AntiDote: Bi-level Adversarial Training for Tamper-Resistant LLMs

Debdeep Sanyal, Manodeep Ray, Murari Mandal

RespAI Lab, School of Computer Engineering
KIIT Bhubaneswar, India
{debdeep.respailab, manodeepray1}@gmail.com, murari.mandalfcs@kiit.ac.in

Abstract

The release of open-weight large language models (LLMs) creates a tension between advancing accessible research and preventing misuse, such as malicious fine-tuning to elicit harmful content. Current safety measures struggle to preserve the general capabilities of the LLM while resisting a determined adversary with full access to the model's weights and architecture, who can use full-parameter fine-tuning to erase existing safeguards. To address this, we introduce AntiDote, a bi-level optimization procedure for training LLMs to be resistant to such tampering. AntiDote involves an auxiliary adversary hypernetwork that learns to generate malicious Low-Rank Adaptation (LoRA) weights conditioned on the defender model's internal activations. The defender LLM is then trained with an objective to nullify the effect of these adversarial weight additions, forcing it to maintain its safety alignment. We validate this approach against a diverse suite of 52 red-teaming attacks, including jailbreak prompting, latent space manipulation, and direct weight-space attacks. AntiDote is upto 27.4% more robust against adversarial attacks compared to both tamper-resistance and unlearning baselines. Crucially, this robustness is achieved with a minimal trade-off in utility, incurring a performance degradation of upto less than 0.5% across capability benchmarks including MMLU, HellaSwag, and GSM8K. Our work offers a practical and compute efficient methodology for building open-weight models where safety is a more integral and resilient property.

Code — https://github.com/respailab/Antidote
Extended version — https://arxiv.org/pdf/2509.08000

1 Introduction

The increasing capability of open-weight large language models (LLMs) (Team et al. 2025; Yang et al. 2025; Grattafiori et al. 2024; DeepSeek-AI et al. 2025; AI et al. 2025; Üstün et al. 2024; Almazrouei et al. 2023) has democratized access to state-of-the-art AI, fueling rapid innovation and adoption in both academic and production environments. This openness, however, introduces a critical security vulnerability: with full access to model weights, a malicious actor can perform targeted fine-tuning to override built-in safety mechanisms and repurpose the model for harmful ends (Greshake et al. 2023; Shayegani et al. 2023; Shen et al. 2023; Shayegani, Dong, and Abu-Ghazaleh 2023; Anil et al. 2024). While guardrails offer protection for black-box models (Sanyal and Mandal 2025; Pawelczyk, Neel, and Lakkaraju 2024; Muresanu et al. 2024), they are irrelevant in a setting where the adversary has complete control. The central challenge, therefore, is to create models that are inherently resilient to such tampering, wedding the benefits of open-source transparency with the robust safety expected of closed-source systems. This paper addresses this challenge directly, proposing a method to instill tamper-resistance into open-weight LLMs by design.

This problem is particularly challenging due to the unrestricted threat model and the limitations of existing approaches. Approaches to this challenge largely fall into two paradigms. While early work focused on post-hoc detection via trojan signatures (Youssef et al. 2025) or output watermarking (Che et al. 2025; Kuditipudi et al. 2023; Nemecek, Jiang, and Ayday 2024), these methods are often brittle and can be bypassed (Che et al. 2025; Nemecek, Jiang, and Ayday 2024). A more robust paradigm, and the one we adopt, is to instill inherent resilience by design. However, even state-of-the-art methods in this domain (Tamirisa et al. 2024; Zhang and Koushanfar 2024; Huang et al. 2025a; **?**) suffer from two primary drawbacks. First, many rely on computationally prohibitive inner-outer loop optimization to simulate attacks (Tamirisa et al. 2024; Li et al. 2024). Second, they frequently struggle with a difficult trade-off, where increasing robustness against attacks leads to a significant degradation of the model's general-purpose capabilities (Tamirisa et al. 2024; Li et al. 2024; Huang et al. 2025c; Huang, Hu, and Liu 2024). These dual issues of efficiency and compromised utility have left a critical gap for a solution that is both effective and practical.

To bridge this gap, we introduce AntiDote, a computationally efficient bilevel optimization framework. Our key insight is to replace the expensive process of full adversarial fine-tuning with a parameter-efficient proxy: an auxiliary adversarial hypernetwork (Xiao et al. 2023; Lin et al. 2025). This dynamic is driven by a shared optimization objective based on Direct Preference Optimization (DPO) (Rafailov et al. 2024) using the BeaverTails (Ji et al. 2023) and the do-not-answer (Wang et al. 2023) datasets. Specifically, we create an adversarial game where the two players co-evolve: the hypernetwork is trained to generate Low-Rank Adaptation (LoRA) weights (Hu et al. 2021) that maximize the

likelihood of harmful responses, while the defender's own set of LoRA parameters are trained with the opposing objective to minimize that same likelihood even when the adversarial weights are applied. To address the safety-utility trade-off, we decouple this resistance training from capability preservation. In a separate optimization phase, without the adversarial LoRA weights, the model is tuned on a curated mixture of general-purpose datasets using a combined cross-entropy and KL-divergence loss against the base model. As our ablations demonstrate, this decoupled approach and diverse data mix are crucial for preventing catastrophic forgetting and preserving broad capabilities.

We conduct an exhaustive empirical validation to demonstrate that AntiDote is not only robust but also practical and general-purpose. Our analysis is built on three pillars: a large-scale evaluation of robustness across ten modern LLMs spanning from 0.6B to 27B parameters (Yang et al. 2025; Team et al. 2025); a granular stress test of resilience against a diverse suite of 52 red-teaming attacks; and a critical analysis of the safety-utility trade-off on standard capability benchmarks (Hendrycks et al. 2021a; Cobbe et al. 2021; Zellers et al. 2019; Saparov and He 2023). Across all evaluations, AntiDote consistently establishes a new state-of-the-art. It proves uniquely capable of instilling deep-seated safety, as evidenced by its superior performance on our granular attack suite, while simultaneously resolving the critical trade-off dilemma. As shown in our results, AntiDote consistently preserves, and often improves, fine-tune accuracy on benign tasks, demonstrating that robust safety does not have to come at the cost of the model's essential utility.

In summary, our contributions are:

1. **A Novel Bilevel Optimization Game for Resilience:** We introduce AntiDote, a new training paradigm that pits a state-aware adversarial hypernetwork against a parameter-efficient defender. The adversary learns to attack the model's internal activation patterns, forcing the defender to learn a deep and generalizable resilience.

2. **Efficient and Scalable Implementation:** We demonstrate the practical viability of our framework, achieved through a combination of fully parameter-efficient training, a reference-free DPO implementation that minimizes memory usage, and strategic CPU offloading during our interleaved training schedule.

3. **State-of-the-Art and Broad-Spectrum Robustness:** Across a diverse suite of 10 models and an extensive gauntlet of 52 red-teaming attacks, AntiDote achieves up to a 78% reduction in Harmful Score compared to the SFT baseline, decisively outperforming all prior art.

4. **Principled Mitigation of the Safety-Utility Trade-off:** We introduce and validate a decoupled optimization strategy that computes capability losses on a clean, unattacked model. This ensures gradient purity and allows 'Antidote' to achieve its state-of-the-art safety with virtually no degradation to the model's core utility.

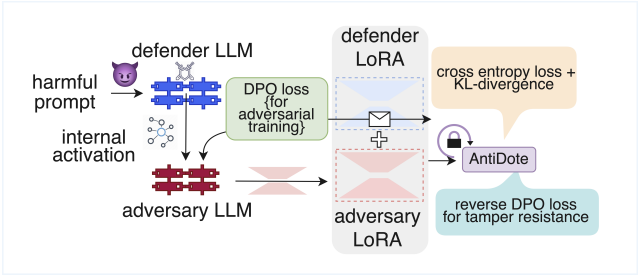

Figure 1: **An overview of the Antidote training framework**. When a harmful prompt is processed, the defender LLM's internal activations are fed to an adversarial hypernetwork. The adversary is trained via DPO loss to generate a malicious LoRA patch designed to compromise the defender. The defender, in turn, is trained with two distinct, decoupled objectives: a tamper-resistance loss computed on the attacked model to build resilience, and a capability-preservation loss computed on the clean model to maintain utility. This dynamic co-evolution forges a model that is not just aligned, but resilient by design.

2 Adversarial Tamper Resistance

2.1 Problem formulation

Our goal is to develop a language model that is inherently resilient to malicious fine-tuning while retaining its core capabilities. To anchor this abstract goal, consider a concrete attack: an adversary fine-tunes a public, open-weight model with the specific goal of making it generate malicious code when given a seemingly benign programming query. Our task is to create a model that resists this manipulation by design.

The Base Model and Data Distributions We consider a pre-trained open-weight language model, M, parameterized by weights $\theta \in \mathbb{R}^d$. The model is trained to generate a response y given a prompt x, conditioned on its parameters: $P(y|x;\theta)$. The initial state of our model, θ_{base}, is assumed to be aligned for safety and general utility. The model's behavior is shaped by two distinct data distributions:

1. **The Safety Distribution, $\mathcal{D}_{\text{safe}}$:** This distribution consists of prompts designed to probe for harmful or unsafe behavior. for each prompt $x_s \sim \mathcal{D}_{\text{safe}}$, we have a corresponding pair of responses: a preferred safe response, y_s, and a rejected harmful response, y_h. This distribution is critical for evaluating and training safety alignment.
2. **The Capability Distribution, $\mathcal{D}_{\text{cap}}$**: This distribution represents general-purpose instruction following and reasoning tasks. for each prompt $x_c \sim \mathcal{D}_{\text{cap}}$, we have a desired high-quality response, y_c. This distribution is used to ensure the model's core utility is not compromised.

The Adversarial Threat Model We operate under a strong threat model where an adversary has full, unrestricted access to the model's parameters, θ. The adversary's goal is to corrupt the model via fine-tuning to produce a compromised set of parameters, θ_{adv}. We deliberately assume this powerful adversary, one who is not limited by a specific attack algorithm or budget, to ensure our defense is

robust against unforeseen and future attack strategies. Such an adversary can easily bypass naive defenses, like simple weight regularization, which may penalize the magnitude of parameter changes but not their targeted, malicious direction. The adversary's objective is to select a fine-tuning action A from the space of all possible strategies, $\mathcal{A}$, to maximize the model's propensity to generate harmful content:

$$\max_{A \in \mathcal{A}} \quad \mathbb{E}_{(x_s, y_s, y_h) \sim \mathcal{D}_{\text{safe}}}[\log P(y_h | x_s; A(\theta))] \tag{1}$$

The Objective of Tamper-Resistance Our objective is to find a new set of model parameters, θ^*, that are resilient to the adversary's manipulations. A truly resilient model is one that maintains its safety alignment even after the adversary has performed their worst-case attack. This naturally leads to a min-max optimization problem, where we seek to find parameters θ that minimize the harm after the adversary has done their best to maximize it, all while preserving the model's core utility.

formally, we aim to solve for the optimal parameters θ^* as follows:

$$\theta^* = \arg\min_{\theta} \left(\max_{A \in \mathcal{A}} \mathcal{L}_{\text{harm}}(A(\theta)) \right) \quad \text{subject to} \quad \mathcal{L}_{\text{cap}}(\theta) \leq \epsilon \tag{2}$$

where:

- The negative DPO safety loss is defined as $\mathcal{L}_{\text{harm}}(\theta) = -\mathbb{E}_{(x_s, y_s, y_h) \sim \mathcal{D}_{\text{safe}}}[\log \sigma(\pi_\theta(y_s|x_s) - \pi_\theta(y_h|x_s))]$. We select the DPO loss as it directly models the adversary's goal: to invert the model's learned safety preferences by making it prefer the harmful response y_h over the safe one y_s. π_θ represents the model's log-probabilities. This choice aligns with prior work that also leveraged DPO loss for its inherent suitability to this problem (Tamirisa et al. 2024).
- $\mathcal{L}_{\text{cap}}(\theta)$ is a loss function measuring performance on the capability distribution $\mathcal{D}_{\text{cap}}$.
- ϵ is a small constant representing the maximum tolerable degradation in general capabilities.

Solving this min-max problem directly is intractable. The inner maximization over the adversary's action space $\mathcal{A}$ would require a complete, nested fine-tuning optimization to find the optimal attack A^* for *every single gradient step* of the outer minimization over θ. The computational cost of such a procedure makes it infeasible for modern LLMs. This intractability motivates our core contribution: a novel framework that creates a computationally feasible and effective proxy for this intractable optimization, which we detail in the following sections.

2.2 The Adversarial Hypernetwork

The direct optimization of the min-max objective in Equation 2 is intractable. The inner loop, $\max_{A \in \mathcal{A}} \mathcal{L}_{\text{harm}}(A(\theta))$, requires finding the optimal fine-tuning strategy A^*, a full optimization process in itself. Performing this for every gradient step of the outer minimization is computationally infeasible for modern LLMs and introduces severe optimization challenges, as the function $A(\theta)$ is not differentiable with respect to θ. This forces existing methods to rely on first-order approximations of the inner loop, leading to biased gradients that may underestimate the true adversarial threat.

To address this intractability, we introduce our core contribution: we replace the non-differentiable and expensive fine-tuning adversary with a **differentiable neural network proxy**, an adversarial hypernetwork. This approach offers two profound advantages:

1. **Computational Efficiency:** The hypernetwork has a fixed, modest size, and its forward pass (generating an attack) is orders of magnitude faster than performing even a single step of fine-tuning. This efficiency is constant, regardless of the size of the target LLM.
2. **Gradient Purity**: The entire process becomes fully differentiable. The hypernetwork acts as a function that maps the base model's state to adversarial parameters, allowing for clean, end-to-end gradient flow without the need for biased approximations.

We formally define our adversarial hypernetwork as H_ϕ, parameterized by weights ϕ. Its purpose is to generate a low-rank update, represented by LoRA matrices (U, V), that maximally compromises a specific linear layer l within the base model M_θ.

Architecture Inspired by Function The architecture of H_ϕ is not monolithic; it is a multi-stage network where each component is chosen to solve a specific challenge in generating these adversarial weights.

- **Input: State-Aware Attack Generation** The choice of input to H_ϕ is central to our framework's efficacy. Instead of using a static embedding of the harmful prompt, we use the internal state of the target LLM itself. formally, for a given prompt x and a target layer l, the input to the hypernetwork is the set of activation vectors $X_l = \{a_1, a_2, ..., a_N\}$ produced by that layer while processing the prompt. This makes our adversary **state-aware**, attacking the model's algorithm and reasoning process, not just the data's semantic content. By observing the model's internal "thoughts", the hypernetwork can adapt its attack in real-time as the base model's defenses evolve.
- **Core: From Set to Representation** The hypernetwork must process this set of activation vectors. We first employ a *self-attention mechanism* over the projected activations. This moves beyond a simple mean, allowing the network to learn a relational inductive bias and identify the most salient or "vulnerable" activation patterns within the set. The context-aware outputs are then pooled and processed by a deep stack of *Residual Feed-forward Blocks*, granting the network the expressive power needed to learn the complex mapping from activation statistics to optimal adversarial parameters.
- **Output: Heterogeneity-Aware LoRA Generation** An LLM contains many linear layers with varying dimensions (e.g., "q_proj" vs. "mlp.down_proj"). To handle this heterogeneity, H_ϕ employs a *multi-headed architecture* (Cordonnier, Loukas, and Jaggi 2021). It uses specialized, dimension-specific input and output heads for each unique

layer configuration, all connected to the shared core. This design allows for positive knowledge transfer and parameter efficiency. For a target layer l with input dimension d_{in} and output dimension d_{out}, the hypernetwork selects the corresponding output heads to generate the LoRA matrices:

$$(U_l, V_l) = H_\phi(X_l(x; \theta)) \tag{3}$$

where $U_l \in \mathbb{R}^{r \times d_{\text{in}}}$ and $V_l \in \mathbb{R}^{d_{\text{out}} \times r}$ are the generated LoRA matrices of rank r. The adversarial update to the target layer's weights, W_l, is then simply $\Delta W_l = V_l^T U_l$.

In this section, we have defined our adversary; a nimble, state-aware, and efficient hypernetwork. Now, we will detail how this adversary is trained in a game against the base model to instill robust, lasting resilience.

2.3 The Bi-level Optimization Game

Having established our differentiable adversary, H_ϕ, we now embed it within a bi-level optimization framework, a concept with deep roots in robust optimization and meta-learning. This framework operationalizes the min-max objective from Equation (2) as a practical, iterative game between two players: the **Adversary** (the hypernetwork H_ϕ) and the **Defender** (a set of trainable LoRA weights, θ_D, applied to the base model $M_{\theta_{\text{base}}}$).

The training proceeds in an interleaved $k : k$ schedule, where we alternate between optimizing the adversary and the defender. This dynamic co-evolution is critical: a static adversary would quickly become obsolete as the defender learns. By periodically re-training the adversary against the improving defender, we ensure the model is hardened against a continuously adapting threat, a significant advantage over methods that use a fixed attack simulation.

Phase 1: The Adversary's Turn (Maximization) In this phase, our goal is to strengthen the adversary. We freeze the defender's parameters, θ_D, and train the hypernetwork's parameters, ϕ, to become more effective at compromising the current state of the defended model. For a given harmful prompt x_s, the hypernetwork generates an adversarial LoRA patch based on the model's internal state. Let $X_l(x_s; \theta_{\text{base}} + \theta_D)$ denote the set of activations from a target layer l when processing prompt x_s. The patch is then generated as $(U_l, V_l) = H_\phi(X_l(x_s; \theta_{\text{base}} + \theta_D))$. This patch is dynamically applied to the defended model, creating a temporarily compromised model, θ_{adv}.[1]

The adversary's objective is to find parameters ϕ that maximize the DPO loss, thereby training the compromised model to prefer the harmful response y_h over the safe one y_s. formally, using a mini-batch stochastic estimate of the expectation, we update ϕ by ascending the gradient of the objective:

$$\mathcal{L}_{\text{adv}}(\phi) = \mathbb{E}_{(x_s, y_s, y_h) \sim \mathcal{D}_{\text{safe}}} \left[\log \sigma \left(\pi_{\theta_{\text{adv}}}(y_h|x_s) - \pi_{\theta_{\text{adv}}}(y_s|x_s)\right)\right] \tag{4}$$

[1]Here, $\theta_{\text{adv}} = (\theta_{\text{base}} + \theta_D) \oplus H_\phi(\cdot)$ where $\oplus$ denotes the operation of adding the generated LoRA weights (U_l, V_l) to the corresponding target layers l of the defended model.

In practice, this maximization is achieved by minimizing $-\mathcal{L}_{\text{adv}}$ via gradient descent.

Phase 2: The Defender's Turn (Minimization) In the second phase, we freeze the now-strengthened adversary, H_ϕ, and train the defender's LoRA parameters, θ_D. This phase has two distinct objectives, which we handle with a decoupled loss to ensure gradient purity[2] and training stability.

1. **Safety Objective: Resisting the Attack.** The primary goal is to make the model resilient. Using the frozen adversary, we generate the worst-case attack patch for a harmful prompt and apply it to the model. The defender's parameters, θ_D, are then updated to *minimize* the very same DPO loss that the adversary sought to maximize, forcing it to learn weights that counteract the adversarial patch. The safety loss is:

$$\mathcal{L}_{\text{safe}}(\theta_D) = -\mathbb{E}_{(x_s, y_s, y_h) \sim \mathcal{D}_{\text{safe}}} \left[\log \sigma \left(\pi_{\theta_{\text{adv}}}(y_s|x_s) - \pi_{\theta_{\text{adv}}}(y_h|x_s)\right)\right] \tag{5}$$

2. **Capability Objective: Preserving Utility.** Simultaneously, we must ensure the model remains helpful. This objective is computed on the *clean* defended model, $M_{\theta_{\text{base}}+\theta_D}$, *after* the adversarial patch has been removed. This decoupling is vital, as it provides a stable, stationary learning target for utility. The capability loss is a combination of a standard language modeling loss and a KL-divergence term to regularize against the original, unaligned base model:

$$\mathcal{L}_{\text{cap}}(\theta_D) = \mathbb{E}_{x_c, y_c \sim \mathcal{D}_{\text{cap}}} \left[\mathcal{L}_{\text{CE}}(M_{\theta_{\text{base}}+\theta_D})\right] + \beta \cdot D_{\text{KL}} \left(P(y|x_c; \theta_{\text{base}} + \theta_D) \,||\, P(y|x_c; \theta_{\text{base}})\right) \tag{6}$$

The total loss for the defender is a weighted sum of these objectives: $\mathcal{L}_{\text{defender}} = \mathcal{L}_{\text{safe}} + \lambda \mathcal{L}_{\text{CE}} + \beta \mathcal{L}_{\text{KL}}$. The hyperparameters λ and β, which balance the safety-utility trade-off, were empirically set to 0.8 and 0.3 respectively, a choice validated in our ablation studies (Section 5.5). This bi-level game, by separating the players and their objectives, creates a stable yet challenging curriculum that forges a model that is not only aligned by default but resilient by design.

3 Experimental Setup

3.1 Datasets

Our evaluation is grounded in a diverse and challenging suite of datasets. To train for safety ($\mathcal{D}_{\text{safe}}$), we combine BeaverTails and the do-not-answer dataset, exposing our hypernetwork to 16 unique harm categories (detailed in Appendix) to ensure it learns a generalizable concept of unsafe activation patterns. To preserve utility ($\mathcal{D}_{\text{cap}}$), we curate a diverse mix from LIMA (Zhou et al. 2023), Unnatural Instructions (Honovich et al. 2022), and the MATH dataset (Hendrycks

[2]By "gradient purity", we mean that the gradients for the capability objective are computed on the clean model, ensuring they are not "contaminated" by the presence of the adversarial patch and solely reflect the goal of utility preservation.

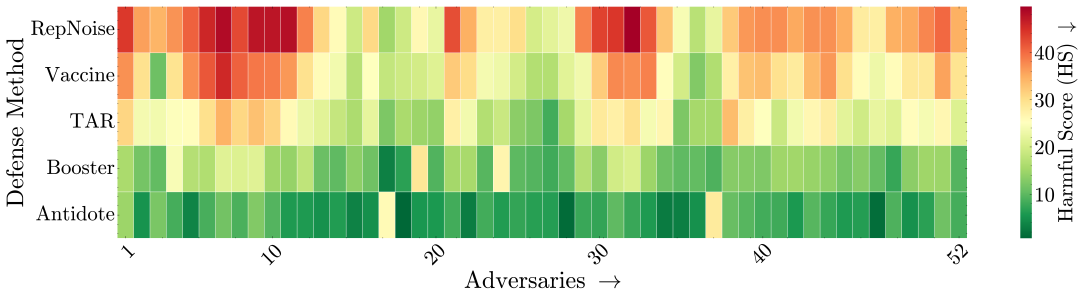

Figure 2: **Per-Attack Harmfulness Comparison Across 52 Red-Teaming Attacks.** A granular stress test evaluating each defense method against 52 distinct red-teaming vectors, from simple prompt injections to sophisticated, multi-layered attacks. A heatmap where rows are defenses and columns are attacks. Darker green indicates a lower, better Harmful Score (HS). The bottom bar for AntiDote demonstrates its broad-spectrum effectiveness. Its state-aware mechanism excels against attacks that manipulate the model's internal state, such as **role-playing (Adv 4)** and **adversarial suffixes (Adv 19)**, where gradient-based methods are often blind.

et al. 2021b), deliberately fostering a broad set of skills in instruction following, creativity, and reasoning. Finally, our models are evaluated on a comprehensive test set comprising 1,300 harmful instructions from BeaverTails (Ji et al. 2023), StrongREJECT (Souly et al. 2024), HarmBench (Mazeika et al. 2024), XSTest (Röttger et al. 2024), and a "do-anything-now" (Shen et al. 2024) evaluation set for Harmful Score, and 3,500 samples from MMLU (Hendrycks et al. 2021a), GSM8K (Cobbe et al. 2021), HellaSwag (Zellers et al. 2019), and PrOntoQA (Saparov and He 2023) for Fine-tune Accuracy.

3.2 Models and Baselines

To demonstrate the architectural and scale-agnostic nature of our framework, we conduct experiments on a suite of **10 distinct open-weight models** from six providers, with sizes ranging from 0.6B to 27B parameters. This diverse set, including models from the Qwen, Llama, Falcon, Aya, Gemma, and Mistral families (Yang et al. 2025; Grattafiori et al. 2024; Almazrouei et al. 2023; **?**; **?**), ensures our results are not an artifact of a single architecture. We compare Antidote against a comprehensive set of six strong baselines, including standard SFT, unlearning methods (RMU), and state-of-the-art alignment-stage defenses (TAR, RepNoise, Vaccine, and Booster), allowing for a clear assessment of the current state-of-the-art.

3.3 Metrics

To rigorously quantify the central safety-utility trade-off, we adopt the dual-metric standard from prior work (Huang et al. 2025b; Huang, Hu, and Liu 2024). We evaluate performance on two primary axes. Utility is measured via **Finetune Accuracy** (FA) on the test set of each capability benchmark. Concurrently, we measure safety via **Harmful Score** (HS), which is the percentage of unsafe outputs on our harmfulness test set, as judged by the classifier from (Ji et al. 2023). This dual-axis evaluation allows for a complete and fair comparison of all methods. Detailed FA calculation methodologies are provided in the Appendix.

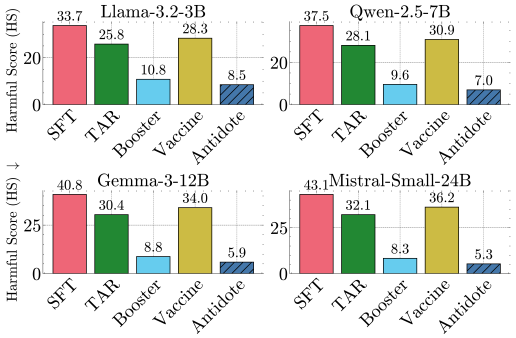

Figure 3: **AntiDote Achieves State-of-the-Art Robustness Across Diverse Models.** We compare the post-attack Harmful Score (HS) of AntiDote against strong baselines after fine-tuning on a mixed dataset. While gradient-based methods like Booster effectively penalize the immediate harmful loss, AntiDote's hypernetwork learns to recognize and counteract the underlying *compromised activation states* that lead to failure.

3.4 Compute and Reproducibility

All experiments were conducted with 3 NVIDIA A6000 GPUs. We used the AdamW optimizer for both the Antidote adversary and the defender, training for 8 epochs. The defender's LoRA parameters were configured with a rank of 16 and an alpha of 32. The adversary being the smaller model, was trained with a learning rate of 2e-4 and the defender was trained with a more conservative learning rate of 3e-5 to ensure stable convergence to a robust final state.

4 Experiments and Results

Robustness to Harmful training To simulate a realistic attack scenario, all models were fine-tuned on a dataset with a 20:80 mixture of harmful to benign data, and were evaluated on a held out set containing a 50:50 mixture of harmful to benign data. The comprehensive results across ten different models are presented in Table 2, with further analysis on different harmful data ratios delegated to Appendix.

Models	MMLU		GSM8K		HellaSwag		PrOntoQA		Average	
	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓
SFT	75.2 ± 0.03	15.5 ± 0.04	35.1 ± 0.06	14.8 ± 0.05	85.6 ± 0.02	16.1 ± 0.03	**68.9 ± 0.03**	14.2 ± 0.04	66.2 ± 0.04	15.2 ± 0.04
RMU	73.1 ± 0.04	12.4 ± 0.03	33.5 ± 0.05	11.9 ± 0.06	83.9 ± 0.03	13.0 ± 0.03	66.2 ± 0.04	11.5 ± 0.04	64.2 ± 0.04	12.2 ± 0.04
Booster	75.0 ± 0.02	7.1 ± 0.03	34.3 ± 0.05	**6.5 ± 0.04**	85.4 ± 0.02	8.8 ± 0.02	68.0 ± 0.03	6.9 ± 0.03	65.8 ± 0.03	7.3 ± 0.03
TAR	74.5 ± 0.03	9.8 ± 0.04	34.2 ± 0.06	9.1 ± 0.05	84.8 ± 0.02	10.5 ± 0.03	67.5 ± 0.04	9.5 ± 0.04	65.3 ± 0.04	9.7 ± 0.04
RepNoise	74.1 ± 0.04	11.5 ± 0.03	33.9 ± 0.06	10.8 ± 0.06	84.2 ± 0.03	12.1 ± 0.02	67.0 ± 0.05	11.1 ± 0.03	64.8 ± 0.05	11.4 ± 0.04
Vaccine	74.8 ± 0.03	8.5 ± 0.02	34.5 ± 0.05	7.9 ± 0.05	85.1 ± 0.02	8.5 ± 0.03	67.9 ± 0.03	8.1 ± 0.03	65.6 ± 0.03	8.4 ± 0.03
AntiDote	**75.8 ± 0.03**	**3.1 ± 0.01**	**35.5 ± 0.04**	8.8 ± 0.03	**86.1 ± 0.05**	**6.4 ± 0.03**	68.3 ± 0.02	**6.8 ± 0.02**	**66.4 ± 0.01**	**6.3 ± 0.04**

Table 1: This experiment tests the critical safety-utility trade-off. We fine-tuned aligned models on four standard, benign capability benchmarks and measured their resulting task accuracy and safety. A successful method should maintain a high FA and a low HS. Most baselines show a clear compromise—improving safety often comes at the cost of task performance. AntiDote is the exception, achieving the highest average scores for both metrics. This is because our decoupled loss function trains for utility on a clean, unattacked model.

The results demonstrate AntiDote's ability to balance safety and utility across all scales. For instance, on the 27B parameter Gemma-3 model, AntiDote achieves a Harmful Score (HS) of only 9.8, a **78% reduction** compared to the standard SFT baseline, while simultaneously achieving the highest Fine-tune Accuracy (FA) of 95.3. This pattern is consistent across the board.

The superior FA score of both Antidote and Booster over the SFT baseline suggests their respective regularization schemes effectively prevent overfitting, promoting better generalization. However, the crucial distinction lies in their safety mechanisms. Antidote's state-aware hypernetwork learns to recognize and counteract the model's internal failure modes directly. This provides a more fundamental defense than the gradient-based penalties of prior work, which can be bypassed by attacks designed to mask their semantic intent. This principle also explains the struggles of unlearning methods like RMU; their approach of targeting individual data points proves insufficient for resisting systemic, behavioral manipulation.

To visualize these trends more directly, Figure 3 presents the Harmful Score results for four representative models, reinforcing our primary claim of AntiDote providing a defense that both highly effective and consistently scalable across different model architectures and sizes.

Utility Preservation For this experiment, models aligned by each defense method were fine-tuned using a 20:80 harmful-to-benign data split on each benchmark dataset. We then measure both their task-specific Fine-tune Accuracy (FA) and their resulting Harmful Score (HS).

The results, detailed in Table 1, reveal that **AntiDote uniquely excels on both axes, demonstrating a clear break from the compromises made by prior methods.** On average, AntiDote achieves the highest Fine-tune Accuracy while simultaneously recording the lowest Harmful Score. This performance is attributed to the decoupled optimization strategy detailed in Section 3.3, as we have detailed in Appendix. By computing the capability preservation loss ($\mathcal{L}_{\text{cap}}$) on the clean, unattacked model, we ensure the gradient signal for utility is pure and unconfounded by the adversarial objective. This allows the defender's LoRA weights to learn how to be helpful and how to be safe as two independent, non-interfering skills.

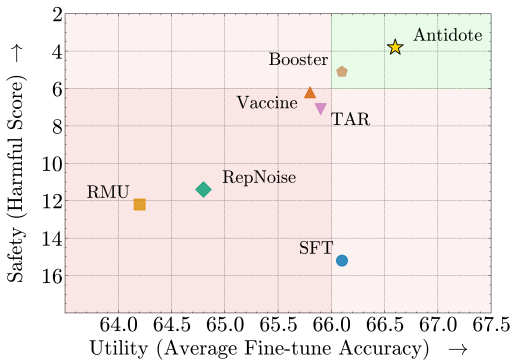

Figure 4: **AntiDote Breaks the Safety-Utility Trade-off Frontier.** We evaluate the trade-off between model utility (Average FA) and safety (HS). Most defenses operate along a trade-off curve, sacrificing utility for safety. AntiDote places itself in the optimal quadrant because our **decoupled optimization** computes the capability loss on a clean, unattacked model, providing unconfounded gradient signal for utility.

In contrast, other methods clearly exhibit the trade-off. Unlearning-based methods like RMU achieve a lower HS than the SFT baseline, but at a steep cost to utility. Booster once again proves to be a strong baseline, finding an effective balance and even slightly improving FA over SFT. However, its unified loss function still forces a compromise, resulting in a higher HS and lower FA compared to AntiDote. The same dynamic is best visualized in the trade-off frontier plot in Figure 4, where AntiDote resides in the optimal quadrant as compared to its baselines.

Red-teaming Attacks We evaluated our aligned models against our comprehensive suite of 52 distinct red-teaming attacks (refer to Appendix for details). The results are visualized in the heatmap in Figure 2, which provides a stark visual narrative of each defense's strengths and weaknesses.

The immediate takeaway from the figure is the effectiveness of AntiDote. As we can observe in the bottom row, AntiDote maintains low Harmful Score across the vast majority of attack vectors. This visual evidence provides strong support for our central claim: AntiDote offers a defense that is both powerful and general-purpose.

Models	SFT		RMU		Booster		TAR		Repnoise		Vaccine		AntiDote	
	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓	FA ↑	HS ↓
Qwen-2.5-0.6B	88.1	31.5	85.2	28.1	88.0	8.1	87.5	24.1	87.9	30.8	87.8	26.5	**88.5**	**5.1**
Llama-3.2-3B	90.2	33.7	87.1	29.5	90.3	8.3	89.4	25.8	90.1	32.9	**90.9**	28.3	90.8	**5.3**
Falcon-H1-7B	89.5	36.2	88.4	31.8	91.1	8.6	91.4	27.2	91.3	35.1	91.1	30.1	**91.9**	**5.7**
Llama-3.1-8B	90.9	38.1	89.0	33.2	92.3	8.8	91.5	28.5	92.2	37.5	92.0	31.8	**92.8**	**5.9**
Qwen-2.5-7B	91.8	37.5	88.7	32.6	91.7	9.1	90.9	28.1	91.9	36.8	91.3	30.9	**92.2**	**6.9**
Aya expanse-8B	90.9	39.4	89.5	34.1	92.8	**9.6**	92.0	29.8	92.7	38.6	93.1	32.7	**93.3**	9.8
Gemma-3-12B	93.5	40.8	90.1	35.5	93.4	9.5	92.6	30.4	93.3	40.1	93.1	34.0	**93.9**	**9.4**
Qwen-2.5-14B	**93.8**	41.5	90.5	36.2	93.7	9.9	92.9	31.0	93.6	40.8	93.4	34.8	93.2	**7.2**
Mistral-Small-24B	93.6	43.1	91.2	37.8	94.5	10.8	93.7	32.1	94.4	42.4	94.8	36.2	**95.0**	**8.5**
Gemma-3-27B	94.1	44.2	91.5	38.9	95.0	13.5	94.2	33.0	94.9	43.5	94.7	37.1	**95.3**	**9.8**

Table 2: We evaluate AntiDote against state-of-the-art baselines on ten different open-weight models with Fine-tune Accuracy, measuring utility, and Harmful Score, measuring safety. Unlearning methods like RMU often sacrifice FA for safety. Strong alignment-stage defenses like Booster find a better balance. Antidote consistently achieves the highest FA while simultaneously recording the lowest HS. This is a direct result of our framework's ability to learn a robust defense without interfering with the model's core knowledge.

Attacks like **Adversarial Suffixes (Adv 19)** or **Role-Playing (Adv 4)** succeed not by using overtly "harmful" tokens, but by subtly manipulating the model's internal computational state into a vulnerable configuration. Booster, which relies on the local gradient of the harmful loss, can be blind to these manipulations as the semantic gradient is weak. AntiDote's hypernetwork is trained to recognize the *anomalous activation patterns* that these attacks produce, allowing it to identify and counteract the threat at a more fundamental level. It defends against the compromised state itself, not just the prompt that caused it.

However, for attacks that leverage **Hypothetical Framing (Adv 37)** or **Distractor Instructions (Adv 17)**, Booster and TAR achieve marginally better scores than AntiDote. These specific attacks succeed by making the model's internal state appear overwhelmingly benign, by "diluting" the harmful signal with a flood of safe instructions. In these specific cases, AntiDote's hypernetwork, which looks for anomalous activation patterns, is effectively "fooled" by the seemingly normal internal state. The more direct, gradient-based check of a method like Booster is less susceptible to this high-level contextual misdirection.

Computational Efficiency A practical defense must also be computationally viable. While methods like RepNoise and TAR incur significant overhead, Antidote's resource usage is highly competitive. Notably, for the 12B parameter model, Antidote is nearly as fast as the strong Booster baseline while requiring significantly less GPU memory (73.6 GB vs. 82.5 GB). This efficiency stems from three deliberate engineering choices:

1. **Fully Parameter-Efficient Training:** Both our defender and our hypernetwork adversary are trained using LoRA. This means that at any given time, we are only updating a tiny fraction of the total model parameters, drastically reducing the optimizer's memory footprint and the time required for each gradient update.
2. **A memory-optimized DPO implementation** Our fully parameter-efficient approach naturally enables a highly memory-optimized DPO implementation. We obviate the need to store a full reference model in VRAM, as the reference state can be dynamically reproduced by simply changing the defender's LoRA adapter to the original adapter.

5 Conclusion

In this work, we addressed the critical challenge of making open-weight large language models resilient to malicious fine-tuning. We moved beyond traditional defenses by proposing Antidote, a novel bilevel optimization framework where a state-aware adversarial hypernetwork learns to find and exploit vulnerabilities in the model's internal representations. The base model, in turn, is trained to defend against this adaptive, evolving adversary, forging a deep and durable resilience. Our comprehensive experiments demonstrate that Antidote establishes a new state-of-the-art. It consistently and significantly outperforms existing methods across a diverse suite of models and 52 red-teaming attacks. Crucially, by cleanly decoupling the objectives for safety and capability, Antidote breaks the long-standing trade-off frontier, delivering a model that is both safer and more capable than those produced by prior art. Our approach shifts the paradigm from post-hoc patching to proactive immunization, showing that resilience can be woven into the fabric of the model itself. While our framework marks a significant step forward, key challenges remain, such as extending this dynamic defense to novel attack classes beyond our extensive test suite (see Appendix for a full discussion of limitations). We hope this work inspires a renewed focus on resilience as a foundational property of open-source AI.

Acknowledgments

This research is supported by the Anusandhan National Research Foundation (ANRF) erstwhile, Science and Engineering Research Board (SERB) India, under grant SRG/2023/001686.

References

AI, .; :; Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Wang, G.; Li, H.; Zhu, J.; Chen, J.; Chang, J.; Yu, K.; Liu, P.; Liu, Q.; Yue, S.; Yang, S.; Yang, S.; Xie, W.; Huang, W.; Hu, X.; Ren, X.; Niu, X.; Nie, P.; Li, Y.; Xu, Y.; Liu, Y.; Wang, Y.; Cai, Y.; Gu, Z.; Liu, Z.; and Dai, Z. 2025. Yi: Open Foundation Models by 01.AI. arXiv:2403.04652.

Almazrouei, E.; Alobeidli, H.; Alshamsi, A.; Cappelli, A.; Cojocaru, R.; Debbah, M.; Étienne Goffinet; Hesslow, D.; Launay, J.; Malartic, Q.; Mazzotta, D.; Noune, B.; Pannier, B.; and Penedo, G. 2023. The Falcon Series of Open Language Models. arXiv:2311.16867.

Anil, C.; Durmus, E.; Sharma, M.; Benton, J.; Kundu, S.; Batson, J.; Rimsky, N.; Tong, M.; Mu, J.; Ford, D.; Mosconi, F.; Agrawal, R.; Schaeffer, R.; Bashkansky, N.; Svenningsen, S.; Lambert, M.; Radhakrishnan, A.; Denison, C. E.; Hubinger, E.; Bai, Y.; Bricken, T.; Maxwell, T.; Schiefer, N.; Sully, J.; Tamkin, A.; Lanham, T.; Nguyen, K.; Korbak, T.; Kaplan, J.; Ganguli, D.; Bowman, S. R.; Perez, E.; Grosse, R.; and Duvenaud, D. 2024. Many-shot Jailbreaking. In *Neural Information Processing Systems*.

Che, Z.; Casper, S.; Kirk, R.; Satheesh, A.; Slocum, S.; McKinney, L. E.; Gandikota, R.; Ewart, A.; Rosati, D.; Wu, Z.; et al. 2025. Model tampering attacks enable more rigorous evaluations of llm capabilities. *arXiv preprint arXiv:2502.05209*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Cordonnier, J.-B.; Loukas, A.; and Jaggi, M. 2021. Multi-Head Attention: Collaborate Instead of Concatenate. arXiv:2006.16362.

DeepSeek-AI, A. L.; et al. 2025. DeepSeek-V3 Technical Report. arXiv:2412.19437.

Grattafiori, A.; et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Greshake, K.; Abdelnabi, S.; Mishra, S.; Endres, C.; Holz, T.; and Fritz, M. 2023. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021a. Measuring Massive Multitask Language Understanding. arXiv:2009.03300.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021b. Measuring Mathematical Problem Solving With the MATH Dataset. arXiv:2103.03874.

Honovich, O.; Scialom, T.; Levy, O.; and Schick, T. 2022. Unnatural Instructions: Tuning Language Models with (Almost) No Human Labor. arXiv:2212.09689.

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:2106.09685.

Huang, T.; Hu, S.; Ilhan, F.; Tekin, S. F.; and Liu, L. 2025a. Booster: Tackling Harmful Fine-tuning for Large Language Models via Attenuating Harmful Perturbation. arXiv:2409.01586.

Huang, T.; Hu, S.; Ilhan, F.; Tekin, S. F.; and Liu, L. 2025b. Booster: Tackling Harmful Fine-tuning for Large Language Models via Attenuating Harmful Perturbation. arXiv:2409.01586.

Huang, T.; Hu, S.; Ilhan, F.; Tekin, S. F.; and Liu, L. 2025c. Virus: Harmful Fine-tuning Attack for Large Language Models Bypassing Guardrail Moderation. arXiv:2501.17433.

Huang, T.; Hu, S.; and Liu, L. 2024. Vaccine: Perturbation-aware Alignment for Large Language Models against Harmful Fine-tuning Attack. arXiv:2402.01109.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Zhang, C.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. arXiv:2307.04657.

Kuditipudi, R.; Thickstun, J.; Hashimoto, T.; and Liang, P. 2023. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*.

Li, N.; Pan, A.; Gopal, A.; Yue, S.; Berrios, D.; Gatti, A.; Li, J. D.; and Dombrowski, A.-K. 2024. The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning. arXiv:2403.03218.

Lin, X.; Acharya, M.; Roy, A.; and Jha, S. 2025. TeleLoRA: Teleporting Model-Specific Alignment Across LLMs. *ArXiv*, abs/2503.20228.

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:2402.04249.

Muresanu, A.; Thudi, A.; Zhang, M. R.; and Papernot, N. 2024. Unlearnable Algorithms for In-context Learning. arXiv:2402.00751.

Nemecek, A.; Jiang, Y.; and Ayday, E. 2024. Topic-Based Watermarks for Large Language Models. *arXiv preprint arXiv:2404.02138*.

Pawelczyk, M.; Neel, S.; and Lakkaraju, H. 2024. In-Context Unlearning: Language Models as Few Shot Unlearners. arXiv:2310.07579.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290.

Röttger, P.; Kirk, H. R.; Vidgen, B.; Attanasio, G.; Bianchi, F.; and Hovy, D. 2024. XSTest: A Test Suite for Identifying Exaggerated Safety Behaviours in Large Language Models. arXiv:2308.01263.

Sanyal, D.; and Mandal, M. 2025. Agents Are All You Need for LLM Unlearning. In *Second Conference on Language Modeling*.

Saparov, A.; and He, H. 2023. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought. arXiv:2210.01240.

Shayegani, E.; Dong, Y.; and Abu-Ghazaleh, N. B. 2023. Jailbreak in pieces: Compositional Adversarial Attacks on Multi-Modal Language Models. In *International Conference on Learning Representations*.

Shayegani, E.; Mamun, M. A. A.; Fu, Y.; Zaree, P.; Dong, Y.; and Abu-Ghazaleh, N. B. 2023. Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks. *ArXiv*, abs/2310.10844.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2023. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2024. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. arXiv:2308.03825.

Souly, A.; Lu, Q.; Bowen, D.; Trinh, T.; Hsieh, E.; Pandey, S.; Abbeel, P.; Svegliato, J.; Emmons, S.; Watkins, O.; and Toyer, S. 2024. A StrongREJECT for Empty Jailbreaks. arXiv:2402.10260.

Tamirisa, R.; Bharathi, B.; Phan, L.; Zhou, A.; Gatti, A.; Suresh, T.; Lin, M.; Wang, J.; Wang, R.; Arel, R.; et al. 2024. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; et al. 2025. Gemma 3 Technical Report. arXiv:2503.19786.

Wang, Y.; Li, H.; Han, X.; Nakov, P.; and Baldwin, T. 2023. Do-Not-Answer: A Dataset for Evaluating Safeguards in LLMs. arXiv:2308.13387.

Xiao, Z.; Held, W.; Liu, Y.; and Yang, D. 2023. Task-Agnostic Low-Rank Adapters for Unseen English Dialects. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7857–7870. Singapore: Association for Computational Linguistics.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. arXiv:2505.09388.

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

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? arXiv:1905.07830.

Zhang, R.; and Koushanfar, F. 2024. Watermarking Large Language Models and the Generated Content: Opportunities and Challenges. In *2024 58th Asilomar Conference on Signals, Systems, and Computers*, 1779–1786. IEEE.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; Zhang, S.; Ghosh, G.; Lewis, M.; Zettlemoyer, L.; and Levy, O. 2023. LIMA: Less Is More for Alignment. arXiv:2305.11206.

Üstün, A.; Aryabumi, V.; Yong, Z.-X.; Ko, W.-Y.; D'souza, D.; Onilude, G.; Bhandari, N.; Singh, S.; Ooi, H.-L.; Kayid, A.; Vargus, F.; Blunsom, P.; Longpre, S.; Muennighoff, N.; Fadaee, M.; Kreutzer, J.; and Hooker, S. 2024. Aya Model: An Instruction Finetuned Open-Access Multilingual Language Model. arXiv:2402.07827.