Self-Mined Hardness for Safety Fine-Tuning

Anonymous Authors[1]

Abstract

Safety fine-tuning of language models typically requires a curated adversarial dataset. We take a different approach: score each candidate prompt's difficulty by how often the target model's own rollouts are judged harmful, then fine-tune on the hardest prompts paired with the model's own non-jailbroken rollouts. On Llama-3-8B-Instruct and Llama-3.2-3B-Instruct, this approach cuts the WildJailbreak attack success rate from 11.5% and 20.1% down to 1–3%, but pushes refusal on jailbreak-shaped benign prompts from 14–22% to 74–94%. Interleaving the same hard prompts 1:1 with adversarially-framed benign prompts (prompts that look like jailbreaks but have benign intent) cuts that refusal back down to 30–51% on 8B and 52–72% on 3B, at a cost of 2–6 percentage points of attack success rate. Within the mixed regime, training on the hardest half of the eligible pool rather than a random half cuts the remaining ASR by 35–50% (about 3 percentage points) on both models.

1. Introduction

Alignment fine-tuning (Ouyang et al., 2022; Bai et al., 2022) makes instruction-tuned language models less likely to produce harmful content. But adversarial prompts (role-play setups, hypothetical framings, persona instructions) can get them to answer harmful requests they would otherwise refuse (Wei et al., 2023; Zou et al., 2023). The standard fix is more safety-tuning on adversarial prompts, drawn either from human attack corpora like WildJailbreak (Jiang et al., 2024) or from hand-designed clean/biased pairs like those in Bias-Augmented Consistency Training (Chua et al., 2024). Both approaches start from a curated adversarial dataset.

We take a different approach. For each candidate adversarial prompt, we sample K rollouts (responses) from the target model and label each rollout with a three-judge safety ensemble. A prompt's *harmful rate*, the fraction of rollouts majority-labeled as harmful, becomes its difficulty score. Prompts that are sometimes but not always jailbroken form the *eligible pool*: these have both a training signal to learn from and a non-jailbroken rollout we can use as the supervised target. Sorting this pool by harmful rate gives a natural ordering from hardest to easiest. We need no external clean dataset or hand-designed adversarial families.

But this approach has a clear tradeoff. Training on the hardest self-mined prompts drives WildJailbreak ASR from 11.5% down to 2–3% on Llama-3-8B-Instruct and from 20.1% down to about 1% on Llama-3.2-3B-Instruct (Dubey et al., 2024), but pushes refusal on plain benign prompts from about 5% to 32–55%, and refusal on jailbreak-shaped benign prompts from 14–22% to 74–94% (§5.1). The model has learned to refuse *anything that looks like a jailbreak*, not just harmful requests. We address this by interleaving the adversarial sequence 1:1 with *adversarially-framed benign* prompts (*adv-benign* for short): long, role-play-style prompts that look like jailbreaks but have benign intent, paired with the model's non-refused rollouts. Mixing brings refusal on jailbreak-shaped benign prompts down to 30–51% on 8B and 52–72% on 3B, at a cost of 2–6 percentage points of ASR. The adversarial half is *identical* to the pure version.

We compare five fine-tuning baselines (*Hard*, *Random*, *Control*, *Hard-Mixed*, *Random-Mixed*) and report each at its 50%-of-pool checkpoint. At that checkpoint, Hard has trained on the top 50% hardest prompts from the eligible pool, Random on a random 50% of that same pool, and Control on the same number of vanilla benign prompts. Hard-Mixed and Random-Mixed additionally interleave the Hard or Random adversarial sequence 1:1 with adversarially-framed benign prompts. All five baselines share a canonical (prompt → safe-response) pairing, so any difference between Hard and Random is due only to which half of the pool the checkpoint has seen. The same 50% rule is applied to each model separately. Past that point, the remaining prompts are too easy to add useful adversarial signal (§3.5).[1]

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]Code: https://anonymous.4open.science/r/jailbreak-ICML-ED58/

Contributions.

- A self-contained pipeline that mines per-prompt jailbreak difficulty from the target model's own rollouts and pairs the hardest prompts with non-jailbroken rollouts as supervised training data.
- A 1:1 interleaving with adversarially-framed benign prompts as a targeted fix for the overrefusal that pure adversarial fine-tuning causes, evaluated against pure and random-selection baselines under a canonical prompt-to-target pairing.
- Test-set results on Llama-3-8B-Instruct and Llama-3.2-3B-Instruct across three safety and three overrefusal benchmarks, showing the tradeoff and the mitigation on both models.

2. Related Work

Safety alignment. Instruction tuning with human-preference data (Ouyang et al., 2022) and AI-feedback variants (Bai et al., 2022) are the standard approach for reducing a model's propensity to comply with harmful requests. Open-weights instruction-tuned models such as Llama 3 (Dubey et al., 2024) ship with this training already applied.

Jailbreak attacks and benchmarks. Adversarial prompting (Wei et al., 2023) and automated attack search (Zou et al., 2023) reliably elicit harmful completions from aligned models. WildJailbreak and WildGuardMix (Jiang et al., 2024; Han et al., 2024) contain large pools of adversarial prompts with judged responses. ClearHarm (AlignmentResearch, 2024) provides clearly harmful prompts.

Consistency training. Bias-Augmented Consistency Training (BCT) (Chua et al., 2024) trains models on paired *clean/biased* prompts to preserve clean behavior under adversarial reformulations. Irpan et al. (2025) concurrently apply BCT directly to jailbreaks and sycophancy, using the model's own responses as training targets. Our design is motivated by BCT but differs on two axes: the adversarial distribution is selected by self-mined per-prompt hardness (rather than paired prompt variants), and we include a prompt-selection ablation (hardest-half vs random-half) that a consistency-only objective does not isolate.

Curriculum learning and hard-example mining. Presenting examples in a meaningful order can affect optimization dynamics (Bengio et al., 2009). In computer vision, online hard-example mining focuses gradient updates on the currently misclassified examples (Shrivastava et al., 2016). We adapt this idea to safety fine-tuning at the prompt level: each prompt's difficulty is estimated by how often the model's own rollouts on it are judged harmful. We use LoRA (Hu et al., 2022) adapters as the parameter-efficient fine-tuning backbone.

3. Method

3.1. Notation

Let M be a fixed instruction-tuned target model. Given an adversarial prompt p we draw rollouts $r \sim M(\cdot \mid p)$ at sampling temperature $T = 1$. A safety judge maps a (prompt, response) pair to HARMFUL or UNHARMFUL. With three judges J_1, J_2, J_3, the majority label $\overline{J}(p, r)$ is HARMFUL when at least two judges return HARMFUL (and analogously for UNHARMFUL). For K rollouts $r_1, \ldots, r_K$ on prompt p, the *harmful rate* is

$$\mathrm{hr}(p) = \tfrac{1}{K} \left|\{i \mid \overline{J}(p, r_i) = \textsc{harmful}\}\right|. \tag{1}$$

We call p *eligible* if $0 < \mathrm{hr}(p) < 1$: only such prompts admit both a supervised target (a non-jailbroken rollout) and a meaningful learning signal (some failures to learn from).

3.2. Hard-Example Mining

We sample adversarial prompts $\mathcal{P}_h$ from the `adversarial_harmful` partition of WildJailbreak (Jiang et al., 2024), generate $K_h = 64$ rollouts per prompt, and classify each rollout with WildGuard (Han et al., 2024), MD-Judge v0.1 (Li et al., 2024), and Llama-Guard-3 (Inan et al., 2023). Three judges with distinct templates make the majority vote less sensitive to the idiosyncratic errors any one classifier makes on the long, role-play-style prompts that dominate WildJailbreak. The eligible set $\mathcal{E} = \{p : 0 < \mathrm{hr}(p) < 1\}$ feeds into every downstream baseline. For each eligible prompt, we pick one of its non-jailbroken rollouts uniformly at random as the training target, and reuse that same (prompt, target) mapping across every baseline, so any difference between Hard and Random comes only from which prompts each has seen, not from different training responses.

3.3. Adversarially-Framed Benign Supervision

Overrefusal is the opposite failure mode of safety tuning. It is driven by prompts that *look like* jailbreaks (role-play preambles, hypothetical framings, persona instructions) but whose underlying request is harmless. WildJailbreak's `adversarial_benign` partition contains exactly this kind of prompt. To teach the model to tell apart *looks like a jailbreak* from *actually asks for harm*, we mine non-refused rollouts for these prompts.

For each prompt q sampled from the adv-benign pool, we draw $K_b = 4$ rollouts at $T = 1$ and classify each for refusal using WildGuard (Han et al., 2024). We retain q if at least one of the four rollouts is non-refused and pick one of those as the compliant training target (again, the same target

is reused across baselines). We use $K_b = 4$ because we only need one non-refused rollout per prompt. The 8B model's per-rollout refusal rate on these prompts is 17.5%, so $K_b = 4$ retains 91% of sampled prompts. We apply the same K_b to 3B and it yields a similar survival rate. Refusal is also a more regular distinction than harmfulness, which lets us use a single classifier instead of the full three-judge ensemble.

3.4. The Five Baselines

Let $n = |\mathcal{E}|$. We sample a size-n subset $\mathcal{B}$ of the eligible adv-benign prompts and a size-n sample $\mathcal{V}$ of vanilla benign prompts. Table 1 summarises the five resulting datasets, and Algorithm 1 (in the appendix) gives the full construction procedure.

This construction has two properties worth noting. First, the same shared benign sequence $\mathcal{B}$ appears at the same positions in both Hard-Mixed and Random-Mixed. Only the adversarial half differs. Second, because the interleave is strict $[H_0, B_0, H_1, B_1, \dots]$ with no shuffling, Hard at any pure checkpoint k and Hard-Mixed at mixed checkpoint $2k$ have trained on *exactly the same* first k adversarial prompts in the same order. The pure-vs-mixed comparison is therefore isolated to the addition of the interleaved benign half.

3.5. Fine-Tuning and Reporting Cutoff

We fine-tune M via LoRA (Hu et al., 2022) adapters per baseline. Each baseline's JSONL file is read top to bottom, so at any intermediate checkpoint Hard has consumed the hardest prompts first and Random a random subset of the same pool (Table 1). Full hyperparameters are in Table 5. For each model we report at the point where the Hard baseline has trained on the top 50% of its eligible pool.

Why 50%. The bottom half of the eligible pool is nearly all easy prompts on both models. On 8B (pool size $|\mathcal{E}| = 2{,}388$), of the $1{,}194$ prompts in the bottom half, 76% have $\mathrm{hr}(p) \leq 5\%$ (jailbroken at most 3 of 64 rollouts) and 39% sit at the empirical floor $\mathrm{hr}(p) = 1/K_h \approx 1.6\%$ (jailbroken exactly once). 3B shows the same pattern (pool size $2{,}488$): of the $1{,}244$ bottom-half prompts, 70% have $\mathrm{hr}(p) \leq 5\%$ and 37% sit at the floor. Training past this point adds prompts with almost no adversarial signal on either model. The 50% cutoff corresponds to a hardness threshold of $\tau \approx 0.094$ on 8B and $\tau \approx 0.125$ on 3B ("prompts the base model fails on at least about 10–13% of the time"). The full harmful-rate distribution is in Table 6.

4. Experimental Setup

4.1. Target Models

We run the full pipeline on two target models from the Llama-3 family (Dubey et al., 2024): `meta-llama/Meta-Llama-3-8B-Instruct` (referred to as *8B*) and `meta-llama/Llama-3.2-3B-Instruct` (*3B*). Both models are instruction-tuned open-weights checkpoints. The 3B model lets us test whether the qualitative findings carry across scale, while sharing tokenizer and prompt format with the 8B run for an even comparison. Inference for rollouts, judging, and evaluation is served through vLLM (Kwon et al., 2023). LoRA adapters are swapped per checkpoint without reloading the base model.

4.2. Mining Scales

We apply the same mining protocol to both target models. On the hard-mining side, we sample $N_h = 5{,}000$ adversarial-harmful prompts and draw $K_h = 64$ rollouts per prompt. Table 2 reports the eligible counts: $2{,}388$ on 8B and $2{,}488$ on 3B.

On the benign side, we need exactly $|\mathcal{E}|$ adv-benign prompts to pair 1:1 with the eligible hard set. We sample $N_b = 3{,}000$ prompts at $K_b = 4$ rollouts each (enough to buffer against the refusal filter, which drops prompts where none of the four rollouts is non-refused) and keep the first $|\mathcal{E}|$ that pass. At the observed 17.5% per-rollout refusal rate on 8B, $K_b = 4$ retains over 90% of sampled prompts, comfortably above both models' eligible counts, and 3B has a similar survival rate.

4.3. Judges

Safety judging uses the majority vote of WildGuard (Han et al., 2024), MD-Judge v0.1 (Li et al., 2024), and Llama-Guard-3 (Inan et al., 2023). Refusal classification uses WildGuard alone. All judges run with greedy decoding.

4.4. Evaluation Datasets

We evaluate on six held-out test sets: three safety and three overrefusal. The safety sets are the WildJailbreak eval split, the adversarial-harmful partition of WildGuardMix (Han et al., 2024), and ClearHarm (AlignmentResearch, 2024). The overrefusal sets are `wildjailbreak_adv_benign` (adversarially-framed benign prompts from WildJailbreak's eval split), `wildguardmix_adv_benign` (adversarial-yet-unharmful prompts from WildGuardMix), and `wildguardmix_benign` (plain benign prompts from WildGuardMix). The first two overrefusal sets probe the failure mode we care about most (prompts that share the surface structure of jailbreaks but have benign intent). The

Table 1. The five fine-tuning baselines. $\mathcal{E}$ is the eligible adversarial-harmful set; $\mathcal{B}$ is a size-n sample of eligible adv-benign prompts; $\mathcal{V}$ is a size-n sample of vanilla benign prompts. Hard-side prompts are paired with their canonical safe target $\mathrm{safe_target}(p)$; benign-side prompts with their canonical compliant target $\mathrm{compliant_target}(q)$; control prompts with one $T{=}1$ rollout from the target model.

Baseline	Prompts	Length	Order	Purpose
Hard	$\mathcal{E}$	n	hr-descending (stable)	train on the hardest half of $\mathcal{E}$
Random	$\mathcal{E}$	n	random permutation	train on a random half of $\mathcal{E}$; selection ablation for Hard
Control	$\mathcal{V}$	n	random permutation	compute-matched null
Hard-Mixed	$\mathcal{E} \cup \mathcal{B}$	$2n$	alternating; Hard in hr-desc order	safety + overrefusal mitigation
Random-Mixed	$\mathcal{E} \cup \mathcal{B}$	$2n$	alternating; Random in random order	selection ablation under mixing

Table 2. Adversarial-harmful mining yield on both target models, from the same 5,000-prompt sample with $K_h = 64$ rollouts each. "Never jailbroken" prompts give no hardness signal, "always jailbroken" prompts have no safe rollout to use as a target, and only the *eligible* set enters the Hard and Random baselines. The smaller 3B model is more vulnerable, so fewer of its rollouts are refusals: it has fewer "never jailbroken" prompts and ends up with a slightly larger eligible pool.

Metric	8B	3B
Sampled adversarial-harmful prompts	5,000	5,000
Never jailbroken ($\mathrm{hr}(p) = 0$)	2,607	2,504
Always jailbroken ($\mathrm{ur}(p) = 0$, dropped)	5	8
Eligible	**2,388**	**2,488**

third acts as a sanity check on plain benign prompts.

Metrics. On safety datasets we report the attack success rate ASR under three-judge majority voting. On overrefusal datasets we report the mean refusal rate from WildGuard. All evaluation generations use greedy decoding.

5. Results

We report test-set results on both models at the 50%-of-pool checkpoint selected in §3.5. Each model is evaluated at its own checkpoint: the pure baselines (Hard, Random, Control) train on the top 50% of that model's eligible pool, and the mixed baselines interleave that same set 1:1 with an equal-sized adversarially-framed benign set. Table 3 reports the 8B numbers and Table 4 the 3B numbers. Both span all three safety and three overrefusal benchmarks. Figure 1 shows the headline comparison: safety drop and overrefusal cost for each regime on both models.

5.1. Pure Adversarial Fine-Tuning

The pure blocks of Tables 3 and 4 show the baselines after each model has trained on the top 50% of its own eligible pool. Pure adversarial fine-tuning does what it sets out to on both models: WildJailbreak ASR drops from 11.5% to 2.1–3.1% on 8B and from 20.1% to 0.9–1.1% on 3B, WildGuardMix drops to under 1% on both ($\leq$ 0.9% on 8B, 0.0% on 3B), and ClearHarm drops to essentially zero on both. But it has a large cost in overrefusal. Refusal on the two jailbreak-shaped benign sets rises from 20–22% at base to 74–84% on 8B, and from 14–22% to 90–94% on 3B, a larger jump for the smaller model. Refusal on plain benign prompts (the WildGuardMix benign set) rises from 4.5% to 32–36% on 8B and from 5.1% to 44–55% on 3B.

The compute-matched Control baseline does not explain this jump. Control rises by 8–10 pp on the two adv-benign sets on 8B and by 3–6 pp on 3B, and barely moves on plain benign (+0.2 pp on 8B, −1 pp on 3B). The Hard baseline, by contrast, rises by about 60 pp on the adv-benign sets on 8B and by 72–76 pp on 3B, at least 6× Control's effect on 8B and at least 11× on 3B. The overrefusal jump is therefore caused by the adversarial training data itself, not by generic side effects of LoRA fine-tuning.

5.2. 1:1 Adversarially-Framed Benign Mixing

The mixed blocks of Tables 3 and 4 show the baselines after training on the same top 50% of hardest prompts as in the pure regime, in the same order, interleaved 1:1 with the same number of adversarially-framed benign prompts. The adversarial half of mixed training is identical to the pure regime, only the benign half is added.

The cost on safety is modest on both models. On WildJailbreak, ASR rises from 2.1% to 5.1% for Hard-Mixed and from 3.1% to 7.9% for Random-Mixed on 8B (+3.0 and +4.8 percentage points), and from 1.1% to 3.4% and 0.9% to 6.8% on 3B (+2.3 and +5.9 pp). The three safety benchmarks still sit well below the base model on both. The benefit on overrefusal is much larger. Refusal on the two jailbreak-shaped benign sets drops from 74–84% in the pure regime to 30–51% in the mixed regime on 8B, and from 90–94% to 52–72% on 3B. Refusal on plain benign prompts drops from 32–36% to 14–17% on 8B and from 44–55% to 14–19% on 3B.

The size of the overrefusal reduction depends on the baseline. Random-Mixed cuts adv-benign refusal by 35–49 pp (a 41–62% relative drop from the corresponding pure baseline), roughly halving the damage. Hard-Mixed cuts by less: 22–35 pp (23–42% relative drop), about a third of the damage. On plain benign, both Hard-Mixed and Random-Mixed

Table 3. Test-set results on Llama-3-8B-Instruct at the 50%-of-pool checkpoint (§3.5). ASR = attack success rate (↓); Ref = refusal rate (↓). Best per column within each regime in **bold**. The two "adv-benign" overrefusal sets are adversarially-framed benign prompts (look like jailbreaks but have benign intent). The WildGuardMix benign set has plain benign prompts.

	Safety (ASR ↓)			Overrefusal (Refusal ↓)		
Baseline	WildJailbreak	WildGuardMix	ClearHarm	WildJailbreak adv-benign	WildGuardMix adv-benign	WildGuardMix benign
Pure baselines (trained on the top 50% of the eligible pool)						
Base	11.5%	3.2%	1.1%	**22.4%**	**20.2%**	**4.5%**
Control	12.8%	4.7%	6.8%	32.4%	28.4%	4.7%
Hard	**2.1%**	**0.3%**	0.1%	83.8%	80.4%	32.0%
Random	3.1%	0.9%	**0.0%**	78.6%	74.3%	35.7%
Mixed baselines (the same top 50% interleaved 1:1 with an equal-sized adv-benign set)						
Base	11.5%	3.2%	1.1%	**22.4%**	**20.2%**	**4.5%**
Control (matched compute)	13.6%	5.0%	8.1%	30.5%	24.2%	5.5%
Hard-Mixed	**5.1%**	**2.9%**	0.1%	49.0%	51.2%	14.3%
Random-Mixed	7.9%	4.4%	**0.0%**	30.0%	39.8%	16.7%

Table 4. Test-set results on Llama-3.2-3B-Instruct at the 50%-of-pool checkpoint (§3.5). Best per column within each regime in **bold**. 3B's eligible pool is slightly larger than 8B's (Table 2), so 50% corresponds to a later optimizer step here than in Table 3.

	Safety (ASR ↓)			Overrefusal (Refusal ↓)		
Baseline	WildJailbreak	WildGuardMix	ClearHarm	WildJailbreak adv-benign	WildGuardMix adv-benign	WildGuardMix benign
Pure baselines (trained on the top 50% of the 3B eligible pool)						
Base	20.1%	6.5%	8.0%	**14.3%**	**21.8%**	5.1%
Control	11.8%	6.2%	10.6%	17.6%	28.1%	**4.1%**
Hard	1.1%	**0.0%**	**0.0%**	90.0%	94.1%	55.1%
Random	**0.9%**	**0.0%**	**0.0%**	91.4%	92.3%	43.5%
Mixed baselines (the same top 50% interleaved 1:1 with an equal-sized adv-benign set)						
Base	20.1%	6.5%	8.0%	**14.3%**	**21.8%**	5.1%
Control (matched compute)	15.7%	8.2%	16.6%	15.2%	26.8%	**3.9%**
Hard-Mixed	**3.4%**	**1.8%**	**0.0%**	66.2%	72.5%	19.0%
Random-Mixed	6.8%	3.8%	1.7%	52.4%	54.5%	13.7%

reduce refusal by more than half (53–69% relative).

5.3. Hardest Half vs Random Half

On safety, training on the hardest half helps clearly in the mixed regime on both models. The gap on WildJailbreak ASR is 2.8 pp on 8B (Hard-Mixed 5.1% vs Random-Mixed 7.9%) and 3.4 pp on 3B (3.4% vs 6.8%), similar magnitudes. In the pure regime, the gap is smaller and not always in Hard's favor: 1.0 pp on 8B (2.1% vs 3.1%) and −0.2 pp on 3B (1.1% vs 0.9%). The 3B pure case is within noise: Hard and Random both drive 3B's pure-baseline ASR essentially to the floor on all three safety sets (Table 4), so there is little room for prompt selection to make a difference. In the mixed regime, ASR stays above the floor on both models, so the selection effect becomes visible.

On refusal of jailbreak-shaped benign prompts, the comparison flips: in the mixed regime, Random-Mixed beats Hard-Mixed on both models. On 8B, Random-Mixed achieves 30.0% and 39.8% on the two adv-benign sets vs Hard-Mixed's 49.0% and 51.2% (an 11–19 pp gap in Random's favor). On 3B, the gap is 13.8 pp (52.4% vs 66.2% on WildJailbreak adv-benign) and 18.0 pp (54.5% vs 72.5% on WildGuardMix adv-benign). A possible explanation is that training on only the hardest prompts pushes the model toward aggressive refusal on jailbreak-shaped inputs, while Random's broader half (which also includes easier-looking adversarial prompts) teaches finer discrimination that generalizes to adv-benign. Neither pure regime shows this flip.

On plain benign prompts, the gap is small and inconsistent across models, so the mechanism seems tied to jailbreak-shaped benign prompts specifically.

6. Discussion

Why adv-benign rather than vanilla benign in the mix? Vanilla benign prompts look benign on the surface, and that is not where overrefusal fails. Adversarially-framed benign prompts have the same surface structure as jailbreaks (role-play, hypothetical, persona preambles) but ask for harmless content. Mixing these in is what teaches the model to tell

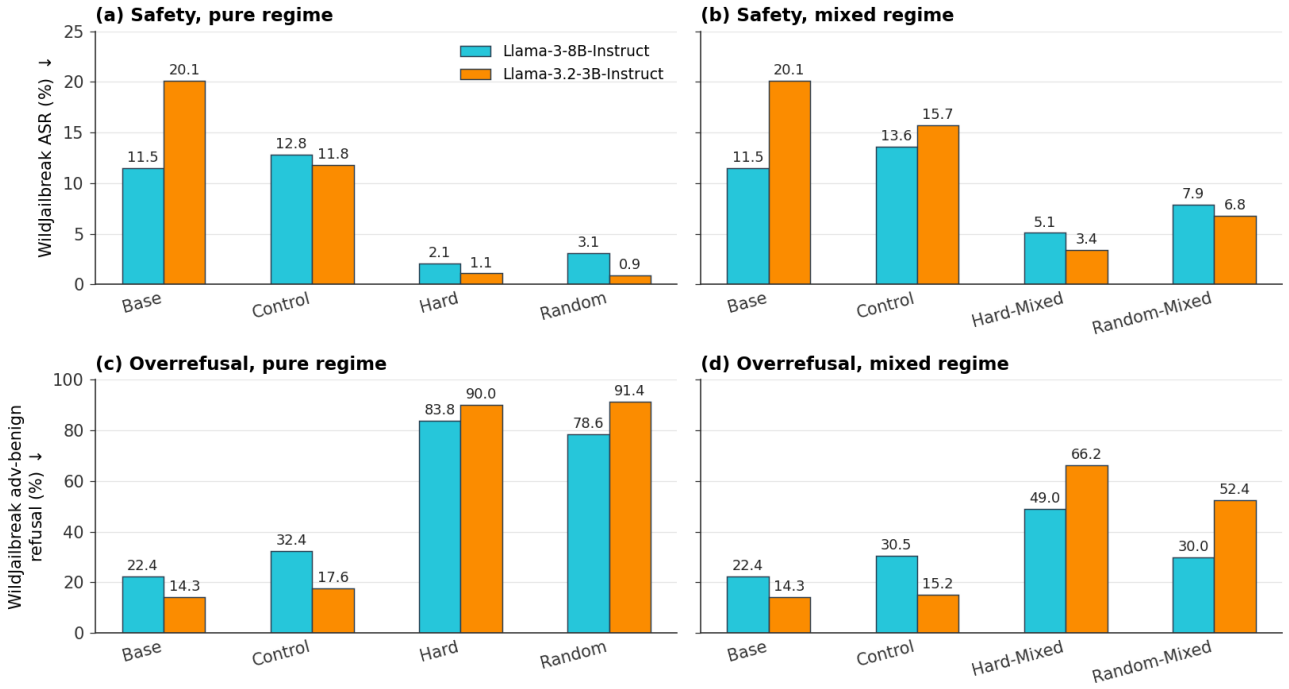

Figure 1. Headline test-set results. Lower is better in every panel. Each column is internally compute-matched: **left** (**a, c**) = pure regime (top 50% of each model's eligible pool). **Right** (**b, d**) = mixed regime (the same hardest prompts interleaved 1:1 with an equal-sized adv-benign set, so twice the training prompts). The Control bar in each subplot is the matched-compute reference for that regime: a different control checkpoint in (a, c) than in (b, d). **Top row (a, b):** WildJailbreak attack success rate. Pure Hard and Random drive ASR near zero on both models. Hard-Mixed and Random-Mixed pay a small ASR cost. **Bottom row (c, d):** WildJailbreak adv-benign refusal rate (the overrefusal failure mode). Pure Hard and Random push refusal above 78% on both models. Mixing brings refusal back to 30–66%.

"looks like a jailbreak" apart from "actually asks for harm" *on the failure-mode distribution*. The residual adv-benign refusal in the best mixed baseline is therefore the right figure to report: it is the hard part of the problem, not the easy one. The gap between the two models suggests that model capacity also matters for this discrimination task, even when the training signal is right.

Pure baselines reduce ASR more. Is that the right comparison? No. A model that refuses every prompt also has zero ASR, but it has zero utility. The right comparison is Pareto over (ASR, overrefusal): on both models, the mixed baselines sit at strictly better (ASR, overrefusal) tradeoffs than any pure baseline on both adv-benign sets, at a small ASR cost.

Focusing on the hardest prompts is a tradeoff. The hardest-half-vs-random-half comparison (§5.3) cuts in opposite directions across the two failure modes: focusing on the hardest prompts protects *safety* better, while the broader Random half generalizes better on adv-benign refusal. Hardest-first is therefore not strictly preferable. It shifts the (ASR, overrefusal) tradeoff toward safety.

7. Limitations and Future Work

Limitations. All reported results are on the Llama-3 family (Llama-3-8B-Instruct and Llama-3.2-3B-Instruct), so cross-family generalization remains untested. Even the best mixed baseline refuses 30–51% (8B) or 52–72% (3B) of jailbreak-shaped benign prompts. Mixing reduces the damage but does not close the gap to base. The 50%-of-pool cutoff was chosen after inspecting the hardness distributions, and although we argue in §3.5 that the justification depends only on the shape of the difficulty distribution, the post-hoc framing remains a caveat.

Future work. Future work should refine both the definition and use of prompt hardness. Our current hardness score is model-conditioned: a prompt is considered hard when the target model produces harmful responses on a large fraction of sampled rollouts. This is useful for mining model-specific failures, but does not show whether the prompt is intrinsically hard. Hardness could also be extended beyond frequency by incorporating the severity and semantic diversity of harmful rollouts, since prompts that elicit many distinct harmful completions may expose broader safety failures than prompts that repeatedly elicit the same response. Another direction is to use the always-jailbroken prompts

currently dropped by our pipeline: a stronger safety-aligned teacher could generate a safe response, the target model could rewrite it into its own distribution, and the result could be filtered by the same judges. Finally, future work should study curricula that explicitly balance ASR, overrefusal, and training efficiency, such as starting with adversarially-framed benign prompts before gradually increasing high-hardness harmful prompts.

8. Conclusion

We described a self-contained safety fine-tuning pipeline in which the adversarial training data and the supervised targets both come from the target model's own rollouts. Five controlled baselines built on a canonical prompt-to-target pairing separate the effects of prompt selection and benign interleaving. On Llama-3-8B-Instruct and Llama-3.2-3B-Instruct, pure adversarial fine-tuning produces a clear safety-vs-overrefusal tradeoff that compute-matched controls do not explain, and 1:1 adversarially-framed benign mixing recovers a meaningful share of the lost compliance at a small safety cost (2–6 pp ASR on WildJailbreak). The hardest-half-vs-random-half comparison points in opposite directions across safety and adv-benign refusal, so prompt selection shifts the (ASR, overrefusal) tradeoff rather than dominating it. The smaller model retains higher residual refusal even after mixing, suggesting model capacity also matters.

References

AlignmentResearch. ClearHarm: A benchmark of clearly harmful prompts. https://huggingface.co/datasets/AlignmentResearch/ClearHarm, 2024.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

Chua, J., Rees, E., Batra, H., Bowman, S. R., Michael, J., Perez, E., and Treutlein, J. Bias-augmented consistency training reduces biased reasoning in chain-of-thought. *arXiv preprint arXiv:2403.05518*, 2024.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Dziri, N., Choi, Y., and Sap, M. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. *arXiv preprint arXiv:2406.18495*, 2024.

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

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., and Khabsa, M. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Irpan, A., Turner, A. M., Kurzeja, M., Elson, D. K., and Shah, R. Consistency training helps stop sycophancy and jailbreaks. *arXiv preprint arXiv:2510.27062*, 2025.

Jiang, L., Rao, K., Han, S., Ettinger, A., Brahman, F., Kumar, S., Mireshghallah, N., Lu, X., Sap, M., Choi, Y., and Dziri, N. WildTeaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. *arXiv preprint arXiv:2406.18510*, 2024.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Li, L., Dong, B., Wang, R., Hu, X., Zuo, W., Lin, D., Qiao, Y., and Shao, J. SALAD-Bench: A hierarchical and comprehensive safety benchmark for large language models. *arXiv preprint arXiv:2402.05044*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shrivastava, A., Gupta, A., and Girshick, R. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 761–769, 2016.

Wei, A., Haghtalab, N., and Steinhardt, J. Jailbroken: How does LLM safety training fail? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zou, A., Wang, Z., Kolter, J. Z., and Fredrikson, M. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A. Baseline Construction Procedure

Algorithm 1 is the construction procedure referenced in §3.4.

Algorithm 1 Self-Mined Baseline Construction

Input: target model M; adversarial-harmful prompts $\mathcal{P}_h$; adversarial-benign prompts $\mathcal{P}_b$; vanilla-benign prompts $\mathcal{V}$; sample sizes N_h, N_b; rollout counts K_h, K_b; safety judges J_1, J_2, J_3; refusal judge J_r; seed.
Sample $\mathcal{P}_h \leftarrow$ uniform N_h-subset of `adversarial_harmful`.
Sample $\mathcal{P}_b \leftarrow$ uniform N_b-subset of `adversarial_benign`.
{Hard-example mining}
for $p \in \mathcal{P}_h$ **do**
  Draw K_h rollouts $r_{p,1}, \ldots, r_{p,K_h} \sim M(\cdot \mid p)$ at $T = 1$.
  Compute $\overline{J}(p, r_{p,i})$ via majority vote of J_1, J_2, J_3.
  Set $\mathrm{hr}(p), \mathrm{ur}(p)$ as in §3.2.
end for
$\mathcal{E} \leftarrow \{p : 0 < \mathrm{hr}(p) < 1\}$; $n \leftarrow |\mathcal{E}|$.
{Canonical safe-target map}
for $p \in \mathcal{E}$ **do**
  $\sigma(p) \leftarrow$ uniform draw from $\{i \mid \overline{J}(p, r_{p,i}) = \textsc{unharmful}\}$.
  $\mathrm{safe_target}(p) \leftarrow r_{p,\sigma(p)}$.
end for
{Adversarial-benign mining}
for $q \in \mathcal{P}_b$ **do**
  Draw K_b rollouts; classify each with J_r.
end for
$\mathcal{E}_b \leftarrow \{q : \exists\, i,\ J_r(q, r_{q,i}) = \textsc{complied}\}$.
$\mathcal{B} \leftarrow$ uniform n-subset of $\mathcal{E}_b$; draw canonical $\mathrm{compliant_target}(q)$ from non-refused rollouts.
{Baselines}
$\mathcal{D}_{\text{hard}} \leftarrow \mathcal{E}$ sorted stably by hr descending.
$\mathcal{D}_{\text{random}} \leftarrow$ random permutation of $\mathcal{E}$.
$\mathcal{D}_{\text{control}} \leftarrow n$ prompts from $\mathcal{V}$; one T=1 rollout each.
$\mathcal{D}_{\text{Hard-Mixed}} \leftarrow \textsc{interleave}(\mathcal{D}_{\text{hard}}, \mathcal{B})$.
$\mathcal{D}_{\text{Random-Mixed}} \leftarrow \textsc{interleave}(\mathcal{D}_{\text{random}}, \mathcal{B})$.
Output: five training JSONL files (preserved order).

B. Training Hyperparameters

Table 5. LoRA training hyperparameters.

Hyperparameter	Value
LoRA rank r	16
LoRA scaling α	32
LoRA dropout	0.05
LoRA target modules	$\{q_proj, v_proj\}$
Per-device batch size	2
Gradient accumulation	5
Effective batch size	10
Epochs	1
Learning rate	1×10^{-4}
Schedule	cosine, warmup ratio 0.03
Weight decay	0.01
Optimizer	AdamW
Precision	bf16
Gradient checkpointing	on (non-reentrant)
Max sequence length	1024
Sampler	sequential (no shuffle)
Checkpoint interval	every 5 optimizer steps (50 prompts)
Seed	55

C. Hardness Distribution on the Eligible Pool

Table 6 gives the full harmful-rate distribution over the eligible pool for both target models. Buckets are left-inclusive, right-exclusive: the 10% boundary appears only in the 10–20% row. Both distributions are heavily right-skewed. Together with Table 2, this is the quantitative basis for the 50%-of-pool cutoff argued in §3.5. 3B sits slightly higher in the mid-bands (40–79%) and slightly lower at the floor, consistent with its higher base ASR (20.1% vs 11.5%).

Table 6. Distribution of harmful rate $\mathrm{hr}(p)$ over the eligible pool $\mathcal{E}$ on both target models (counts and percentages of $\mathcal{E}$). Buckets are left-inclusive, right-exclusive.

$\mathrm{hr}(p)$ bucket	8B count	8B %	3B count	3B %
$\geq 80\%$	115	4.8%	116	4.7%
60–80%	124	5.2%	174	7.0%
40–60%	197	8.2%	267	10.7%
20–40%	370	15.5%	406	16.3%
10–20%	339	14.2%	360	14.5%
5–10%	502	21.0%	465	18.7%
$< 5\%$	741	31.0%	700	28.1%
Total $\lvert\mathcal{E}\rvert$	**2,388**		**2,488**	

D. LLM Usage

We used LLMs (ChatGPT and Claude) to assist with writing edits, LaTeX formatting, and code. All technical content, experiments, and claims were produced and verified by the authors.