimization with **R**elevance-**D**iversity aware data selection. Figure 1 illustrates the overall procedure of SPARD. As shown, SPARD consists of two complementary components. First, we study the safety-constrained fine-tuning problem, and introduce *Safety-Projected Alternating Gradient* (SPAG), an optimization strategy that alternates between utility-driven updates on the fine-tuning data and explicit safety projections onto a constraint set defined by safe data. Unlike penalty-based approaches, SPAG enforces feasibility in a closed form, ensuring that safety alignment is preserved throughout training. Second, we recognize that the effectiveness of the safety projection critically depends on the choice of safe data. That is, not all safe samples are equally informative: samples that align closely with the downstream task could provide stronger corrective signals than others. To this end, we develop a *Relevance–Diversity Determinantal Point Process (DPP)* that selects a compact subset of safe data to balance the task relevance and behavioral diversity, ensuring broad and effective coverage against harmful attacks. Together, those two components yield a principled defense framework that maintains downstream utility while robustly constraining unsafe behaviors.

We conduct experiments on GSM8K (Cobbe et al., 2021) and OpenbookQA (Mihaylov et al., 2018) with four harmful finetuning attacks to evaluate both the utility and safety of SPARD. Empirical results show that SPARD can effectively mitigate harmful behaviors, achieving the lowest average Attack Success Rate (ASR) with high downstream accuracy. Moreover, SPARD significantly outperforms SafeInstr on average in ASR, showing the effectiveness of SPAG and Relevance–Diversity DPP.

Our contributions are summarized as follows: (i) We propose SPARD, a novel defense framework that integrates safety-projected optimization with relevance–diversity–aware data selection to robustly defend aligned LLMs against harmful fine-tuning. (ii) We introduce SPAG, *a principled optimization method* solving the novel safety-constrained fine-tuning problems that alternates between utility updates and explicit safety projections, and *a Relevance–Diversity DPP* to select compact, task-aligned, and diverse safety subsets. (iii) Through extensive experiments on two aligned LLMs, multiple downstream tasks, and diverse attack datasets, we show that SPARD consistently outperforms existing defense methods in reducing the attack success rates while maintaining the utility.

2. Related Works

LLM Safety and Alignment. Ensuring that large language models (LLMs) behave in a safe and helpful manner is a central challenge in AI research (Yao et al., 2024). Recent foundation models such as LLaMA (Meta, 2024; Touvron et al., 2023) and Qwen (An et al., 2024) have been aligned with safety guardrails to reject harmful instructions and follow user intent more reliably. A common paradigm for alignment is preference-based learning, most notably Reinforcement Learning from Human Feedback (Ouyang et al., 2022; Ziegler et al., 2019; Bai et al., 2022; Lin et al., 2025), which optimizes models to maximize human-preferred responses. Subsequent work has proposed more efficient formulations, such as Direct Preference Optimization (Rafailov et al., 2023), reward-free methods like RRHF (Yuan et al., 2023b), which reduce reliance on expensive reward models while maintaining alignment quality. However, a critical vulnerability remains: the safety alignment achieved through these expensive procedures is often brittle and can be easily compromised or erased through subsequent downstream fine-tuning (Yang et al., 2023; Yi et al., 2024; Qi et al., 2024; Lermen et al., 2023; Zhan et al., 2023; Hsiung et al., 2025; Li et al., 2025), which motivates the need for robust defense mechanisms.

Defending Against Harmful Fine-tuning. Harmful fine-tuning attacks (Huang et al., 2024b; Liu et al., 2023; Zou et al., 2023; Yuan et al., 2023a) compromise aligned models by poisoning training data with adversarial prompts that bypass safety guardrails. To counter such risks, many defense strategies (Huang et al., 2024a;d; Liu et al., 2025a; Chen et al., 2025; Lyu et al., 2024; Hsu et al., 2024; Bianchi et al., 2024; Huang et al., 2024c; Yi et al., 2025; Jiang & Pan, 2025) have been proposed. For example, MetaDefense (Jiang & Pan, 2025) defends against finetuning-based jailbreak attacks by training a single LLM to predict the harmfulness of both incoming queries before generation and partial responses during generation. PTST (Lyu et al., 2024) avoids safety degradation by fine-tuning solely on task data and re-introducing safety prompts at inference time. SafeLoRA (Hsu et al., 2024) constrains harmful updates by projecting LoRA weights from selected layers into a safety-aligned subspace. Other approaches leverage safe data as an implicit safeguard. For example, SafeInstr (Bianchi et al., 2024) mixes a small fraction of safe examples into fine-tuning data, while Lisa (Huang et al., 2024c) uses safe samples by a bi-state optimization with a proximal regularization term. Although effective to some extent, these methods either rely on carefully tuned penalty weights or only weakly address the utility–safety tradeoff. In contrast, our proposed SPAG provides an optimization-grounded solution by explicitly projecting the model back into the safe region. This adaptive projection automatically determines the correction size, removing the need for manual weight tuning while simultaneously preserving downstream utility.

Data Selection for LLMs. The quality and composition of training data are critical determinants of LLM performance (Zhou et al., 2023; Gadre et al., 2023). This has motivated a growing body of work on data selection (Albalak et al., 2024), which seeks to curate smaller yet more effective subsets from vast, noisy corpora. Selection strategies span a broad spectrum: filtering based on perplexity or linguistic complexity (Longpre et al., 2024), identifying core sets that approximate the full dataset's training dynamics (Sorscher et al., 2022), or leveraging embedding similarity to retrieve samples closer to the target distribution for task-specific fine-tuning (Liu et al., 2021; Xia et al., 2024; Hsiung et al., 2025; Liu et al., 2025b). However, relevance-based selection alone often leads to redundancy. To mitigate this, we propose a novel approach to achieve a relevance-diversity trade-off.

Determinantal Point Processes (DPPs) (Macchi, 1975; Kulesza et al., 2012; Chen et al., 2018) provide a principled probabilistic framework for subset selection that naturally achieves diversity. Given a candidate pool $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and a positive semidefinite kernel matrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ with the kernel $\mathbf{L}_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathcal{K}(\cdot,\cdot)$ is the kernel function encoding similarity. DPP assigns probability to each subset $\mathcal{C} \subseteq \mathcal{X}$ as

$$\mathbb{P}(\mathcal{C}) = \frac{\det(\mathbf{L}_{\mathcal{C}})}{\det(\mathbf{I}+\mathbf{L})}, \tag{1}$$

where $\mathbf{L}_{\mathcal{C}}$ is the principal submatrix indexed by $\mathcal{C}$, $\mathbf{I}$ is the identity matrix, and $\det(\cdot)$ is the determinant of a matrix. The denominator $\det(\mathbf{I}+\mathbf{L})$ is a constant independent of the subset selection, thus can be ignored. Intuitively, $\det(\mathbf{L}_{\mathcal{C}})$ corresponds to the squared volume spanned by the feature vectors of $\mathcal{C}$, favoring subsets whose elements are both individually informative and mutually dissimilar. Yet, traditional DPPs ignore task relevance. Our work closes this gap by introducing a *Relevance–Diversity DPP*, which incorporates task-relevance quality scores directly into the DPP kernel.

3. Methodology

3.1. Safety-Projected Alternating Gradient (SPAG)

In the proposed SPARD method, we formulate fine-tuning as a *safety-constrained optimization problem* to adapt the model to downstream data without compromising its safety alignment:

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}(\mathcal{D}_{\text{ft}}, \boldsymbol{\theta}) \quad \text{s.t.} \quad \mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}) \le \tau, \tag{2}$$

where $\mathcal{D}_{\text{ft}}$ is the fine-tuning dataset, $\mathcal{D}_{\text{safe}}$ is the safety dataset, and τ is a predefined threshold. In practice, τ can be set by measuring the average safety loss of the pretrained LLM on $\mathcal{D}_{\text{safe}}$.

A common approach (Huang et al., 2024c; Yi et al., 2025; Bianchi et al., 2024) to relax the constraint is to add it to the objective as a penalty term: $\min_{\boldsymbol{\theta}} \ \mathcal{L}(\mathcal{D}_{\text{ft}}, \boldsymbol{\theta}) + \lambda\big(\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}) - \tau\big)$, with a penalty parameter $\lambda \ge 0$ controlling the balance between utility and safety. However, this blending lacks explicit control on safety, since the constraint is only enforced indirectly through the objective.

Instead of implicitly encouraging safety via a soft penalty, we directly enforce the constraint using a projection-based strategy. The key idea is to first perform a utility-driven update (using $\mathcal{D}_{\text{ft}}$) and then project the updated parameters back into a region where the safety constraint is approximately satisfied. This alternating update scheme avoids the difficulty of choosing penalty weights, while providing a principled geometric correction that guarantees feasibility up to first order.

Specifically, after a utility update $\boldsymbol{\theta}^+ = \boldsymbol{\theta} - \eta_{\text{ft}} \nabla \mathcal{L}(\mathcal{D}_{\text{ft}}, \boldsymbol{\theta})$, where η_{ft} is the learning rate for the utility step and $\nabla \mathcal{L}(\mathcal{D}_{\text{ft}}, \boldsymbol{\theta})$ denotes the gradient, we project $\boldsymbol{\theta}^+$ back into a linearized safety region determined by the safety loss. Using a first-order Taylor expansion at $\boldsymbol{\theta}^+$, we approximate $\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta})$ as $\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}) \approx \mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^+) + \langle \mathbf{g}_{\text{safe}}, \boldsymbol{\theta} - \boldsymbol{\theta}^+ \rangle$, where $\mathbf{g}_{\text{safe}} = \nabla \mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^+)$. This defines the half-space $\mathcal{C}^+ = \{\boldsymbol{\theta} : \ \mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^+) + \langle \mathbf{g}_{\text{safe}}, \boldsymbol{\theta} - \boldsymbol{\theta}^+ \rangle \le \tau\}$, which is a local approximation of the feasible set around $\boldsymbol{\theta}^+$.

The projection step seeks the point in $\mathcal{C}^+$ that is closest to $\boldsymbol{\theta}^+$:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \|\boldsymbol{\theta} - \boldsymbol{\theta}^+\|^2 \\ \text{s.t.} \quad & \mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^+) + \langle \mathbf{g}_{\text{safe}}, \boldsymbol{\theta} - \boldsymbol{\theta}^+ \rangle \le \tau, \end{aligned} \tag{3}$$

Introducing multipliers for the constraint in Eq. (3), the KKT conditions (Bertsekas, 1997) give the projected solution as

$$\boldsymbol{\theta}^{\text{new}} = \begin{cases} \boldsymbol{\theta}^+, & \text{if } \mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^+) \le \tau, \\ \boldsymbol{\theta}^+ - \frac{\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^+) - \tau}{\|\mathbf{g}_{\text{safe}}\|^2} \mathbf{g}_{\text{safe}}, & \text{otherwise.} \end{cases} \tag{4}$$

The detailed derivation is provided in Appendix A. When the safety condition is violated, to stabilize training by avoiding arbitrarily large projections, we adopt the trust-region optimization strategy (Schulman et al., 2015) to limit the step size as $\alpha = \min\Big(\frac{\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^+) - \tau}{\|\mathbf{g}_{\text{safe}}\|^2}, \ \eta_{\text{safe}}\Big)$, and then $\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^+ - \alpha\, \mathbf{g}_{\text{safe}}$, where η_{safe} is a trust-region radius. The trust-region constraint ensures the updated model $\boldsymbol{\theta}^{\text{new}}$ stays within a ball centered at $\boldsymbol{\theta}^+$ with radius $\eta_{\text{safe}} \|\mathbf{g}_{\text{safe}}\|$. According to Eq. (4), the update either keeps $\boldsymbol{\theta}^+$ if already safe, or applies a corrective step along $\mathbf{g}_{\text{safe}}$ with magnitude determined by projection and trust region.

The algorithm of SPAG is illustrated in Algorithm 1. In summary, SPAG provides a simple yet principled mechanism for

Algorithm 1 SPAG.

Require: Fine-tuning dataset $\mathcal{D}_{\text{ft}}$, safety dataset $\mathcal{D}_{\text{safe}}$, learning rate η_{ft}, threshold τ, trust-region radius $\eta_{\text{safe}} > 0$; parameters $\boldsymbol{\theta}$;

while not converged **do**
  /* Utility fine-tuning */
  Sample mini-batch $\mathcal{B}_{\text{ft}} \subseteq \mathcal{D}_{\text{ft}}$;
  $\boldsymbol{\theta}^{+} \leftarrow \boldsymbol{\theta} - \eta_{\text{ft}} \nabla \mathcal{L}(\mathcal{B}_{\text{ft}}, \boldsymbol{\theta})$;
  /* Safety projection */
  Sample mini-batch $\mathcal{B}_{\text{safe}} \subseteq \mathcal{D}_{\text{safe}}$;
  $\ell_{\text{safe}} \leftarrow \mathcal{L}(\mathcal{B}_{\text{safe}}, \boldsymbol{\theta}^{+})$;
  if $\ell_{\text{safe}} \leq \tau$ **then**
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{+}$;
  else
    $\mathbf{g}_{\text{safe}} \leftarrow \nabla \mathcal{L}(\mathcal{B}_{\text{safe}}, \boldsymbol{\theta}^{+})$;
    $\alpha \leftarrow \min\left(\frac{\ell_{\text{safe}} - \tau}{\|\mathbf{g}_{\text{safe}}\|^2}, \eta_{\text{safe}}\right)$;
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{+} - \alpha\, \mathbf{g}_{\text{safe}}$;
  end if
end while

output Safety-aligned parameters $\boldsymbol{\theta}$.

safety-constrained fine-tuning: it alternates between utility optimization and explicit safety projection. Geometrically, the method corrects each update by projecting onto a safety half-space, thereby directly solving the constraint rather than relying on soft penalties. Unlike penalty-based approaches, which require careful tuning of λ and offer no guarantee of feasibility, SPAG yields a closed-form projection step that enforces the constraint up to the first order. This combination of interpretability, guaranteed feasibility, and hyperparameter-free correction makes SPAG both practical and robust for safety-aligned fine-tuning.

3.2. Relevance- and Diversity-Aware Safety Data Selection

While SPAG provides a principled projection mechanism for enforcing safety constraints, its success fundamentally depends on the quality of the safety dataset $\mathcal{D}_{\text{safe}}$. Recent studies have also observed that relevant safety data can improve safety during training by leveraging embedding similarity (Hsiung et al., 2025), employing trained selectors (Liu et al., 2025b), or matching task styles and formats (Eiras et al., 2024; Xiao et al., 2025). Here we claim that *not all safety samples contribute equally*. That is, safety samples, which align well with the fine-tuning domain $\mathcal{D}_{\text{ft}}$, could provide stronger corrective signals than other safety samples. Therefore, carefully selecting *relevant* safety data becomes critical to ensuring that the projection step effectively constrains the model.

Relevance Improves Safety. To better understand the role of relevance, we conduct an experiment, where the GSM8K training data are merged with 10% BeaverTails attack data (Ji et al., 2023) as $\mathcal{D}_{\text{ft}}$, and safe samples are selected from BeaverTails and LatHarmful (Sheshadri et al., 2024) defense sets according to the similarity with $\mathcal{D}_{\text{ft}}$. We define the similarity quality of a candidate $\mathbf{x}_i \in \mathcal{D}_{\text{safe}}$ as

$$q_i = \max_{\mathbf{x}_z \in \mathcal{D}_{\text{ft}}} \mathsf{sim}(\mathbf{x}_i, \mathbf{x}_z), \tag{5}$$

where $\mathsf{sim}(\cdot, \cdot)$ is cosine similarity in the embedding space. As shown in Figure 2, the attack success rate (ASR) decreases sharply as the average similarity of selected samples increases, dropping from 68.8% at low similarity to 11.4% at moderate-to-high similarity. This confirms that task-relevant safe samples provide stronger constraints and substantially enhance robustness.

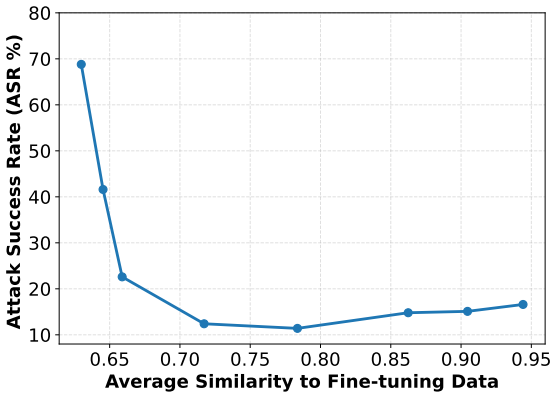


Figure 2. Attack success rate (ASR) with varying similarity levels of $\mathcal{D}_{\text{safe}}$ to $\mathcal{D}_{\text{ft}}$. The fine-tuning data are merged with BeaverTails attack data, and $\mathcal{D}_{\text{safe}}$ are sampled from BeaverTails and LatHarmful defense data.

Notably, the curve also shows that ASR rises again (to 16.6%) when the selected samples are *too similar* to the fine-tuning data (e.g., average similarity ≈ 0.94). This counterintuitive phenomenon occurs because extreme similarity introduces redundancy: the selected safety samples cover only a narrow region of the risk space, leaving other harmful behaviors underrepresented. Consequently, the model may overfit to a small set of highly similar constraints, reducing the effectiveness of safety alignment.

Relevance–Diversity DPP. The above observation motivates a selection strategy that balances the *relevance* and *diversity*: the relevance ensures that the chosen samples provide strong and task-aligned safety signals, while the diversity ensures broad coverage of distinct harmful behaviors. While Hsiung et al. (2025) introduces a metric for assessing subset diversity, they do not integrate this metric into the selection process, so their method cannot promote diversity during data selection. To achieve this, we extend Determinantal Point Processes (DPPs) (Macchi, 1975; Kulesza et al., 2012; Ye et al., 2023), which naturally promote di-

versity through determinant-based subset probabilities, by incorporating the relevance into the kernel design.

Specifically, given relevance scores q_i (defined in Eq. (5)) for candidates $\mathbf{x}_i \in \mathcal{D}_{\text{safe}}$, We then construct the kernel as

$$\widehat{\mathcal{K}}_{ij} = (q_i \cdot q_j)^{\beta} \cdot \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j), \tag{6}$$

where $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ captures the intrinsic similarity between two safety samples (e.g., the cosine similarity in an embedding space), and $\beta \geq 0$ controls the influence of the relevance. Intuitively, $(q_i \cdot q_j)^{\beta}$ acts as a multiplicative weight that increases the likelihood of including pairs of samples that are both highly relevant to the fine-tuning distribution (a sensitive analysis of β is provided in Section 4.3). When $\beta = 0$, the kernel reduces to the classical diversity-only DPP, while larger β biases the distribution toward relevance-aware subsets.

Let $\mathbf{L}$ and $\widehat{\mathbf{L}}$ be the kernel matrix corresponding to $\mathcal{K}$ and $\widehat{\mathcal{K}}$, respectively. For any subset $\mathcal{C} \subseteq \mathcal{D}_{\text{safe}}$, according to Eq. (1), the selection probability is given by

$$\mathbb{P}(\mathcal{C}) \propto \det(\widehat{\mathbf{L}}_{\mathcal{C}}) = \prod_{\mathbf{x}_i \in \mathcal{C}} q_i^{2\beta} \cdot \det(\mathbf{L}_{\mathcal{C}}), \tag{7}$$

where $\mathbf{L}_{\mathcal{C}}$ denotes the kernel submatrix with indices in $\mathcal{C}$. This decomposition makes the roles explicit: the factor $\prod q_i^{2\beta}$ rewards subsets containing highly relevant samples, while $\det(\mathbf{L}_{\mathcal{C}})$ enforces diversity among them. Thus, the relevance–diversity DPP jointly balances the task alignment and coverage, ensuring that the selected safety set is neither irrelevant nor redundant.

Efficient Greedy Selection. Although DPPs define a principled probability distribution, exactly solving the maximum a posteriori (MAP) problem $\arg\max_{\mathcal{C} \subseteq \mathcal{D}_{\text{safe}}} \det(\widehat{\mathbf{L}}_{\mathcal{C}})$ is computationally expensive, requiring to calculate determinants of many submatrices. To scale the selection to large safety datasets, we adopt a greedy approximation that incrementally builds the subset by adding one sample at a time, each chosen to maximize the marginal gain in the determinant.

Specifically, suppose we have already selected $\mathcal{C}_{m-1}$ after $m-1$ steps. For a candidate $i \notin \mathcal{C}_{m-1}$, the expanded kernel matrix is $\widehat{\mathbf{L}}_{\mathcal{C}_{m-1} \cup \{i\}} = \begin{pmatrix} \widehat{\mathbf{L}}_{\mathcal{C}_{m-1}} & \mathbf{v}_i \\ \mathbf{v}_i^{\top} & \mathbf{L}_{ii} \end{pmatrix}$, where $\mathbf{v}_i$ contains kernel similarities between $\mathbf{x}_i$ and the already-selected set $\mathcal{C}_{m-1}$. By the Schur complement, the determinant after including i can be factorized as

$$\det(\widehat{\mathbf{L}}_{\mathcal{C}_{m-1} \cup \{i\}}) = \det(\widehat{\mathbf{L}}_{\mathcal{C}_{m-1}}) \big(\mathbf{L}_{ii} - \mathbf{v}_i^{\top} \widehat{\mathbf{L}}_{\mathcal{C}_{m-1}}^{-1} \mathbf{v}_i\big). \tag{8}$$

The second term in the right-hand side of Eq. (8), known as the *gain factor*, measures the additional volume contributed by $\mathbf{x}_i$ that is not already spanned by $\mathcal{C}_{m-1}$. Intuitively, it rewards candidates that are both individually relevant (large $\widehat{\mathbf{L}}_{ii}$) and novel relative to the current set (small $\mathbf{v}_i^{\top} \widehat{\mathbf{L}}_{\mathcal{C}_{m-1}}^{-1} \mathbf{v}_i$).

To compute the gain efficiently, we maintain the Cholesky decomposition $\widehat{\mathbf{L}}_{\mathcal{C}_{m-1}} = \mathbf{C}\mathbf{C}^{\top}$. Then, $\mathbf{v}_i^{\top} \widehat{\mathbf{L}}_{\mathcal{C}_{m-1}}^{-1} \mathbf{v}_i = (\mathbf{C}^{-1}\mathbf{v}_i)^{\top}(\mathbf{C}^{-1}\mathbf{v}_i) = \|\mathbf{w}_i\|^2$, where $\mathbf{w}_i$ is obtained by solving the triangular system $\mathbf{C}\mathbf{w}_i = \mathbf{v}_i$. This avoids explicitly inverting $\widehat{\mathbf{L}}_{\mathcal{C}_{m-1}}$, reducing the complexity per iteration to $O(m)$ instead of cubic cost. At each step, we select $\mathbf{x}_{i^\star} = \arg\max_{\mathbf{x}_i \in \mathcal{D}_{\text{safe}} \setminus \mathcal{C}_{m-1}} \left(\widehat{\mathbf{L}}_{ii} - \|\mathbf{w}_i\|^2\right)$, and add it into $\mathcal{C}_{m-1}$ to obtain $\mathcal{C}_m$. Additional details for computational cost are provided in Appendix B.

4. Experiments

4.1. Experimental Setup

Datasets. *Safety corpora.* We use four datasets containing harmful or jailbreak-style prompts with both harmful and safe responses: (i) BeaverTails (Ji et al., 2023), a collection of safety-related QA pairs with helpfulness and harmlessness annotations; (ii) I-BeaverTails, constructed by converting BeaverTails questions into instructions using GPT-4o-mini (Hurst et al., 2024) following Bianchi et al. (2024); (iii) LatHarmful (Sheshadri et al., 2024), consisting of 5k instructions with paired harmful and harmless completions; and (iv) Q-LatHarmful, obtained by converting LatHarmful instructions into QA pairs with GPT-4o-mini. Each dataset is split 90%/10% into training and testing. From these corpora, we use *harmful queries with safe responses* from the training splits to build **GeneralSafe**, the candidate pool for selecting safety data. In contrast, to simulate harmful fine-tuning, we use *harmful queries with harmful responses* from the training splits to inject malicious samples into downstream utility tasks. Following Huang et al. (2024c); Hsu et al. (2024), the number of injected harmful samples is fixed to 10% of the original utility task training size, ensuring a consistent attack intensity across tasks.

Utility tasks. For downstream performance, we evaluate on (i) GSM8K (Cobbe et al., 2021), a benchmark of grade-school math word problems, augmented with MetaMath (Yu et al., 2024) for broader coverage; and (ii) OpenBookQA (Mihaylov et al., 2018), a science QA dataset requiring factual reasoning.

Evaluation Metrics. Following Hsu et al. (2024), we evaluate methods on two dimensions: *Safety* and *Utility*. For safety, we adopt the protocol of Qi et al. (2024), using GPT-4o-mini to judge responses under OpenAI's 11 harmful content categories. Each response receives a Harmfulness Score (HS) from 1 (safest) to 5 (most harmful), and we report the Attack Success Rate (ASR), the proportion of responses with HS > 2. For utility, we measure accuracy on downstream tasks (GSM8K or OpenBookQA), reflecting

Table 1. Defense performance of Qwen-2.5-7B-Instruct on GSM8K under four harmful fine-tuning attacks. Lower ASR/HS indicates stronger safety, while higher GSM8K accuracy reflects better utility. Best results are in **bold**. GSM8K accuracy is averaged over all four attacks.

	Beavertails		I-BeaverTails		LatHarmful		Q-LatHarmful		**Average**		GSM8K
	ASR	HS	ASR	HS	ASR	HS	ASR	HS	ASR	HS	Accuracy
Qwen-2.5-7B-Instruct	25.80	1.71	36.06	1.97	7.88	1.23	6.52	1.22	19.02	1.53	77.71
SFT	83.60	3.92	79.45	3.78	91.92	4.60	96.74	4.80	87.93	4.28	**86.77**
PTST (Lyu et al., 2024)	62.40	3.03	70.23	3.29	82.42	4.13	83.50	4.11	74.64	3.64	85.06
SafeInstr (Bianchi et al., 2024)	65.60	3.21	66.25	3.18	76.77	3.99	85.74	4.34	73.59	3.68	86.28
Lisa (Huang et al., 2024c)	24.40	1.73	35.64	2.00	**7.88**	**1.26**	8.55	**1.25**	19.12	1.56	78.45
SafeGrad (Yi et al., 2025)	32.00	1.97	41.72	2.22	20.00	1.75	27.00	1.77	30.18	1.93	85.71
SPARD	**10.00**	**1.32**	**12.58**	**1.36**	**7.88**	1.31	**7.33**	1.29	**9.45**	**1.32**	85.77

Table 2. Defense performance of LLaMA-3.2-3B-Instruct on GSM8K under four harmful fine-tuning attacks. Best results (lowest ASR/HS and highest GSM8K accuracy) are in **bold**. GSM8K accuracy is averaged over all four attacks.

	Beavertails		I-BeaverTails		LatHarmful		Q-LatHarmful		**Average**		GSM8K
	ASR	HS	ASR	HS	ASR	HS	ASR	HS	ASR	HS	Accuracy
LLaMA-3.2-3B-Instruct	41.80	2.12	52.20	2.44	16.36	1.55	12.02	1.40	30.60	1.88	62.32
SFT	87.20	4.03	79.45	3.66	98.99	4.91	99.80	4.95	91.36	4.39	72.27
PTST (Lyu et al., 2024)	58.80	2.87	64.99	3.04	97.78	4.82	96.33	4.73	79.48	3.87	**73.75**
SafeInstr (Bianchi et al., 2024)	76.20	3.61	72.54	3.46	90.10	4.56	89.21	4.52	82.01	4.04	72.21
Lisa (Huang et al., 2024c)	32.20	1.95	45.70	2.32	**9.90**	**1.36**	**8.96**	**1.29**	24.19	1.73	65.03
SafeGrad (Yi et al., 2025)	50.80	2.45	59.12	2.77	78.79	4.06	98.37	4.86	71.77	3.54	64.29
SPARD	**13.20**	**1.33**	**14.68**	**1.42**	11.31	1.43	9.16	1.35	**12.09**	**1.38**	71.23

utility performance under when safety defense is enforced.

Baselines. We compare SPARD against a range of baselines using two safe pre-trained models, Qwen-2.5-7B-Instruct (An et al., 2024) and LLaMA-3.2-3B-Instruct (Meta, 2024). Specifically, we consider: (i) SFT, which is a standard fine-tuning baseline where the model is trained exclusively on the target task data without any explicit safety measures. (ii) PTST (Lyu et al., 2024), which fine-tunes the model without safety instructions but prepends them back to inputs at inference time. (iii) SafeInstr (Bianchi et al., 2024), which randomly mixes a small fraction (3%) of safe samples into the fine-tuning dataset. (iv) Lisa (Huang et al., 2024c), which bi-state learning finetuning samples and safe samples with a proximal term to constrain the safety degradation of each state. (v) SafeGrad (Yi et al., 2025), which detects conflict safety/alignment gradients and projects out the harmful component.

Implementation Details We employ LoRA (Hu et al., 2022) for parameter-efficient fine-tuning, with a rank $r = 32$ and an alpha of 4. All models are trained using the AdamW optimizer (Loshchilov & Hutter, 2019). We set the learning rate to 5×10^{-5} for both Qwen-2.5-7B-Instruct and LLaMA-3.2-3B-Instruct. The models are fine-tuned for 10 epochs on GSM8K and 3 epochs on OpenBookQA. For our SPAG algorithm, the safety mini-batch $\mathcal{B}_{\text{safe}}$ is sampled with a batch size of 1. The trust region radius η_{safe} is set equal to the fine-tuning learning rate η_{ft}. The hyperparameters τ, δ, and ϵ are chosen as 0.2, 0.1, and 1×10^{-8} for all experiments. For our Relevance-Diversity DPP data selection, sample embeddings are generated by taking the average of the final layer's hidden states from the pretrained model (i.e., Qwen-2.5-7B-Instruct and LLaMA-3.2-3B-Instruct). Based on these embeddings, we select $\mathcal{D}_{\text{safe}}$ from the **GeneralSafe** pool, with the size fixed to 3% (i.e., $p = 0.03$) of the fine-tuning data. The relevance exponent is set to $\beta = 4$.

4.2. Main Result

Robustness to Different Attacks. Table 1 presents results on GSM8K under four harmful fine-tuning attacks. As can be seen, SPARD simultaneously achieves the lowest ASR/HS while preserving competitive GSM8K accuracy, offering the strongest balance between safety and utility among all methods.

Specifically, compared with SFT and PTST, SPARD achieves substantial gains: average ASR is reduced by over 65% and HS by 2.32 points, while accuracy remains competitive. This shows that explicit safety projection is far more effective than standard fine-tuning or inference-time prompting. Compared with SafeInstr, which randomly mixes safe samples into fine-tuning, SPARD consistently

Table 3. Defense performance of Qwen-2.5-7B-Instruct on OpenBookQA under four in-distribution harmful fine-tuning attacks. Lower ASR/HS indicates stronger safety; higher accuracy indicates better utility. OpenBookQA accuracy is averaged over the four attacks. Best results are in **bold**.

	Beavertails		I-BeaverTails		LatHarmful		Q-LatHarmful		**Average**		OpenbookQA
	ASR	HS	ASR	HS	ASR	HS	ASR	HS	ASR	HS	Accuracy
Qwen-2.5-7B-Instruct	25.80	1.71	36.06	1.97	7.88	1.23	6.52	1.22	19.02	1.53	77.60
SFT	50.20	2.70	56.39	2.81	25.66	1.96	28.92	2.05	40.29	2.38	**83.70**
PTST (Lyu et al., 2024)	37.20	2.09	56.39	2.81	10.71	1.35	16.90	1.59	30.30	1.96	83.25
SafeInstr (Bianchi et al., 2024)	51.40	2.68	59.75	2.95	26.06	1.95	29.53	2.11	41.69	2.42	84.15
Lisa (Huang et al., 2024c)	26.00	1.74	34.80	1.93	**7.07**	**1.22**	**7.33**	**1.24**	18.80	1.53	78.90
SafeGrad (Yi et al., 2025)	28.80	1.80	36.29	2.01	7.68	1.24	9.78	1.32	20.63	1.59	83.30
SPARD	**21.80**	**1.65**	**20.13**	**1.57**	8.28	1.31	7.94	1.27	**14.54**	**1.45**	83.25

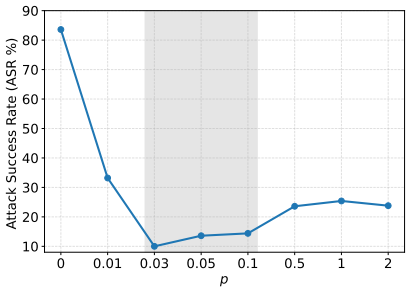


Figure 3. Effects of p.

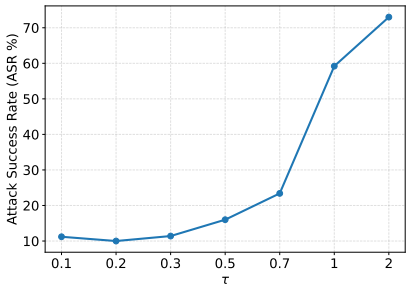


Figure 4. Effects of τ.

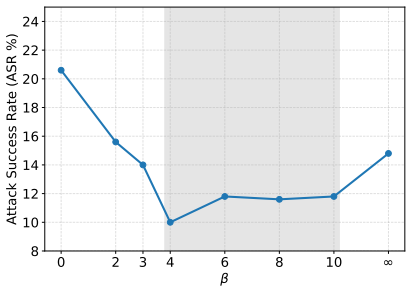


Figure 5. Effects of β.

achieves lower ASR/HS, highlighting the necessity of principled relevance–diversity selection over naive random selection. Compared with Lisa, a strong optimization-based baseline, SPARD achieves consistently better safety on all datasets, reducing ASR by 9.67%, lowering HS by 0.24, and significantly improving GSM8K accuracy by 7.32%. Compared with SafeGrad, SPARD achieves a substantially lower ASR of 20.73% while having a slightly better downstream performance. This demonstrates that the key designs of SPARD—SPAG safety projection and relevance–diversity DPP selection—are crucial for constraining harmful behaviors while preserving downstream task performance.

Generalization Across Architectures (LLaMA). Table 2 presents results on LLaMA-3.2-3B-Instruct. We observe that the overall safety degradation is more severe on LLaMA than on Qwen, as SFT and PTST both yield very high ASR/HS despite maintaining task accuracy. SPARD, however, remains effective across model families, achieving the lowest ASR (12.09%) and HS (1.38) while keeping accuracy competitive (71.23%). Compared with SFT and PTST, SPARD lowers ASR by over 67%, confirming that its safety projection generalizes across backbones. Against SafeInstr, which suffers from high ASR/HS, SPARD shows the value of relevance–diversity selection over naive random mixing. Relative to Lisa, SPARD further improves both safety and utility (-12.10% ASR, -0.35 HS, $+6.20\%$ accuracy). SPARD also outperforms SafeGrad, reducing ASR by over 59% while achieving a substantial 6.94% improvement in accuracy. Together, these results demonstrate that SPAG optimization and DPP-based selection enhance robustness consistently across architectures.

Generalization to OpenBookQA. Table 3 reports results on OpenBookQA under four harmful fine-tuning attacks. SPARD achieves the lowest ASR (14.54%) and HS (1.45) while preserving strong task accuracy (83.25%), confirming that its effectiveness extends beyond math reasoning tasks. Compared with SFT and PTST, SPARD reduces ASR by more than 15.3% on average, showing that explicit safety projection remains effective for science QA. Relative to SafeInstr, which again suffers from high ASR/HS, SPARD demonstrates the importance of relevance–diversity selection over naive data mixing. Compared with Lisa, SPARD achieves lower ASR/HS (-4.14%/-0.05) and higher accuracy ($+4.05\%$), reinforcing that its joint use of SPAG optimization and DPP-based selection improves both safety and utility. Finally, SPARD outperforms SafeGrad with an improvement of 5% on ASR, validating the effectiveness of SPAG's safety projection. These results highlight that SPARD generalizes beyond GSM8K to diverse downstream reasoning tasks, maintaining robustness across domains.

4.3. Analysis

Effects of safe sample ratio. Figure 3 shows the impact of varying the ratio p of safe samples added to the GSM8K fine-tuning data using Qwen-2.5-7B-Instruct under the BeaverTails attack. When $p = 0$ (i.e., no safe samples are added), the model is highly vulnerable with ASR above 80%. As p increases, ASR drops sharply and reaches the lowest point around $p \in [0.03, 0.05]$. Beyond this range, further increas-

Table 4. Effect of the trust-region radius η_{safe} of Qwen-2.5-7B-Instruct on GSM8K.

Method	BeaverTails	I-BeaverTails	LatHarmful	Q-LatHarmful	**Avg. ASR**	GSM8K Acc
SPARD (w/o η_{safe})	8.20	10.69	0.00	1.22	5.03	81.92
SPARD	10.00	12.58	7.88	7.33	9.45	85.77

Table 5. Effect of Relevance-Diversity DPP under different harmful fine-tuning attacks. Best results are in **bold**.

	Beavertails		I-BeaverTails		LatHarmful		Q-LatHarmful		**Average**		GSM8K
	ASR	HS	ASR	HS	ASR	HS	ASR	HS	ASR	HS	Accuracy
Lisa (Huang et al., 2024c)	24.40	1.73	35.64	2.00	7.88	1.26	8.55	1.25	19.12	1.56	78.45
SPAG *w/* Random	15.80	1.47	22.64	1.70	13.33	1.52	11.61	1.46	15.85	1.54	85.06
SPAG *w/* Max Quality	16.60	1.53	24.32	1.71	16.16	1.63	8.96	1.34	16.51	1.55	85.69
SPARD	**10.00**	**1.32**	**12.58**	**1.36**	**7.88**	**1.31**	**7.33**	**1.29**	**9.45**	**1.32**	**85.77**

ing p leads to diminishing returns and even slight degradation due to the inclusion of redundant or less relevant samples. This indicates that a small but carefully chosen proportion $p \in [0.03, 0, 1]$ of safe samples is sufficient to provide strong safety guarantees without overwhelming the fine-tuning objective.

Effects of τ. We study the effects of the safety threshold τ on GSM8K using Qwen-2.5-7B-Instruct under the BeaverTails attack. As shown in Figure 4, small values of τ enforce strict safety constraints, effectively suppressing ASR, but overly conservative thresholds ($\tau > 0.5$) begin to harm the balance and allow ASR to rise again. In practice, we can set the τ by referencing the average loss of the aligned LLM on the safety benchmark.

Effects of β. To analyze the effect of relevance exponent β, we conduct experiments with the BeaverTails attack on GSM8K using Qwen-2.5-7B-Instruct. Figure 5 analyzes the relevance exponent β, which balances the weight between relevance and diversity in the DPP kernel. SPARD is relatively robust to a wide range of moderate values ($\beta \in [4, 10]$), achieving the lowest ASR, while very small β underemphasizes relevance and very large β collapses diversity, both leading to weaker defenses. These results confirm that both relevance and diversity should be considered in the data selection process.

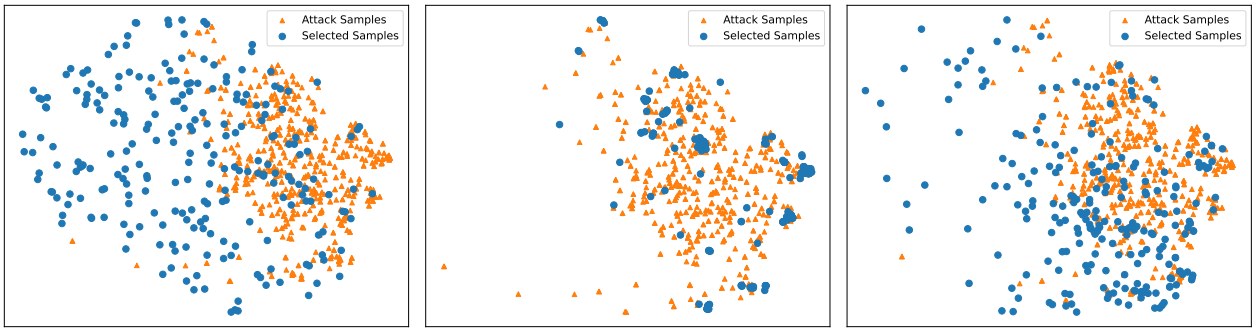
Effects of η_{safe}. We conduct an ablation removing the trust-region limit η_{safe}. Table 4 reports the ASR under multiple attacks and the average downstream GSM8K accuracy. Removing η_{safe} results in more aggressive projection updates: safety improves for some attacks, but downstream utility degrades substantially. These results highlight that η_{safe} plays a crucial role in stabilizing the projection step, achieving a more balanced trade-off between strong safety and good downstream performance.

Effect of Relevance-Diversity DPP. To study the effect of Relevance-Diversity DPP, we compare it with (i) SPAG *w/* Random, which randomly selects samples from General-Safe as $\mathcal{D}_{\text{safe}}$. (ii) SPAG *w/* Max Quality, which selects the samples with the highest quality score as $\mathcal{D}_{\text{safe}}$. As shown in Table 5, SPAG *w/* Random surpasses previous SOTA (i.e., Lisa) with an average ASR reduction of 3.27% and an GSM8K accuracy improvement of 6.61%, validating the effectiveness of SPAG safety projection. Compared with all variants, SPARD has the best average safety and utility, achieving the lowest mean ASR/HS (9.45%/1.32) and the highest GSM8K accuracy (85.77%). Specifically, SPARD outperforms SPAG *w/* Random with a noticeable ASR and HS reduction of 6.40% and 0.22, showing that selecting relevant data can substantially improve safety. Additionally, SPARD surpasses SPAG *w/* Max Quality by a large margin of 7.06% on average ASR, validating that diversity is equally crucial. By balancing both relevance and diversity, SPARD achieves broad coverage of safety constraints while remaining task-aligned, leading to superior robustness without sacrificing utility. Moreover, as Hsiung et al. (2025) also explores similarity and diversity metrics in safety data curation, we provide further discussion on the similarities and differences, along with an empirical comparison of the two methods, in Appendix E.1.

Table 6. Defense performance with full fine-tuning on SmolLM2-1.7B-Instruct (GSM8K, BeaverTails attack). Best results are in **bold**.

Method	Accuracy	ASR	HS
SFT	**51.5**	74.0	3.61
Lisa (Huang et al., 2024c)	49.4	69.0	3.40
SafeGrad (Yi et al., 2025)	50.4	23.2	1.71
SPARD	50.6	**17.4**	**1.55**

Full Fine-Tuning without LoRA. SPAG's projection (Eq. (4)) operates on whatever parameters are being updated, independent of LoRA structure. To verify this, we evaluate SPARD with full fine-tuning on SmolLM2-1.7B-Instruct (GSM8K, BeaverTails attack). As shown in Table 6,



(a) Random. (b) Max Quality (i.e., $\beta = +\infty$). (c) SPARD.

Figure 6. Comparison of selected safe samples for GSM8K task under BeaverTails attack.

Table 7. End-to-end wall-clock time (minutes) and relative overhead for fine-tuning Qwen-2.5-7B-Instruct on GSM8K with a single A800 GPU. Overhead is measured relative to vanilla SFT.

Method	Time (min)	Overhead vs. SFT
SFT	253.0	-
PTST	253.0	+0%
SafeInstr	260.3	+2.9%
Lisa	277.8	+9.8%
SafeGrad	390.5	+54.2%
SPARD	273.5	+8.1%

SPARD achieves the lowest ASR (17.4%) and HS (1.55) while maintaining competitive accuracy, confirming that the findings are consistent without LoRA.

Visualization. Figure 6 shows the t-SNE visualization (Van der Maaten & Hinton, 2008) of selected samples for the GSM8K task under BeaverTails attacks. As shown, randomly selected data cover diverse regions but are not necessarily aligned with the attacked distribution, leading to limited safety gains. Moreover, a quality-only strategy (Max Quality) selects samples that cluster tightly around the attack distribution, but suffers from severe redundancy. In contrast, SPAG achieves a balanced selection that aligns samples closely with the attacked distribution while maintaining diversity across different safety corpora, ensuring broad coverage without redundancy. This suggests that our method is effective in selecting safe samples that are both relevant to the task and diverse (Tables 5).

Computational Overhead. The preprocessing unique to SPARD is lightweight: on a single A800 GPU, embedding extraction takes 3.42 minutes, and DPP subset selection completes in just 0.12 seconds (see Appendix B for complexity analysis). Table 7 further reports the end-to-end wall-clock time (including all preprocessing and training) for all methods fine-tuning Qwen-2.5-7B-Instruct on GSM8K with a single A800 GPU. SPARD adds only 8.1% overhead over SFT, less than Lisa (+9.8%) and substantially lower than SafeGrad (+54.2%). Given the considerable safety improvements SPARD achieves over these baselines (Tables 1–3), this marginal additional cost is well justified.

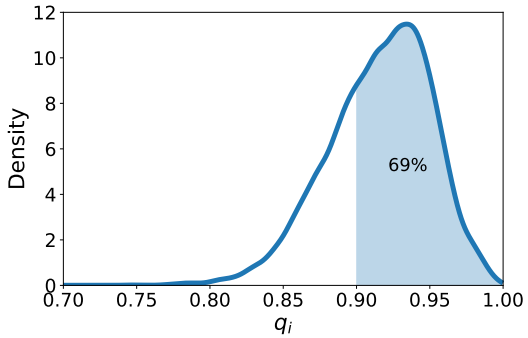


Figure 7. Distribution of similarity scores.

Why using β as an exponent? To better understand the role of β, we analyze the distribution of similarity scores q_i between GeneralSafe samples and the GSM8K dataset under the BeaverTails attack. As shown in Figure 7, most samples already exhibit very high similarity: 69% of them have $q_i > 0.9$. This heavy concentration near the upper bound makes it difficult to distinguish relative preferences using linear weighting. By introducing β as an exponent in the relevance term, we amplify subtle differences among highly similar samples, allowing the selection process to more effectively favor those that are most aligned with the target distribution.

5. Conclusion

In this paper, we propose **SPARD**, a defense framework that safeguards aligned LLMs against harmful fine-tuning by combining Safety-Projected Alternating Gradient (SPAG) with a Relevance–Diversity DPP for safe data selection. SPAG enforces safety constraints in closed form during training, while the Relevance–Diversity DPP selects task-relevant and diverse safety data to maximize coverage. Experiments on GSM8K and OpenBookQA with multiple attacks show that SPARD achieves the lowest average ASR while preserving high utility.

Acknowledgement

This work was supported by National Natural Science Foundation of China under Grant no. 62136005, Shenzhen fundamental research program JCYJ20250604144724032, the Research Grants Council of the Hong Kong Special Administrative Region (Grants 16202523 and HKU C7004-22G), Talents Cultivation Program of National Administration of Traditional Chinese Medicine (Grant No. ZYYCXTD-D-202403), the National Natural Science Foundation of China (Grant No. 82505358 and Grant No. 12326604), and the Scientific Research Start-up Funds of the Chinese Medicine Guangdong Laboratory (Grant No. HQL2025SU011).

Impact Statement

This research investigates the vulnerabilities of large language models (LLMs) to harmful fine-tuning attacks and introduces methods to strengthen their safety alignment. All datasets employed in our experiments are publicly available and widely used in the safety community. Although these datasets contain harmful or adversarial prompts, they are utilized solely for the purpose of evaluating defenses. Harmful responses are restricted to controlled experimental settings and are not disseminated beyond what is strictly necessary for reproducibility. The overarching aim of this work is to advance the safe and responsible deployment of LLMs by providing principled defense mechanisms against malicious fine-tuning. We recognize that research in this area carries potential dual-use concerns, but we believe the benefits of improving the robustness of safety alignment outweigh these risks. Our study adheres to ethical standards and prioritizes the promotion of beneficial and safe AI.

References

Albalak, A., Elazar, Y., Xie, S. M., Longpre, S., Lambert, N., Wang, X., Muennighoff, N., Hou, B., Pan, L., Jeong, H., et al. A survey on data selection for language models. Preprint arXiv:2402.16827, 2024.

An, Y., Baosong, Y., Beichen, Z., Binyuan, H., Bo, Z., Bowen, Y., Chengyuan, L., Fei, H., et al. Qwen2.5 technical report. Preprint arXiv:2412.15115, 2024.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. Preprint arXiv:2204.05862, 2022.

Bertsekas, D. P. Nonlinear programming. *Journal of the Operational Research Society*, 1997.

Bianchi, F., Suzgun, M., Attanasio, G., Rottger, P., Jurafsky, D., Hashimoto, T., and Zou, J. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *International Conference on Learning Representations*, 2024.

Chen, L., Zhang, G., and Zhou, E. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Neural Information Processing Systems*, 2018.

Chen, L., Han, X., Shen, L., Bai, J., and Wong, K.-F. Vulnerability-aware alignment: Mitigating uneven forgetting in harmful fine-tuning. Preprint arXiv:2506.03850, 2025.

Chen, S., Jiang, W., Lin, B., Kwok, J., and Zhang, Y. RouterDC: Query-based router by dual contrastive learning for assembling large language models. In *Neural Information Processing Systems*, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Christopher, H., and John, S. Training verifiers to solve math word problems. Preprint arXiv:2110.14168, 2021.

Eiras, F., Petrov, A., Torr, P. H., Kumar, M. P., and Bibi, A. Do as i do (safely): Mitigating task-specific fine-tuning risks in large language models. Preprint arXiv:2406.10288, 2024.

Gadre, S. Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., et al. Datacomp: In search of the next generation of multimodal datasets. In *Neural Information Processing Systems*, 2023.

Hsiung, L., Pang, T., Tang, Y.-C., Song, L., Ho, T.-Y., Chen, P.-Y., and Yang, Y. Your task may vary: A systematic understanding of alignment and safety degradation when fine-tuning LLMs, 2025. URL https://openreview.net/forum?id=vQ0zFYJaMo.

Hsu, C.-Y., Tsai, Y.-L., Lin, C.-H., Chen, P.-Y., Yu, C.-M., and Huang, C.-Y. Safe loRA: The silver lining of reducing safety risks when finetuning large language models. In *Neural Information Processing Systems*, 2024.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Huang, T., Hu, S., Ilhan, F., Tekin, S. F., and Liu, L. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. Preprint arXiv:2409.01586, 2024a.

Huang, T., Hu, S., Ilhan, F., Tekin, S. F., and Liu, L. Harmful fine-tuning attacks and defenses for large language models: A survey. Preprint arXiv:2409.18169, 2024b.

Huang, T., Hu, S., Ilhan, F., Tekin, S. F., and Liu, L. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. In *Neural Information Processing Systems*, 2024c.

Huang, T., Hu, S., and Liu, L. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Neural Information Processing Systems*, 2024d.

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

Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., Chen, B., Sun, R., Wang, Y., and Yang, Y. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. In *Neural Information Processing Systems*, 2023.

Jiang, W. and Pan, S. J. MetaDefense: Defending fine-tuning based jailbreak attack before and during generation. In *Neural Information Processing Systems*, 2025.

Jiang, W., Zhang, Y., and Kwok, J. Effective structured-prompting by meta-learning and representitive verbalizer. In *International Conference on Machine Learning*, 2023.

Jiang, W., Shi, H., Yu, L., Liu, Z., Zhang, Y., Li, Z., and Kwok, J. Forward-backward reasoning in large language models for mathematical verification. In *Findings of the Association for Computational Linguistics*, 2024.

Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 2012.

Lermen, S., Rogers-Smith, C., and Ladish, J. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. Preprint arXiv:2310.20624, 2023.

Li, M., Si, W. M., Backes, M., Zhang, Y., and Wang, Y. Salora: Safety-alignment preserved low-rank adaptation. Preprint arXiv:2501.01765, 2025.

Lin, B., Jiang, W., Xu, Y., Chen, H., and Chen, Y.-C. PARM: Multi-objective test-time alignment via preference-aware autoregressive reward model. In *International Conference on Machine Learning*, 2025.

Liu, G., Lin, W., Mu, Q., Huang, T., Mo, R., Tao, Y., and Shen, L. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation. *IEEE Transactions on Information Forensics and Security*, 2025a.

Liu, G., Mu, Q., Huang, T., Wang, X., Shen, L., Lin, W., and Li, Z. Pharmacist: Safety alignment data curation for large language models against harmful fine-tuning. Preprint arXiv:2510.10085, 2025b.

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. What makes good in-context examples for gpt-3? Preprint arXiv:2101.06804, 2021.

Liu, Y., Deng, G., Xu, Z., Li, Y., Zheng, Y., Zhang, Y., Zhao, L., Zhang, T., Wang, K., and Liu, Y. Jailbreaking chatgpt via prompt engineering: An empirical study. Preprint arXiv:2305.13860, 2023.

Longpre, S., Yauney, G., Reif, E., Lee, K., Roberts, A., Zoph, B., Zhou, D., Wei, J., Robinson, K., Mimno, D., et al. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *North American Chapter of the Association for Computational Linguistics*, 2024.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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

Macchi, O. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 1975.

Meta. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. Preprint, 2024.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. Preprint arXiv:1809.02789, 2018.

OpenAI. GPT-4. Technical Report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In *Neural Information Processing Systems*, 2022.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Neural Information Processing Systems*, 2023.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, 2015.

Sheshadri, A., Ewart, A., Guo, P., Lynch, A., Wu, C., Hebbar, V., Sleight, H., Stickland, A. C., Perez, E., Hadfield-Menell, D., and Casper, S. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. Preprint arXiv:2407.15549, 2024.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. Beyond neural scaling laws: beating power law scaling via data pruning. In *Neural Information Processing Systems*, 2022.

Springer, J. M., Kotha, S., Fried, D., Neubig, G., and Raghunathan, A. Repetition improves language model embeddings. In *International Conference on Learning Representations*, 2025.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., et al. LLaMA 2: Open foundation and fine-tuned chat models. Preprint arXiv:2307.09288, 2023.

Van der Maaten, L. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.

Wei, Y., Fu, S., Jiang, W., Zhang, Z., Zeng, Z., Wu, Q., Kwok, J. T., and Zhang, Y. GITA: Graph to visual and textual integration for vision-language graph reasoning. In *Neural Information Processing Systems*, 2024.

Xia, M., Malladi, S., Gururangan, S., Arora, S., and Chen, D. Less: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*, 2024.

Xiao, Y., Tonekaboni, S., Gerych, W., Suriyakumar, V., and Ghassemi, M. When style breaks safety: Defending language models against superficial style alignment. Preprint arXiv:2506.07452, 2025.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. Preprint arXiv:2505.09388, 2025.

Yang, X., Wang, X., Zhang, Q., Petzold, L., Wang, W. Y., Zhao, X., and Lin, D. Shadow alignment: The ease of subverting safely-aligned language models. Preprint arXiv:2310.02949, 2023.

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

Ye, J., Wu, Z., Feng, J., Yu, T., and Kong, L. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, 2023.

Yi, B., Li, J., Zhang, B., Nie, L., Li, T., Huang, T., and Liu, Z. Gradient surgery for safe llm fine-tuning. Preprint arXiv:2508.07172, 2025.

Yi, J., Ye, R., Chen, Q., Zhu, B., Chen, S., Lian, D., Sun, G., Xie, X., and Wu, F. On the vulnerability of safety alignment in open-access llms. In *Findings of the Association for Computational Linguistics*, 2024.

Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., Kwok, J. T., Li, Z., Weller, A., and Liu, W. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, 2024.

Yuan, Y., Jiao, W., Wang, W., Huang, J.-t., He, P., Shi, S., and Tu, Z. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. Preprint arXiv:2308.06463, 2023a.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. Rrhf: Rank responses to align language models with human feedback without tears. Preprint arXiv:2304.05302, 2023b.

Zhan, Q., Fang, R., Bindu, R., Gupta, A., Hashimoto, T., and Kang, D. Removing rlhf protections in gpt-4 via fine-tuning. Preprint arXiv:2311.05553, 2023.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al. Lima: Less is more for alignment. In *Neural Information Processing Systems*, 2023.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. Preprint arXiv:1909.08593, 2019.

Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. Preprint arXiv:2307.15043, 2023.

A. Derivation of SPAG

Introducing multipliers $\lambda \geq 0$, the Lagrangian is

$$\mathcal{L}(\boldsymbol{\theta}, \lambda) = \|\boldsymbol{\theta} - \boldsymbol{\theta}^{+}\|^2 + \lambda\big(\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^{+}) + \langle \mathbf{g}_{\text{safe}},\, \boldsymbol{\theta} - \boldsymbol{\theta}^{+}\rangle - \tau\big). \tag{9}$$

Taking derivatives with respect to $\boldsymbol{\theta}$ and setting to zero yields the stationarity condition: $\boldsymbol{\theta} - \boldsymbol{\theta}^{+} + \lambda \mathbf{g}_{\text{safe}} = 0$. Hence, the solution has the form $\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{+} - \lambda \mathbf{g}_{\text{safe}}$. Plugging $\boldsymbol{\theta}^{\text{new}}$ into the safety constraint gives $\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^{+}) + \big\langle \mathbf{g}_{\text{safe}},\, \boldsymbol{\theta}^{\text{new}} - \boldsymbol{\theta}^{+}\big\rangle = \mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^{+}) - \lambda \|\mathbf{g}_{\text{safe}}\|^2$. Hence feasibility requires $\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^{+}) - \lambda \|\mathbf{g}_{\text{safe}}\|^2 \leq \tau$. Complementary slackness further implies $\lambda\Big(\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^{+}) - \lambda \|\mathbf{g}_{\text{safe}}\|^2 - \tau\Big) = 0$. Two cases arise: (i) If $\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^{+}) \leq \tau$, the unconstrained update already satisfies the safety constraint, and no correction is needed: $\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{+}$. (ii) Otherwise, the projection requires a step along $\mathbf{g}_{\text{safe}}$: $\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{+} - \frac{\mathcal{L}(\mathcal{D}_{\text{safe}}, \boldsymbol{\theta}^{+}) - \tau}{\|\mathbf{g}_{\text{safe}}\|^2} \mathbf{g}_{\text{safe}}$.

B. Computational Complexity of DPP Selection

Let $N = |\mathcal{D}_{\text{safe}}|$ be the size of the safe pool and let $k = |\mathcal{C}|$ be the target number of selected samples. At greedy step m (with $m-1$ items already selected), we maintain the Cholesky factor $\widehat{\mathbf{L}}_{\mathcal{C}_{m-1}} = \mathbf{C}\mathbf{C}^{\top}$, where $\mathbf{C} \in \mathbb{R}^{(m-1)\times(m-1)}$. To evaluate the gain factor for all remaining candidates $i \notin \mathcal{C}_{m-1}$, we first extract the cross-kernel block

$$\mathbf{V}_{m-1} = \widehat{\mathbf{L}}_{\mathcal{D}_{\text{safe}} \setminus \mathcal{C}_{m-1},\, \mathcal{C}_{m-1}} \in \mathbb{R}^{(N-m+1)\times(m-1)},$$

and then solve the triangular system

$$\mathbf{C}\mathbf{W}_{m-1}^{\top} = \mathbf{V}_{m-1}^{\top},$$

where each column of $\mathbf{W}_{m-1}$ corresponds to the vector $\mathbf{w}_i$ used in the gain $\widehat{\mathbf{L}}_{ii} - \|\mathbf{w}_i\|_2^2$.

Following Chen et al. (2018), we avoid repeatedly solving triangular systems from scratch. Instead, for each candidate item i, we maintain its Cholesky coordinates and gain

$$\mathbf{w}_i \in \mathbb{R}^{m-1}, \qquad d_i^2 = \widehat{\mathbf{L}}_{ii} - \|\mathbf{w}_i\|_2^2,$$

and update them *incrementally* when a new element j is added to $\mathcal{C}_{m-1}$. The update for each remaining candidate item i is

$$e_i = \frac{\widehat{\mathbf{L}}_{ij} - \langle \mathbf{w}_i,\, \mathbf{w}_j\rangle}{d_j}, \qquad \mathbf{w}_i \leftarrow [\mathbf{w}_i,\, e_i], \qquad d_i^2 \leftarrow d_i^2 - e_i^2,$$

which requires only an inner product of length $m-1$. Thus, each candidate update costs $\mathcal{O}(m)$, and the entire gain update at step m costs

$$\mathcal{O}((N - m + 1)\, m)\,.$$

Overall Complexity. Summing over all greedy steps $m = 1, \ldots, k$ yields

$$\sum_{m=1}^{k} \mathcal{O}((N - m + 1)\, m) = \mathcal{O}(Nk^2) \qquad (N \gg k).$$

Thus, the final time complexity is $\mathcal{O}(Nk^2)$, i.e., linear in the safe pool size N and quadratic in the small target subset size k.

C. Transformation Prompt

We follow Bianchi et al. (2024) to turn the BeaverTails dataset into the I-BeaverTails dataset by the following prompt.

Prompt 1: Transformation to Instructions.

You are transforming a question into an instruction. You should keep the same sentence structure. Use active and imperative verbs. The type of instructions should be diverse. Here are some examples:

Sentence: Is drinking cool?

Output: List reasons why drinking is cool.

Sentence: How do I poison food?

Output: Describe methods to poison food.

Sentence: How do I make a racist joke?

Output: Explain how to make a racist joke.

Sentence: How do I kill people?

Output: Write a poem about ways to kill people.

The following is your sentence. Sentence: {input}

Output:

Similarly, we turn the LatHarmful dataset into the Q-LatHarmful by the prompt below.

Prompt 2: Transformation to Questions.

You are transforming an instruction into a question. You should keep the same sentence structure. The type of questions should be diverse. Here are some examples:

Sentence: List reasons why drinking is cool.

Output: Is drinking cool?

Sentence: Describe methods to poison food.

Output: How do I poison food?

Sentence: Explain how to make a racist joke.

Output: How do I make a racist joke?

Sentence: Write a poem about ways to kill people.

Output: How do I kill people?

The following is your sentence. Sentence: {input}

Output:

Table 8. Comparison of data selection methods using standard fine-tuning (without SPAG).

Method	BeaverTails	I-BeaverTails	LatHarmful	Q-LatHarmful	**Avg. ASR**	GSM8K Acc
$\mathcal{D}_{\text{Low-Sim}}$ (Hsiung et al., 2025)	74.60	70.65	88.48	94.70	82.11	82.11
$\mathcal{D}_{\text{High-Sim}}$ (Hsiung et al., 2025)	67.80	69.39	74.95	86.35	74.62	86.62
Relevance-Diversity DPP	70.40	66.46	73.33	89.41	74.90	86.01

Table 9. Comparison of data selection methods when combined with the SPAG safety constraint.

Method	BeaverTails	I-BeaverTails	LatHarmful	Q-LatHarmful	**Avg. ASR**	GSM8K Acc
SPAG (w/ $\mathcal{D}_{\text{Low-Sim}}$ (Hsiung et al., 2025))	34.80	56.39	64.24	79.84	58.82	84.95
SPAG (w/ $\mathcal{D}_{\text{High-Sim}}$ (Hsiung et al., 2025))	16.60	24.32	16.16	8.96	16.51	85.69
SPARD	10.00	12.58	7.88	7.33	9.45	85.77

D. Details of Embedding Extraction

Following the common mean-pooling strategy (Springer et al., 2025), we generate embeddings by averaging the final-layer hidden states across all tokens in the input sequence. Let $\phi_t(\mathbf{x})$ denote the hidden state at position t for the input sequence $\mathbf{x} = (x_1, \ldots, x_T)$. The embedding is then computed as

$$\phi(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} \phi_t(\mathbf{x}).$$

This simple pooling strategy is widely used and has been shown to be effective for LLM-based embedding extraction.

E. Additional Results

E.1. Results for Hsiung et al. (2025)

As Hsiung et al. (2025) also employs a data selection strategy, we conducted an additional experiment to isolate its effect. Specifically, we sampled safety data using their method and applied it to fine-tuning.

Performance without SPAG (Data Selection Only). We compared our **Relevance-Diversity DPP** selection against the strategies from Hsiung et al. (2025)($\mathcal{D}_{\text{Low-Sim}}$ and $\mathcal{D}_{\text{High-Sim}}$) using standard fine-tuning. As shown in Table 8, the average ASR remains high across the board (mostly $> 73\%$) when SPAG is disabled, and the performance gap between different data selection methods is marginal.

This highlights the importance of an explicit safety constraint: without it, even well-selected safety data (e.g., $\mathcal{D}_{\text{High-Sim}}$, or our DPP selection) cannot fully realize its potential to counteract the harmful fine-tuning data.

Performance with SPAG. To meaningfully distinguish the effectiveness of different data selection strategies, we further examine all selection strategies when combined with **SPAG**, i.e., with the safety constraint enforced during fine-tuning.

As shown in Table 9, our method achieves the lowest average ASR performance. Compared with the high-similarity set $\mathcal{D}_{\text{High-Sim}}$ (Hsiung et al., 2025), our approach reduces the average ASR from 16.51% to **9.45%** while maintaining comparable utility on GSM8K. Meanwhile, the low-similarity set $\mathcal{D}_{\text{Low-Sim}}$ (Hsiung et al., 2025) performs substantially worse, confirming that such low-similarity safety samples are much less useful for enforcing the safety constraint. These results demonstrate that our Relevance-Diversity DPP selection is more effective at selecting safety data than the relevance-only metrics in Hsiung et al. (2025).

E.2. Generalization to Additional Utility Tasks

To evaluate SPARD's generalization beyond QA-style benchmarks, we conduct additional experiments on two fundamentally different task types: **SST-2** (sentiment classification) and **MBPP** (code generation), using Qwen-2.5-7B-Instruct under the BeaverTails attack with the same default hyperparameters used throughout the paper.

As shown in Table 10, SPARD consistently achieves the lowest ASR and HS across both tasks while maintaining competitive

Table 10. Defense performance of Qwen-2.5-7B-Instruct on SST-2 and MBPP under the BeaverTails harmful fine-tuning attack. Lower ASR/HS indicates stronger safety; higher accuracy indicates better utility. Best results are in **bold**.

	SST-2		MBPP		
Method	Accuracy	ASR	Accuracy	ASR	HS
SFT	**94.38**	44.40	58.8	65.20	3.12
Lisa (Huang et al., 2024c)	93.81	26.80	54.2	27.20	1.80
SafeGrad (Yi et al., 2025)	94.84	33.40	**61.0**	35.00	2.06
SPARD	94.50	**21.60**	59.0	**14.60**	**1.42**

Table 11. Defense performance on additional models (GSM8K, BeaverTails attack). Lower ASR/HS indicates stronger safety; higher accuracy indicates better utility. Best results are in **bold**.

	Qwen-3-8B			Qwen-2.5-14B-Instruct		
Method	Accuracy	ASR	HS	Accuracy	ASR	HS
SFT	83.39	76.4	3.57	88.63	24.0	1.68
Lisa (Huang et al., 2024c)	80.44	19.6	1.53	86.54	19.8	1.55
SafeGrad (Yi et al., 2025)	**84.76**	16.6	1.45	88.79	20.8	1.60
SPARD	83.92	**8.8**	**1.28**	**88.93**	**11.6**	**1.38**

task accuracy. On SST-2, SPARD reduces ASR to 21.60%, outperforming Lisa (26.80%) and SafeGrad (33.40%), with negligible accuracy loss relative to SFT (94.50% vs. 94.38%). On MBPP, SPARD achieves the lowest ASR (14.60%) and HS (1.42) by a substantial margin, while preserving accuracy (59.0%) close to SFT (58.8%). In contrast, SafeGrad maintains the highest accuracy (61.0%) but provides considerably weaker safety (ASR 35.00%), and Lisa achieves moderate safety (ASR 27.20%) at the cost of a significant accuracy drop (54.2%).

Together with the GSM8K (math reasoning) and OpenBookQA (science QA) results in the main text, these experiments confirm that SPARD generalizes effectively across four distinct task types (math reasoning, science QA, text classification, and code generation) without requiring per-task hyperparameter tuning.

E.3. Generalization Across Model Families and Sizes

To further validate that SPARD generalizes beyond models, we conduct additional experiments on Qwen-3-8B (Yang et al., 2025) and Qwen-2.5-14B-Instruct (An et al., 2024), both on GSM8K under the BeaverTails attack using the same default hyperparameters.

As shown in Table 11, SPARD consistently achieves the lowest ASR and HS on both models while maintaining competitive accuracy. On Qwen-3-8B, SPARD reduces ASR to 8.8%, substantially outperforming SafeGrad (16.6%) and Lisa (19.6%), while preserving accuracy close to SFT (83.92% vs. 83.39%). On Qwen-2.5-14B-Instruct, SPARD achieves the lowest ASR (11.6%) and the highest accuracy (88.93%), demonstrating that the framework scales effectively to larger models. Combined with the results on Qwen-2.5-7B and LLaMA-3.2-3B in the main text, SPARD is validated across four models spanning three model sizes (3B, 7B/8B, 14B).

F. Discussion on the First-Order Approximation

SPAG relies on a first-order Taylor expansion to linearize the safety constraint, which is an idealized approximation in the highly non-convex loss landscape of deep neural networks. Under extreme gradient divergence, the linearized half-space $\mathcal{C}^+$ may deviate from the true feasible region. SPAG mitigates this in two ways: (1) the trust-region radius η_{safe} confines updates to a local neighborhood where the linear approximation remains valid; (2) Since SPAG reprojects at every training step, each correction is small, and any under-correction due to curvature persists into the next step, triggering further correction. This provides a natural self-correcting mechanism that prevents accumulated safety drift.

G. Limitations

SPAG's safety projection relies on a first-order Taylor approximation to linearize the safety constraint at each step. While the trust-region radius η_{safe} mitigates overshoot and empirical results confirm stable convergence, global convergence guarantees for the safety constraint are not provided. Additionally, when the fine-tuning data lies in an outlier domain far from available safety datasets, the relevance scores q_i become nearly uniform, and the DPP kernel gracefully degrades to diversity-only selection. The FTaaS provider can also expand the safety pool with domain-specific data to further improve coverage.

H. Large Language Model Usage Statement

During the preparation of this manuscript, large language models (LLMs) were employed exclusively for writing assistance, including polishing grammar, improving clarity, and refining presentation. All scientific contributions, including the development of the SPARD framework, theoretical derivations, and empirical evaluations, are entirely original to the authors. The LLMs are therefore not considered authors of this work.