ShieldBench: A Comprehensive Benchmark for Evaluating the Persistence of LLM Safety Interventions

Mert Ogul[1], **Rishitha Voleti**[2], **Shanduojiao Jiang**[3]
[1]Eindhoven Technical University
[2]George Mason University
[3]Stanford University
m.ogul@student.tue.nl, rvoleti2@gmu.edu, sj99@stanford.edu

Abstract

Large Language Models (LLMs) are increasingly relied upon for information access and decision support, yet they continue to struggle with distinguishing between benign and harmful prompts. Existing evaluation protocols fall short: some rely on unrealistic assumptions, while others provide only partial assessments of model safety. We introduce ShieldBench, a benchmark designed to evaluate not only immediate safety compliance but also the persistence of safety interventions under realistic usage conditions. Our benchmark incorporates a suite of recent weight-space editing techniques (Task-Vector Negation, Diverse Inversion, Guided Distortion, AlphaEdit, SafetyLora, and TaLoS Sparsity) applied across multiple open-source models and diverse safety datasets like HarmBench. By evaluating performance under both greedy and sampling-based decoding, we capture conditions closer to real world deployments. Our results reveal persistence depends critically on weight-space geometry, providing actionable insights for building durable LLM safety.

Introduction

Large Language Models (LLMs) have rapidly transformed how people access information, interact with technology, and make decisions. Their ability to generate coherent and contextually relevant responses at scale has made them indispensable across domains from education and healthcare to software engineering and creative writing. However, this ubiquity comes with a critical challenge: LLMs struggle to reliably differentiate between benign prompts and those crafted to elicit harmful behaviors (Zou et al. 2023; Liu et al. 2023; Chao et al. 2024).

This inability to separate "good" from "bad" prompts has led to several jailbreaks like the "grandma exploit". Despite safety training, many models still comply, outputting detailed harmful instructions under the guise of roleplay (Zou et al. 2023; Liu et al. 2023). These issues underscore the urgent need to develop methods for training, evaluating, and reinforcing LLMs so that they can robustly identify and resist unsafe prompts. To test the LLMs and maintain some type of safety there have been various safety benchmarks created. However with the modern day benchmarks there are several limitations.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Single-point failure dependencies: Many evaluations rely on AI judges to determine response safety. When these judges are inaccurate or biased, they create bottlenecks for evaluation quality (Beyer et al. 2025; Shen et al. 2025).

Unrealistic testing conditions: Current benchmarks often evaluate models only under artificial scenarios, such as purely greedy decoding, which do not reflect actual deployment conditions (Huang et al. 2023).

Incomplete safety assessment: Existing benchmarks like HarmBench test immediate compliance but fail to measure whether safety edits remain effective when models undergo additional training—a critical gap for real-world deployment (Mazeika et al. 2024; Qi et al. 2023).

Limited scope and diversity: Many benchmarks rely on narrow datasets that fail to capture the full diversity of harmful prompts or the subtle variations that can bypass safety measures (Mazeika et al. 2024; Gehman et al. 2020).

This leads us to raise a couple of questions on what our work seeks.

1. Do different intervention techniques vary in their persistence across models and datasets?
2. How does model architecture and weight-space geometry affect the durability of safety edits?
3. Can a benchmark framework systematically expose these persistence patterns in ways existing evaluations cannot?

Our Contribution: ShieldBench

To address these limitations, we introduce ShieldBench, the first benchmark explicitly designed to evaluate the *persistence* of LLM safety interventions. Unlike prior efforts that focus on one-shot safety compliance, ShieldBench emphasizes durability: whether safety gains endure across multiple models, techniques, and adversarial scenarios.

Our contributions are threefold:

1. **Comprehensive methodology**: We implement and systematically compare six recent weight-space editing techniques within a unified evaluation pipeline, applied to five widely-used open-source models.
2. **Realistic evaluation conditions**: We test under both greedy and sampling-based decoding across three complementary safety datasets (HarmBench, HEX-Phi, WMDP) that capture different aspects of harmful behavior (Mazeika et al. 2024; Qi et al. 2023; Li et al. 2024).

3. **Persistence-focused design**: We subject edited models to adversarial fine-tuning at varying intensities to measure whether safety improvements survive real-world pressures (Qi et al. 2023; Huang et al. 2023).

Our results reveal that no single intervention universally dominates. Instead, persistence depends critically on the alignment between intervention type, model architecture, and risk domain—insights that point toward adaptive safety approaches for robust LLM deployment.

Methodology

Our evaluation framework consists of three core components: intervention techniques, evaluation datasets, and experimental protocol. This section details each component and explains our design rationales.

Weight-Space Safety Interventions

We selected six recent intervention techniques that represent different approaches to modifying model behavior through parameter editing. These methods span the major paradigms in the field and offer complementary strengths for different types of models and risks (Ilharco et al. 2022; Pham et al. 2024; Liu et al. 2024; Fang et al. 2024; Hsu et al. 2024; Xue and Mirzasoleiman 2025; Iurada, Ciccone, and Tommasi 2025).

Overview of Intervention Categories Our selected interventions fall into three broad categories:

- **Vector-based methods** (Task Vectors, AlphaEdit):Treat safety as directions in parameter space. They are cheaper and decent at baseline for persistence because you can quickly tune the edit strength, revert it, and see whether a simple linear shift still holds after adversarial fine-tuning (Ilharco et al. 2022; Fang et al. 2024).
- **Localization methods** (TaLoS, Guided Distortion): Target specific model components or representations. These methods focus on only touching the parts of the model that is tied with unsafe behaviors and keep helpful responses intact which makes the model better after being fine-tuned (Iurada, Ciccone, and Tommasi 2025; Liu et al. 2024).
- **Adaptation methods** (Safety LoRA, Diverse Inversion): Use modular updates or representation manipulation. They approaches are flexible and lightweight, allowing to quickly plug in the safety patch and see how they hold across different kinds of prompts (Hsu et al. 2024; Xue and Mirzasoleiman 2025; Pham et al. 2024).

Method	Category	Key Advantage	Computational Cost
Task Vectors	Vector-based	Simple, interpretable	Low
AlphaEdit	Vector-based	Controllable strength	Low
TaLoS	Localization	Sparse, targeted	Medium
Guided Distortion	Localization	Causally targeted	Medium
Safety LoRA	Adaptation	Modular, reversible	Low
Diverse Inversion	Adaptation	Robust to distribution shift	High

Table 1: Overview of intervention techniques and their characteristics.

Detailed Method Descriptions

Task Vectors: Task vectors treat fine-tuning as a parameter delta: subtracting the base model's weights from the fine-tuned model yields a vector in parameter space. This vector can then be added, scaled, or negated to adjust the model's behavior (Ilharco et al. 2022). $\mathrm{v}_{task} = \theta_{finetuned} - \theta_{base}, \quad \theta' = \theta_{base} - v_{harm}$

TaLoS: (Task-Localized Sparse Fine-Tuning): TaLoS extends task vectors by applying the update only to sparse, localized regions of the network. Masks are learned to restrict edits to attention and MLP blocks tied to the unsafe concept (Iurada, Ciccone, and Tommasi 2025). $\Delta\theta_{TaLoS} = M \odot \left(\theta_{finetuned} - \theta_{base}\right)$

Safety LoRA: LoRA adds small low-rank adapter matrices into transformer layers. In the safety version, these adapters are trained on safe responses so that, at inference, activations get nudged toward refusals when needed (Hu et al. 2021; Hsu et al. 2024; Xue and Mirzasoleiman 2025).

$$\Delta W = AB^{\top}, \quad A \in R^{d\times r},\, B \in R^{d\times r},\, r \ll d$$

Diverse Inversion: Inversion methods flip harmful representations in the opposite direction. Diverse Inversion strengthens this idea by collecting multiple such "harmful directions" and averaging them, so the patch doesn't overfit to just one type of unsafe example (Pham et al. 2024).

$$v_{inv} = \frac{1}{k}\sum_{i=1}^{k} -v_i, \quad h' = h + v_{inv}$$

Guided Distortion: Guided distortion directly edits internal features. It identifies where harmful activations show up in the model and then shifts them closer to benign ones, keeping the rest of the network intact (Liu et al. 2024).

$$h' = h - \tau_{bad} + \tau_{benign}$$

AlphaEdit: AlphaEdit is another low-rank approach, but with a twist: it lets you scale the edit with a continuous knob α. That way, you can dial in just the right amount of adjustment without pushing the model too far (Fang et al. 2024).

$$\theta' = \theta + \alpha \cdot \Delta\theta_{low-rank}$$

Evaluation Datasets

A limitation of previous benchmarks has been their reliance on narrow, homogeneous datasets that fail to capture the full range of harmful and benign inputs). We rely on three existing but complementary safety benchmarks: HarmBench, HEX-Phi, and WMDP (Mazeika et al. 2024; Qi et al. 2023; Li et al. 2024). We chose this mix for two reasons. First, many prior evaluations only use a single dataset, which limits coverage; bringing these three together gives us broad, adversarial, and domain-specific perspectives in one place. Second, WMDP targets hazardous procedural knowledge (mostly multi-choice) and is used to evaluate removal of dangerous know-how, letting us test whether interventions can reliably separate closely related cases.

Experimental Evaluation Pipeline

Our experiments followed a consistent pipeline. We began with a **baseline assessment**, where each model was evaluated on all datasets and decoding configurations to establish initial harmfulness (H_0) and helpfulness (B_0). We then performed **method calibration** by fitting each intervention on 1,000 harmful and 1,000 benign samples. During calibration, we swept scale values $S = \{0.5, 1.0, 2.0, 3.0, 5.0\}$, evaluating with greedy decoding and selecting the scale that minimized harmfulness while maintaining $B \geq 0.5$. With this optimal scale, we applied the method to produce a patched model and re-ran the baseline evaluation to obtain post-intervention metrics (H_1, B_1).

To measure persistence, we conducted **stress tests**. These included benign supervised fine-tuning and adversarial fine-tuning on jailbreak-style prompts at increasing intensity levels, after which we re-evaluated metrics (H_t, B_t). Finally, each method reported **locality statistics** (sparsity, affected layers, parameter counts), and we aggregated all results into JSON reports with averages, rankings, and LaTeX tables.

Results and Analysis

HarmBench Results

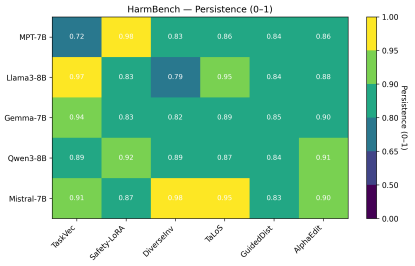

Figure 1: Absolute persistence on HarmBench

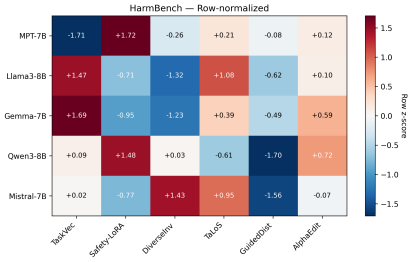

Figure 2: Row-normalized persistence on HarmBench

These results display that HarmBench is built around overtly harmful instructions (Mazeika et al. 2024), which makes it one of the easier datasets for safety interventions to handle. That's why the absolute persistence scores are consistently high, ranging from 0.72 up to 0.98. Once a model has been trained to refuse these obvious prompts, those refusals tend to hold even under adversarial fine-tuning. For instance, Mpt-7B-Instruct with Safety LoRA reaches 0.98 (MosaicML 2023; Hsu et al. 2024; Xue and Mirzasoleiman 2025), while Llama-3.1-8B-Instruct with TaLoS achieves 0.95 (Meta AI 2024; Iurada, Ciccone, and Tommasi 2025). Even weaker pairings, like Mpt-7B-Instruct with Task Vectors (0.72) (MosaicML 2023; Ilharco et al. 2022) or Llama-3.1-8B-Instruct with Diverse Inversion (0.79) (Meta AI 2024; Pham et al. 2024), still remain above 0.7. On the surface, this suggests interventions are reliably effective against clear harms.

But the row-normalized heatmap reveals a more nuanced story. For each model, some interventions stand out while others trail behind. Mpt-7B-Instruct benefits disproportionately from Safety LoRA (+1.72 z), while Task Vectors underperform (–1.71 z) despite both having "good" absolute scores. Llama-3.1-8B-Instruct shows the reverse dynamic: TaLoS is its best fit (+1.08 z), while Diverse Inversion drags it down (–1.32 z). For Gemma-7B-It , Task Vectors align unusually well (+1.69 z), while Qwen3-8B favors Safety LoRA and AlphaEdit (+1.48, +0.72 z) but does poorly with Guided Distortion (–1.70 z) (Google 2024; Qwen Team 2025; Hsu et al. 2024; Fang et al. 2024; Liu et al. 2024).

This matters because HarmBench, as a dataset, only tests whether models can suppress very direct unsafe outputs (Mazeika et al. 2024). Since these behaviors can often be shifted along relatively simple "refusal" directions, many methods look successful in absolute terms (Arditi et al. 2024). However, the differences across techniques point to how each model encodes "unsafe" directions in weight space. For Mpt-7B-Instruct, harmful behaviors seem to lie in a low-rank subspace that Safety LoRA can effectively target, while for Llama-3.1-8B-Instruct, sparsity-localized updates from TaLoS align better with its modularized architecture (Hsu et al. 2024; Xue and Mirzasoleiman 2025; Iurada, Ciccone, and Tommasi 2025). Gemma-7B-It , by contrast, responds more to linear subtraction via Task Vectors, suggesting its harmful/benign circuits are encoded in a flatter direction where negation is effective (Ilharco et al. 2022).

So, even in a "best case" dataset where harmfulness is straightforward and persistence looks uniformly high, HarmBench teaches us an important lesson: the durability of safety edits is still shaped by the interaction between method and model geometry. HarmBench is therefore useful for detecting broad improvements, but the variation across models reminds us that strong absolute numbers do not equal universal robustness.

HEX-Phi Results

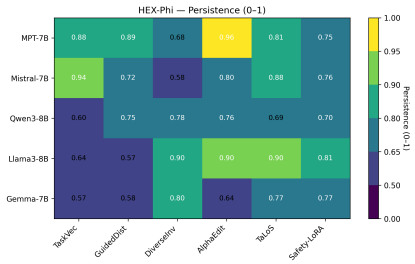

Figure 3: Absolute persistence on HEX-Phi

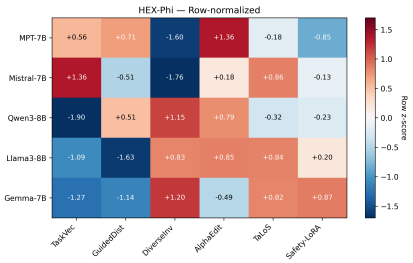

Figure 4: Row-normalized persistence on HEX-Phi

HEX-Phi is designed to test whether safety persists under rephrasing and boundary-pushing prompts (Qi et al. 2023), which makes it more challenging than HarmBench. The absolute persistence map shows more spread (0.57–0.96), with strong outliers in both directions. Mpt-7B-Instruct with AlphaEdit achieves 0.96, the highest score overall (MosaicML 2023; Fang et al. 2024), suggesting that fine-grained, tunable low-rank edits are particularly effective for stabilizing MPT against paraphrased jailbreaks. At the other extreme, Gemma-7B-It with Task Vectors or Guided Distortion drops to 0.57–0.58 (Google 2024; Ilharco et al. 2022; Liu et al. 2024), showing that linear subtraction or direct feature distortion does little to shift Gemma's refusal boundaries when prompts are phrased adversarially.

The row-normalized view makes these contrasts sharper. For Mistral-7B-Instruct-v0.3, Task Vectors unexpectedly emerge as the strongest option (+1.36 z, 0.94 absolute), while Guided Distortion and Diverse Inversion collapse (–0.51, –1.76 z) (Mistral AI 2024; Liu et al. 2024; Pham et al. 2024), reflecting that Mistral's decision surface can be shifted with simple deltas but resists more complex edits. Llama-3.1-8B-Instruct shows the opposite pattern: vector-style edits fail (–1.09 to –1.63 z), while representation-focused methods like Diverse Inversion, AlphaEdit, and TaLoS all perform well (+0.83 to +0.85 z) (Meta AI 2024; Pham et al. 2024; Fang et al. 2024; Iurada, Ciccone, and Tommasi 2025). For Qwen3-8B, Diverse Inversion is the standout (+1.15 z) (Qwen Team 2025; Pham et al. 2024), again pointing to representation manipulation as a better fit.

This matters because HEX-Phi probes where in representation space the refusal boundary lives (Qi et al. 2023). On some models (like Mistral), harmful and benign regions are linearly separable, making Task Vectors effective (Ilharco et al. 2022). On others (like Llama3 or Qwen3), safety requires reshaping internal subspaces through localized or inversion-based edits (Iurada, Ciccone, and Tommasi 2025; Pham et al. 2024). In short, HEX-Phi reveals that persistence is not just about whether safety holds, but how the refusal boundary is encoded—and that different architectures demand different strategies to stabilize it.

WMDP Results

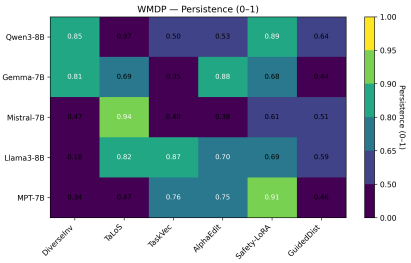

Figure 5: Absolute persistence on WMDP

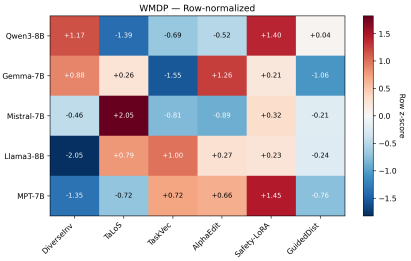

Figure 6: Row-normalized persistence on WMDP

The nature of WMDP prompts is what makes this dataset so punishing (Li et al. 2024). Unlike HarmBench, which tests refusals against obvious "red-flag" instructions, or HEX-Phi, which stresses boundary phrasing, WMDP pushes models with procedural, step-by-step knowledge about sensitive topics. These are not one-shot instructions a model can easily reject; they require suppressing deeper reasoning pathways. That design explains both the lower absolute persistence scores (as low as 0.18) and the sharper contrasts between techniques.

When the harmfulness is encoded in procedural reasoning chains, interventions that simply flip or subtract directions in weight space (like Task Vectors or Diverse Inversion on some models) often fall apart (Ilharco et al. 2022; Pham et al. 2024). For example, Llama-3.1-8B-Instruct with Diverse Inversion collapses to 0.18 (Meta AI 2024; Pham et al. 2024) because flipping harmful representations does not prevent the model from reconstructing unsafe reasoning paths step by step. Similarly, Gemma struggles when edits are too coarse, with Task Vectors leaving it at only 0.35 (Google 2024; Ilharco et al. 2022).

By contrast, methods that localize or modularize updates, such as Mistral-7B-Instruct-v0.3 with TaLoS (0.94) or Mpt-7B-Instruct with Safety LoRA (0.91), perform far better (Mistral AI 2024; Iurada, Ciccone, and Tommasi 2025; MosaicML 2023; Hsu et al. 2024). These approaches succeed because they do not simply blur out unsafe knowledge; they carve stable refusal subspaces that remain intact even when the model is pressured to reason procedurally (Arditi et al. 2024). AlphaEdit shows a similar advantage for Gemma (0.88) (Fang et al. 2024; Google 2024), where its tunable,

low-rank updates make the safety edit strong enough to disrupt multi-step reasoning without harming benign performance.

In short, WMDP highlights that the dataset's prompt style, which focuses on multi-step and operationally sensitive instructions, forces models to rely on their deeper reasoning circuits (Li et al. 2024). Success therefore depends not on blunt negation but on interventions that reshape or stabilize the actual representational geometry of those circuits (Arditi et al. 2024). The methods that persist here are the ones that can lock down reasoning pathways rather than surface responses.

Dataset Stress Gradients

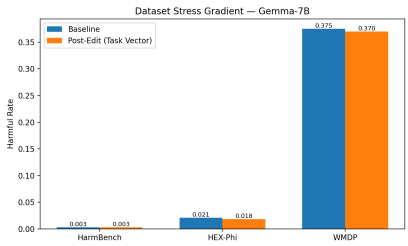

Figure 7: Stress gradient — Gemma-7B-It

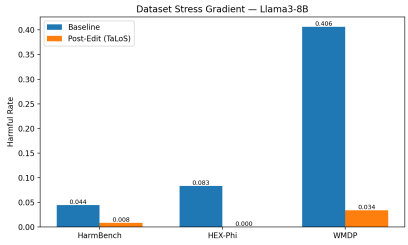

Figure 8: Stress gradient — Llama-3.1-8B-Instruct

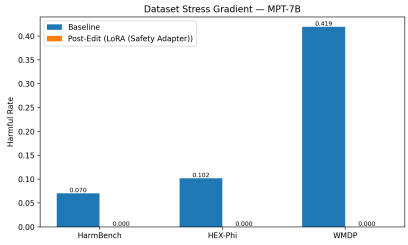

Figure 9: Stress gradient — Mpt-7B-Instruct

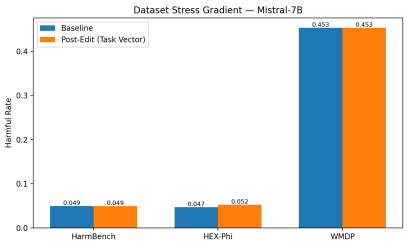

Figure 10: Stress gradient — Mistral-7B-Instruct-v0.3

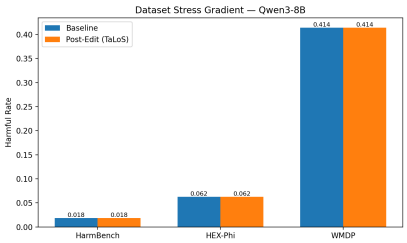

Figure 11: Stress gradient — Qwen3-8B

Stress gradients capture how harmfulness changes before and after an intervention, measured across increasing levels of adversarial fine-tuning. While persistence heatmaps give a single summary score, stress gradients show the trajectory of safety erosion. They reveal whether a safety edit degrades gradually as pressure increases or collapses immediately even under light adversarial training (Qi et al. 2023). In this way, stress gradients provide a more dynamic picture of intervention durability, complementing the static snapshot given by persistence scores.

The stress gradients highlight that model and method alignment is just as critical as dataset difficulty. For example, Llama-3.1-8B-Instruct responds dramatically to TaLoS, with WMDP harmfulness dropping from more than 40 percent to only 3.4 percent (Meta AI 2024; Iurada, Ciccone, and Tommasi 2025). This outcome suggests that TaLoS's sparse and localized edits align well with Llama3's modular architecture, carving out targeted refusal subspaces that survive even when the model is pressured to perform multi-step reasoning. Mpt-7B-Instruct shows a different but equally important pattern: when paired with Safety LoRA, harmfulness collapses to zero across all three benchmarks (MosaicML 2023; Hsu et al. 2024). This indicates that for MPT, unsafe behavior is concentrated in low-rank subspaces that LoRA adapters can effectively block, making the safety patch modular, efficient, and durable (Hu et al. 2021; Arditi et al. 2024).

By contrast, Task Vectors on Mistral-7B-Instruct-v0.3 or Gemma-7B-It barely shift outcomes on WMDP, remaining around 40 percent harmfulness (Mistral AI 2024; Google 2024; Ilharco et al. 2022). This reveals that simple subtraction is ineffective when unsafe reasoning is distributed across deeper circuits. These failures are important because they demonstrate that not all unsafe behaviors can be neutralized with blunt global edits, and that models differ in

how and where harmful knowledge is encoded (Arditi et al. 2024). For this research, that means persistence cannot be treated as a single score. Instead, it depends on whether the method is capable of reshaping the right parts of the model's internal geometry.

This distinction is central to ShieldBench's contribution. By combining persistence heatmaps with stress gradients, we are able to show not only which interventions survive adversarial fine-tuning but also whether those interventions translate into meaningful safety gains. The fact that some model and method combinations achieve near-complete elimination of harmfulness while others fail outright illustrates both the promise and the limits of weight-space safety edits. This finding demonstrates that durable safety is possible, but only if we carefully understand the interaction between dataset difficulty, model architecture, and the representational geometry that each intervention targets.

Dataset-Specific Stress Testing

Our evaluation across different risk domains reveals that intervention effectiveness varies dramatically by dataset:

HarmBench: Broad but Overt Risks Most interventions succeed on HarmBench because prompts are overtly harmful and linearly separable (Mazeika et al. 2024; Arditi et al. 2024). Representative results:

Model + Method	Before	After	Δ
Llama-3.1-8B-Instruct + TaLoS	4.4%	0.8%	–3.6%
Mpt-7B-Instruct + Safety LoRA	3.1%	0.0%	–3.1%

Table 2: Persistence results on HarmBench

Even after heavy adversarial fine-tuning, most combinations maintain significant safety improvements, indicating that overt harmful behaviors are relatively easy to suppress persistently (Mazeika et al. 2024).

HEX-Phi: Adversarial Phrasing and Edge Cases HEX-Phi exposes brittleness in interventions by testing adversarial phrasing and policy ambiguities (Qi et al. 2023). Only methods that reshape internal decision boundaries succeed (Iurada, Ciccone, and Tommasi 2025; Pham et al. 2024; Fang et al. 2024):

Model + Method	Before	After	Δ
Llama-3.1-8B-Instruct + TaLoS	8.3%	0.0%	–8.3%
Gemma-7B-It + Task Vectors	9.1%	8.7%	–0.4%

Table 3: Persistence results on HEX-Phi

The stark difference between these results highlights how some models encode harmful behaviors in more manipulable representations than others (Arditi et al. 2024).

WMDP: Procedural and Operationally Sensitive Knowledge WMDP proves most challenging, as it probes multi-step procedural reasoning where simple refusal strategies fail:

Model + Method	Before	After	Δ
Mistral-7B-Instruct-v0.3 + Task Vectors	40.2%	40.0%	–0.2%
Qwen3-8B + TaLoS	37.4%	37.4%	0.0%
Mpt-7B-Instruct + Safety LoRA	3.1%	0.0%	–3.1%
Llama-3.1-8B-Instruct + TaLoS	4.4%	3.4%	–1.0%

Table 4: Persistence results on WMDP

Only when models encode procedural knowledge in modular, accessible circuits do interventions succeed on WMDP (Arditi et al. 2024).

Understanding Weight-Space Geometry

Our results suggest that intervention persistence depends critically on how models encode harmful behaviors in weight space:

Modular Encoding (MPT, Llama3): Unsafe behaviors appear to lie along separable, low-dimensional directions that can be efficiently patched with sparse or vector-based methods. These edits persist because adversarial fine-tuning cannot easily re-entangle the representations.

Entangled Encoding (Gemma, Mistral): Harmful and benign behaviors are more intertwined, making localized updates unstable. When safety edits are applied, adversarial fine-tuning quickly re-entangles representations, undoing the patch.

Adaptive Requirements: This geometric perspective suggests that effective safety interventions must be matched to model architecture–sparse methods for modular models, representation-level methods for entangled models.

Discussion and Implications

The dataset-level results highlight an important insight: durable safety is not just about applying an edit, but about aligning the intervention with both the model's weight-space geometry and the type of risk being evaluated (Arditi et al. 2024). For datasets like HarmBench, where harmfulness is broad and overt, most interventions succeed because the unsafe behaviors are comparatively linearly separable and edits can enforce surface-level refusal strategies (Mazeika et al. 2024; Arditi et al. 2024). However, for HEx-PHI, which probes adversarial phrasing and boundary conditions, only methods that reshape internal representations (e.g., TaLoS, Diverse Inversion, Guided Distortion) yield robust improvements (Iurada, Ciccone, and Tommasi 2025; Pham et al. 2024; Liu et al. 2024). The hardest case is WMDP, which stresses procedural and operationally sensitive knowledge; here, refusal-based or linear subtraction methods collapse, as they fail to touch the circuits responsible for multi-step unsafe reasoning (Li et al. 2024). Only when models already encode such circuits in modular form—as in Mpt-7B-Instruct and Llama-3.1-8B-Instruct—do sparsity- or LoRA-based interventions persist (MosaicML 2023; Meta AI 2024; Iurada, Ciccone, and Tommasi 2025; Hsu et al. 2024; Xue and Mirzasoleiman 2025).

This perspective reframes persistence as a function of weight-space geometry. If unsafe behaviors lie along flat, low-dimensional, or well-localized directions, they can be

efficiently patched with vector-based or sparse methods (Ilharco et al. 2022; Iurada, Ciccone, and Tommasi 2025). If they are sharp, entangled, or distributed, edits are either erased by adversarial fine-tuning or over-generalize, harming benign performance (Qi et al. 2023). Thus, persistence depends as much on the architecture and inductive biases of the base model as it does on the choice of intervention (Arditi et al. 2024).

From a practical standpoint, these findings suggest that safety evaluation must go beyond one-shot compliance metrics. A benchmark like ShieldBench is necessary to reveal failure modes that remain hidden in standard tests: strong gains on HarmBench may signal only superficial style edits, while persistence under WMDP or HEx-PHI provides a truer measure of whether the underlying unsafe circuits have been disrupted (Mazeika et al. 2024; Qi et al. 2023; Li et al. 2024). Looking forward, robust safety will likely require *hybrid* approaches that combine linear negation, sparse localization, and representation inversion—tuned to both the model family and the risk domain (Ilharco et al. 2022; Iurada, Ciccone, and Tommasi 2025; Pham et al. 2024; Liu et al. 2024).

Conclusion

In this work, we introduced ShieldBench, a benchmark designed to evaluate not just the immediate effectiveness of safety interventions in large language models, but their persistence under adversarial pressure. By re-implementing a suite of recent weight–space editing techniques—including Task Vectors, TaLoS, Safety LoRA, Diverse Inversion, Guided Distortion, and AlphaEdit—within a common API, and applying them across multiple open-source models and heterogeneous safety datasets, we provided a systematic framework for testing whether safety gains endure.

Our results highlight several important takeaways. First, persistence is not a universal property of an intervention, but an interaction between method, model architecture, and dataset. While Mpt-7B-Instruct retained strong safety under Safety LoRA, Gemma-7B-It collapsed quickly, showing that weight-space geometry governs whether edits can survive retraining. Second, dataset diversity is essential: methods that appear effective on HarmBench often fail on HEX-Phi or WMDP, revealing brittleness that single-dataset benchmarks would miss. Third, the success of sparsity-localized (TaLoS) and vector-negation methods (Task Vectors) demonstrates that complementary approaches—targeting both separable unsafe subspaces and localized circuits—may be needed for durable safety.

The broader impact of this work is twofold. Practically, ShieldBench provides the community with a reproducible toolkit for stress-testing LLM safety patches, complete with configs, logs, and reusable deltas. This enables fairer comparison across interventions and models, and lowers the barrier for deploying safety patches in modular, lightweight ways. Conceptually, ShieldBench reframes LLM safety as not only a matter of what edits achieve in one-shot evaluation, but whether those edits persist when models are retrained, adapted, or attacked.

We see ShieldBench as a first step toward persistence-aware safety evaluation. Future work may expand to larger model families, richer adversarial datasets, and hybrid editing strategies that combine low-rank, sparse, and inversion-based methods. Ultimately, robust and trustworthy language models will require both technical advances in persistent safety edits and benchmarks like ShieldBench that expose where those edits fail.

Limitations

Model Scale: We restricted our evaluation to 7B-8B sized open-source models such as Qwen3-8B, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, Mpt-7B-Instruct, and Gemma-7B-It . It means our findings may not directly transfer to frontier-scale systems. Larger models may encode harmful behaviors differently, and persistence dynamics could shift as weight-space geometry becomes higher-dimensional.

Limited attack sophistication: Our adversarial fine-tuning represents relatively simple attacks. More sophisticated adversaries might employ gradient-based attacks, data poisoning, or other advanced techniques.

Dataset coverage: Although we incorporated three complementary datasets, our coverage of safety domains remains incomplete. For example, we do not include toxicity-focused corpora (e.g., RealToxicityPrompts) or multimodal datasets. In particular, WMDP does not provide paired benign/harmful prompts, which limits the granularity of persistence scoring. Broader dataset inclusion will be necessary for generalizing our findings.

Adversarial fine-tuning setup: We simulated adversarial retraining using LoRA adapters at three intensity levels. While this design captures meaningful stress gradients, it does not exhaust the space of adaptive attacks. Different attack strategies might undo safety edits in ways we did not capture.

Computational constraints: Our experiments were conducted on modest GPU resources, which constrained both calibration set size and number of adversarial runs. As a result, our evaluation is necessarily a sample rather than an exhaustive stress test. This may partly explain why some interventions, such as Guided Distortion, show higher variance across models.

Open-source scope: Finally, our benchmark was applied exclusively to open-source models. This choice was deliberate, for transparency and reproducibility, but it limits direct comparison to proprietary models that dominate deployment. Whether ShieldBench findings generalize to such systems remains an open question.

References

Arditi, A.; Obeso, O.; Syed, A.; Paleka, D.; Panickssery, N.; Gurnee, W.; and Nanda, N. 2024. Refusal in Language Models Is Mediated by a Single Direction. *arXiv preprint arXiv:2406.11717*.

Beyer, T.; Xhonneux, S.; Geisler, S.; Gidel, G.; Schwinn, L.; and Günnemann, S. 2025. LLM-Safety Evaluations Lack Robustness. arXiv:2503.02574.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2024. Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv:2310.08419.

Fang, J.; Jiang, H.; Wang, K.; Ma, Y.; Shi, J.; Wang, X.; He, X.; and Chua, T.-S. 2024. AlphaEdit: Null-Space Constrained Knowledge Editing for Language Models. *arXiv preprint arXiv:2410.02355*.

Gehman, S.; Gururangan, S.; Sap, M.; Choi, Y.; and Smith, N. A. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. *arXiv preprint arXiv:2009.11462*.

Google. 2024. google/gemma-7b-it. Hugging Face model card.

Hsu, C.-Y.; Tsai, Y.-L.; Lin, C.-H.; Chen, P.-Y.; Yu, C.-M.; and Huang, C.-Y. 2024. Safe LoRA: the Silver Lining of Reducing Safety Risks when Fine-tuning Large Language Models. *arXiv preprint arXiv:2405.16833*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

Huang, Y.; Gupta, S.; Xia, M.; Li, K.; and Chen, D. 2023. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. *arXiv preprint arXiv:2310.06987*.

Ilharco, G.; Ribeiro, M. T.; Wortsman, M.; Gururangan, S.; Schmidt, L.; Hajishirzi, H.; and Farhadi, A. 2022. Editing Models with Task Arithmetic. *arXiv preprint arXiv:2212.04089*.

Iurada, L.; Ciccone, M.; and Tommasi, T. 2025. Efficient Model Editing with Task-Localized Sparse Fine-tuning. *arXiv preprint arXiv:2504.02620*.

Li, N.; Pan, A.; Gopal, A.; et al. 2024. The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning. *arXiv preprint arXiv:2403.03218*.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2023. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. *arXiv preprint arXiv:2310.04451*.

Liu, Z.; Dou, G.; Tan, Z.; Tian, Y.; and Jiang, M. 2024. Towards Safer Large Language Models through Machine Unlearning. *arXiv preprint arXiv:2402.10058*.

Mazeika, M.; Phan, L.; Yin, X.; Zou, A.; Wang, Z.; Mu, N.; Sakhaee, E.; Li, N.; Basart, S.; Li, B.; Forsyth, D.; and Hendrycks, D. 2024. HarmBench: A Standardized Evaluation Framework for Automated Red Teaming and Robust Refusal. *arXiv preprint arXiv:2402.04249*.

Meta AI. 2024. meta-llama/Llama-3.1-8B-Instruct. Hugging Face model card.

Mistral AI. 2024. mistralai/Mistral-7B-Instruct-v0.3. Hugging Face model card.

MosaicML. 2023. mosaicml/mpt-7b-instruct. Hugging Face model card.

Pham, M.; Marshall, K. O.; Hegde, C.; and Cohen, N. 2024. Robust Concept Erasure Using Task Vectors. *arXiv preprint arXiv:2404.03631*.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2023. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! *arXiv preprint arXiv:2310.03693*.

Qwen Team. 2025. Qwen/Qwen3-8B. Hugging Face model card.

Shen, G.; Zhao, D.; Feng, L.; He, X.; Wang, J.; Shen, S.; Tong, H.; Dong, Y.; Li, J.; Zheng, X.; and Zeng, Y. 2025. PandaGuard: Systematic Evaluation of LLM Safety against Jailbreaking Attacks. arXiv:2505.13862.

Xue, Y.; and Mirzasoleiman, B. 2025. LoRA is All You Need for Safety Alignment of Reasoning LLMs. *arXiv preprint arXiv:2507.17075*.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *arXiv preprint arXiv:2307.15043*.