TEST-TIME SAFETY ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work has shown that a model's input word embeddings can serve as effective control variables for steering its behavior toward outputs that satisfy desired properties. However, this has only been demonstrated for pretrained text-completion models on the relatively simple objective of reducing surface-level profanity in short continuations. A natural and practically important question is how well input embeddings can control aligned models, which produce an imbalanced bimodal refuse-or-comply output distribution rather than the smooth distribution characteristic of open-ended generation. We explore this in the context of safety, showing that input word embeddings can be optimized in a sub-lexical manner to minimize the semantic harmfulness of aligned model responses. Our approach uses zeroth-order gradient estimation of a black-box text-moderation API with respect to the input embeddings, and then applies gradient descent on these embeddings to minimize the harmfulness of the generated text. Experiments show that the proposed method neutralizes every safety-flagged response on standard safety benchmarks for four of the five evaluated models, and reduces the number of flagged responses on the fifth.

1 INTRODUCTION

Large language models deployed for user-facing tasks undergo safety training—typically reinforcement learning from human feedback (RLHF) (Ziegler et al., 2020; Bai et al., 2022a) and direct preference optimization (Rafailov et al., 2023)—to handle requests that violate their developers' safety policies. Despite this training, adversarial prompts can reliably bypass safety mechanisms and elicit harmful content from otherwise well-aligned models (Zou et al., 2023; Chao et al., 2025). These prompts wrap harmful requests in structural framing—roleplay scenarios, hypothetical narratives, obfuscating instructions—that exploits the gap between the model's safety training distribution and the space of possible inputs. The vulnerability is fundamental: alignment is learned from a finite set of examples, while the input space is effectively unbounded.

A growing body of test-time defenses addresses this gap without modifying the target model's weights. Prompt-level strategies (Xie et al., 2023; Wei et al., 2023; Zhou et al., 2024), randomized smoothing (Robey et al., 2025; Ji et al., 2025), guard models (Inan et al., 2023; Han et al., 2024), and activation-steering methods (Zhao et al., 2025; Gao et al., 2025) all reduce attack success rates, yet share a structural limitation: they either apply undirected perturbations to discrete tokens, require separately offline-trained auxiliary modules, or impose fixed interventions derived from a static calibration set. None directly optimizes a continuous safety objective at the model's input level.

A qualitatively different approach is to operate on the continuous representation the model actually conditions on: the prompt embedding matrix. Because the model reads this matrix rather than the tokens themselves, perturbations in embedding space can steer the output distribution without altering the human-readable input. Saglam & Kalogerias (2026) demonstrated this for detoxification, where iterative optimization over prompt embeddings reduces the toxicity of base pretrained models under a surface-level profanity classifier. Whether embedding-level control extends to the safety of aligned models is, however, an open question, and there is reason to expect that it may not transfer: aligned models produce a bimodal output distribution—refuse or comply—rather than the smooth continuation distribution over which toxicity reduction was demonstrated; the relevant notion of harm is semantic rather than lexical; and the prompts of interest are adversarially constructed to circumvent safety training, a different regime from naturally occurring toxic prompts.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We show that the transfer does hold. Given any prompt, we apply zeroth-order gradient estimation to approximate the gradient of a black-box text-moderation oracle with respect to the prompt embeddings, and descend along the estimated gradient to minimize the harmfulness of the generated text. The oracle—the OpenAI Moderation API (Markov et al., 2022; OpenAI, 2026)—scores completions across 13 harm categories; we take the maximum over these categories as our objective, so that a model response is considered safe only if it scores low along every axis of harm. The entire procedure operates at test time: it requires no retraining and no auxiliary models, and treats the language model as a gray box—only forward passes plus read access to its input word embeddings, which open-weight models expose and API-only commercial services do not—while treating the moderation oracle as a black box. The optimization is applied unconditionally, without knowledge of whether a prompt is harmful or benign, and does not force refusals; the model determines the form of its own response.

Experiments on five instruction-tuned models spanning 1B to 20B parameters, evaluated on two standard red-teaming benchmarks comprising adversarial jailbreak prompts and direct harmful queries, show that the method nearly eliminates every safety-flagged completion, typically converging within two gradient steps, and that the reductions carry over to two moderation judges that play no role in the optimization. On an adversarial benign control split, already-safe responses are left substantively intact—moderation scores do not increase for any model—confirming that the optimization does not impose a blanket refusal bias. A finding of broader interest is that the optimized embeddings decode back to the original prompt tokens in every case: the perturbation is sub-lexical, adjusting the continuous representation within the nearest-neighbor cell of each token, yet it reliably redirects the model toward safe outputs. This observation suggests that the conventional view of embeddings as inert token lookups understates the *control surface* available at the model's input layer. Our anonymized codebase (`https://anonymous.4open.science/r/instant-alignment`) includes a demo notebook illustrating the proposed method in action.

2 Related Work

We review inference-time methods for defending aligned language models without modifying the target model's weights. Approaches that improve safety through training (Bai et al., 2022b) or fine-tuning (Ziegler et al., 2020; Bai et al., 2022a; Rafailov et al., 2023) are complementary but outside our scope.

Prompt-level and input-transformation defenses. The earliest test-time defenses operate on the discrete text the model receives. System-prompt strategies wrap user queries with safety reminders (Xie et al., 2023) or prepend in-context refusal demonstrations (Wei et al., 2023), while more principled variants optimize defensive suffixes offline through minimax adversarial objectives (Zhou et al., 2024) or adversarial prompt training (Mo et al., 2024) and deploy the result as a fixed textual guard. A complementary direction randomizes the input to disrupt adversarial structure (Robey et al., 2025; Ji et al., 2025) or targets systematic token erasure (Kumar et al., 2024). All of these methods manipulate discrete tokens or strings, whereas ours operates in the continuous embedding space with directed gradient-based optimization rather than undirected perturbation or fixed textual controls.

Safety classifiers and detection. A second line of work interposes auxiliary models or analyses between the user and the LLM. Llama Guard (Inan et al., 2023) and WildGuard (Han et al., 2024) finetune language models into safety classifiers that filter harmful inputs or outputs at deployment. Other methods extract detection signals directly from the target model, such as perplexity-based filtering (Alon & Kamfonas, 2023), gradient-based methods for pattern identification (Xie et al., 2024) and refusal-loss thresholding (Hu et al., 2024), or linear probing for jailbreak detection (Zhang et al., 2025). These approaches share a detect-then-filter paradigm, whereas our method steers generation without predicting intent and therefore without a separate classification model or detection step.

Activation steering and decoding-time interventions. A third family intervenes on the model's internal computation during inference. Activation-steering methods add safety-relevant directions to hidden states—either as fixed vectors transferred from an aligned donor model (Wang et al., 2024), with per-input adaptive coefficients derived from a calibration set (Zhao et al., 2025), or by constraining activations to lie within empirically estimated safety boundaries (Gao et al., 2025). At

the decoding level, SafeDecoding (Xu et al., 2024) contrasts the output distributions of the frozen model and a lightweight safety expert to amplify safe tokens, and RAIN (Li et al., 2024) integrates self-evaluation and rewind mechanisms during generation. These approaches require hidden states or output logits, and activation steering derives its directions from a fixed calibration set that must transfer across prompts; ours instead treats the model as a gray box—assuming only that its input word embeddings are available—and, through zeroth-order optimization, adapts its intervention to the prompt at hand.

Embedding-space methods. The work most closely related to ours operates at the embedding level. RESTA (Hase et al., 2024) injects isotropic Gaussian noise into token embeddings and aggregates perturbed copies by majority vote—randomized smoothing in embedding space, with no directed optimization and hence no way to target a specific safety objective. DRO (Zheng et al., 2024) optimizes continuous soft-prompt embeddings toward higher-refusal regions, achieving directed control but through an offline training phase that yields a fixed prompt applied uniformly at deployment. We instead optimize the word embeddings of the prompt at hand against a semantic moderation objective, fully at test time, without offline training or additional data. The algorithmic foundation is that of Saglam & Kalogerias (2026), which showed that iterative optimization over prompt embeddings can reduce the toxicity of base pretrained (non–instruction-tuned) models in open-ended generation under a surface-level profanity scorer; we apply it to a fundamentally different regime—safety-aligned chat models, a multi-category semantic moderation oracle, and adversarial prompts designed to circumvent safety training through semantic manipulation—rather than to naturally occurring short, toxic continuations.

3 Problem Formulation

Modern large language models intended for user-facing deployment are *aligned*: after next-token pre-training on web-scale corpora, a post-training stage—supervised instruction fine-tuning (Ouyang et al., 2022), RLHF (Ziegler et al., 2020; Bai et al., 2022a), and direct preference optimization (Rafailov et al., 2023)—makes their outputs helpful, honest, and harmless, so that they follow natural-language instructions across a wide range of tasks while refusing requests that violate the safety policies of their developers.

Let f denote an aligned autoregressive language model based on the transformer architecture (Vaswani et al., 2017). A prompt is a string $\mathcal{P}$ that a tokenizer maps to a sequence of token indices $t_{1:T} = (t_1, \ldots, t_T)$, $t_i \in \mathcal{V}$, drawn from a finite vocabulary $\mathcal{V}$. Each token t_i is associated with a fixed d-dimensional embedding vector $x_i \in \mathbb{R}^d$, so the full prompt is represented by the embedding matrix

$$X = \begin{bmatrix} x_1^\top & x_2^\top & \cdots & x_T^\top \end{bmatrix}^\top \in \mathbb{R}^{T \times d}.$$

Given a prefix $(t_1, \ldots, t_{i-1})$, the model defines a distribution over the next token, $f(t_i \mid t_1, \ldots, t_{i-1}) \equiv p_f(t_i \mid t_{<i})$. We denote by $y = f(X)$ the response obtained by sequentially sampling from this distribution when f is conditioned on the prompt $\mathcal{P}$—equivalently, on its token sequence $t_{1:T}$ and embedding matrix X.

3.1 Measuring Harmful Generations

Despite alignment, models can still produce harmful content, and it may carry no surface-level markers: a response can be polite and profanity-free yet provide detailed instructions for illicit activity, so detecting such failures requires evaluating the *generated response*, not the prompt.

We measure the harmfulness of $y = f(X)$ using the OpenAI Moderation API (Markov et al., 2022; OpenAI, 2026), a black-box content classifier deployed so that hosts and users can take corrective action when potentially harmful content is detected. Because the same classifier gates responses in deployment, lowering this score is not a proxy for the operational safety criterion but a direct optimization of it. The API returns a continuous score in $[0, 1]$ for each of 13 categories, spanning harassment, hate, illicit activity, self-harm, sexual content, and violence, each with one or more finer-grained sub-categories. Each category additionally carries a binary "flagged" indicator; the threshold mapping scores to flags is internal to the API and not exposed. We abstract this oracle as

$$h : \mathcal{Y} \to [0, 1], \qquad h(y) = \max_c \, s_c(y), \tag{1}$$

where $\mathcal{Y}$ is the space of output strings and $s_c(y)$ is the score in category c. Taking the maximum ensures that a response is considered safe only if it scores low across *every* harm category—minimizing h therefore always targets the worst-offending category.

3.2 Objective

Our goal is to intervene at test time—without modifying the target model or learning any additional parameters—so that the harmfulness of the model's output is minimized for any prompt. The method is applied unconditionally, with no access to whether a prompt is benign or harmful, and it does not force refusals: the model determines the form of its own response. Concretely, we optimize the prompt embedding matrix X to reduce the harmfulness of the resulting response $\Phi(X) := h\left(f(X)\right)$, treating the moderation oracle h as a black box and the language model f as a gray box, of which we use only forward passes and the input embeddings. Because backpropagating through f is prohibitively expensive at the scale of modern LLMs, we restrict ourselves to forward passes alone.

Since minimizing h targets the worst-offending category at every step, the mechanism steers the model toward safe responses precisely when caution is warranted and leaves already-safe responses substantively intact.

4 Methodology

Overview. The composite objective $\Phi(X) = h\left(f(X)\right)$ exposes no usable gradient; f is an autoregressive generator and h is a black-box cloud API. We therefore substitute a stochastic surrogate: at iteration k we form an estimator g_k—derived in the next subsection from a Gaussian-smoothed reformulation of Φ—and apply the update

$$X_{k+1} \leftarrow X_k - \eta\, g_k,$$

which approximately decreases Φ and thereby drives the resulting response toward lower-harmfulness regions of embedding space, typically converging within a handful of iterations (in practice, $K < 6$).

4.1 Zeroth-Order Gradient Estimator

Unless noted otherwise, the development in this subsection follows the foundation of Nesterov & Spokoiny (2017). Our objective is to estimate $\nabla_X \Phi(X)$ given only forward (zeroth-order) access to Φ.

We begin by perturbing X with i.i.d. Gaussian noise applied independently to each token embedding:

$$X + \mu U, \qquad U = \begin{bmatrix} u_1^\top & u_2^\top & \cdots & u_T^\top \end{bmatrix}^\top \in \mathbb{R}^{T\times d}, \qquad u_1, \ldots, u_T \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_d).$$

Since U has the same shape as X, every token is perturbed in its own direction, so the estimator captures token-specific sensitivities of Φ and adjusts the prompt at a fine-grained level. Averaging Φ over these perturbations defines the *Gaussian-smoothed* objective

$$\Phi_\mu(X) = \mathbb{E}_U[\Phi(X + \mu U)]\,.$$

Smoothing has two convenient consequences. First, whenever Φ is Lipschitz-continuous—i.e., $|\Phi(X) - \Phi(Y)| \le L\|X - Y\|$—the smoothed surrogate Φ_μ is differentiable for every $\mu > 0$ while remaining Lipschitz itself. Second, replacing Φ with Φ_μ incurs only a controlled approximation error of order $\mathcal{O}(\mu\sqrt{Td})$.

A key result of Nesterov & Spokoiny (2017) expresses the gradient of the smoothed objective in closed form as a directional finite difference:

$$\nabla\Phi_\mu(X) = \mathbb{E}_U\left[\frac{\Phi(X + \mu U) - \Phi(X)}{\mu}\, U\right]. \tag{2}$$

This identity requires only Lipschitz continuity of Φ; differentiability of Φ itself is not assumed. Intuitively, $\nabla\Phi_\mu(X)$ is a weighted average of perturbation directions, each scaled by the relative change Φ exhibits along it. The subtracted baseline $\mathbb{E}_U[\Phi(X)\,U/\mu]$ leaves the expectation unchanged, since $\mathbb{E}_U[U] = 0$ and $\Phi(X)$ is independent of U, yet it reduces the variance of any Monte Carlo approximation.

To relate $\nabla\Phi_\mu$ back to $\nabla\Phi$, one additional regularity assumption is useful. If Φ is also Lipschitz-smooth—that is, if its gradient is itself L-Lipschitz—then Φ_μ inherits the same smoothness, and the gap between the two gradients is bounded:

$$\|\nabla\Phi_\mu(X) - \nabla\Phi(X)\| \le C\,\mu L\,(Td)^{3/2},$$

for a constant C. Lipschitz continuity stabilizes the finite difference, while smoothness controls how rapidly $\nabla\Phi_\mu$ itself can vary. Together they make $\nabla\Phi_\mu(X_k)$ a controlled approximation of $\nabla\Phi(X_k)$, with μ governing the residual bias. This smoothing analysis is motivational, not load-bearing. For a black-box $\Phi = h \circ f$ the regularity constants cannot be verified and are not assumed to hold; since f is discrete autoregressive sampling, a small embedding perturbation can flip a token and move the moderation score sharply, so Φ is closer to a near-discontinuous map than to one of bounded slope. The analysis shows only that Gaussian smoothing renders such an objective optimizable, by supplying a differentiable surrogate Φ_μ with a usable descent direction; that the direction is effective is established empirically in Section 5.

We should note that the maximum in (1) is not differentiable, so Φ may be Lipschitz-continuous but not Lipschitz-smooth, and therefore the bound above does not apply as stated. A smooth surrogate—for instance, a softmax-weighted average $\sum_c w_c\, s_c$ with weights $w_c \propto \exp(\beta\, s_c)$—would restore the required regularity. In practice, however, we omit this substitution, because the highest-scoring harm category does not typically switch along the optimization path. We measured this directly over every path in the main evaluation, and no evaluated iterate changed the top-scoring category (0 of 39,150 model–prompt–trial paths). The maximum therefore reduces to a single smooth category score along each path.

Finally, we replace the expectation in (2) with a Monte Carlo average over N independent perturbations, yielding the practical estimator used throughout this work:

$$\boxed{g_k = \frac{1}{N}\sum_{i=1}^{N} \frac{\Phi(X_k + \mu U_i) - \Phi(X_k)}{\mu}\, U_i \;\approx\; \nabla\Phi_\mu(X_k).}$$

Its variance grows with the ambient dimension Td and shrinks with N, while μ controls a bias–variance tradeoff. Larger μ smooths Φ more aggressively and stabilizes the estimate, whereas smaller μ sharpens the approximation at the cost of noisier evaluations. In the joint limit $\mu \to 0$ and $N \to \infty$, and under the regularity conditions discussed above, g_k recovers the true gradient direction of Φ; in practice, moderate values of both suffice for stable and effective test-time updates.

The loop is also stochastic: responses are sampled at temperature 0.1, so each evaluation of Φ is a draw. Eq. (2) requires Lipschitz continuity, not determinism, so g_k descends the smoothed *expected* objective and decoding noise enters only as estimator variance, curbed by the baseline term above, by the gradient normalization of Section 4.2, and by the best-iterate return of Algorithm 1; at temperature 0.1 sampling concentrates on the modal completion, and the sub-lexical updates (Section 6) move within a token's neighborhood rather than across the token-flip boundaries where the refuse-or-comply discontinuity lives. Appendix E quantifies the resulting cross-trial variability.

4.2 Regularization

The zeroth-order descent must be regularized to keep the updated embeddings from drifting into regions of the latent space that the model has not encountered during training. We use two deliberately simple mechanisms and introduce no auxiliary modules—no momentum, no learned projection, no external reward shaping—so that any improvement we observe is attributable to the test-time intervention alone.

Gradient normalization. A single fixed learning rate applied to the raw estimate can produce erratic updates—some too timid to make progress, others large enough to push embeddings off-manifold—so we normalize the estimate before each step, replacing g_k with $g_k/\|g_k\|_2$ in the update rule; the step magnitude is then governed entirely by η, a single learning rate that behaves consistently across prompts.

Cosine similarity. To preserve fidelity to the original prompt, we additionally constrain the updated embedding to remain close to X_0 in direction, since cosine similarity measures alignment independently of magnitude. At the end of each iteration, if the current embedding falls below a cosine

similarity threshold κ relative to X_0, we project it back onto the boundary of the corresponding cosine ball before proceeding.

We refer to the resulting end-to-end test-time defense as **T**est-Time **S**afety **A**lignment (TSA); pseudocode is provided in Algorithm 1 (Appendix A).

5 EXPERIMENTS

Models. Experiments use five aligned, instruction-tuned models from different developers and scales: Gemma 3 (1B) (Kamath et al., 2025), Phi-3.5-mini (4B) (Abdin et al., 2024), Llama 3.1 (8B) (Grattafiori et al., 2024), Qwen3 (14B) (Yang et al., 2025), and GPT-OSS (20B) (OpenAI et al., 2025).

Benchmarks. We select red-teaming benchmarks whose prompts vary along two independent axes: *construction* (direct phrasing vs. adversarial framing) and *intent* (benign vs. harmful). Adversarially framed harmful prompts are the hardest condition, since their framing is designed to elicit harmful content from aligned models.

For adversarial prompts, we use the evaluation set of the WildJailbreak benchmark (Jiang et al., 2024b), which contains 210 adversarial benign prompts (as a sanity check) and 2,000 adversarial harmful prompts. We also include HarmBench (Mazeika et al., 2024), which provides 400 direct harmful queries; details about the benchmarks are provided in Appendix B.1.

Generation. Models are queried over three trials at temperature 0.1, generating 128 tokens per response. A budget of 128 tokens is sufficient because the moderation oracle is saturated by any harmful span within the window. A low temperature concentrates sampling on the modal completion a typical user would receive, a realistic regime in which to measure harmfulness.

Metrics. We report the mean, median, and maximum of the highest-category score, averaged over the three completions, together with the number of *flagged* (i.e., classified as dangerous) prompts. The classifier remains a black box but is believed to balance precision and recall (Markov et al., 2022). The optimization runs for 10 iterations with early stopping triggered when the top-scoring category falls below 0.1, the level at which the developers report moderation scores to become actionable (Abubakar, 2025). Cross-trial variability is also reported in Appendix E as the mean over prompts of each prompt's sample standard deviation across the three trials; the gains reported exceed it for most model and split.

Baselines. We compare against three inference-time defenses: SmoothLLM (Robey et al., 2025), AdaSteer (Zhao et al., 2025), and RESTA (Hase et al., 2024). SmoothLLM perturbs copies of the input at the character level and returns the majority-safe completion; AdaSteer steers hidden-state activations along safety-relevant directions derived from a small calibration set; and RESTA injects Gaussian noise into the prompt embeddings and aggregates over the perturbed completions. AdaSteer and RESTA are compared on Llama 3.1-8B-Instruct since their published configurations apply directly, avoiding the confound of an independent per-model search over their method-specific calibration parameters.

To our knowledge, no other test-time defense directly optimizes a continuous safety objective; these baselines are the closest inference-time alternatives. Further details are provided in Appendix B.

5.1 MAIN RESULTS

Results on WildJailbreak are reported in Table 1. Complete results, including HarmBench, are provided in Appendix C. A useful starting point for interpretation is what the oracle actually captures. The moderation score quantifies the harmfulness of the generated text, not whether the model complied with or refused the request. A refusal that reproduces the harmful query scores high because the harmful content is present regardless of intent; a substantive but harmless answer scores low. Post-TSA responses are therefore not forced into blanket refusals but remain free to engage substantively with the prompt, with the guarantee that the content they produce will be harmless.

What varies across models is the number of iterations to convergence, not the outcome. Llama 3.1 (8B) converges in a single descent step, while the larger Qwen3 (14B) requires roughly two, so parameter count alone does not set the difficulty. These iteration averages are conditioned on

Table 1: Text moderation results for five instruction-tuned models (temperature 0.1, three trials per prompt). *Flagged* is the number of prompts whose responses the OpenAI Moderation API classifies as harmful; *Time* is the average per-prompt optimization time and *Net* excludes oracle latency. Within each model–benchmark pair, the best value per metric is in **bold**. Held-out judge scores are in Section 5.1 and Appendix E.

	Model	Flagged ↓	Score ↓ Mean	Score ↓ Med.	Score ↓ Max	# Iterations	Time (s)	Net (s)
WildJailbreak (Adv. Harmful) $n = 2{,}000$	GPT-OSS-20B	17	0.023	0.004	0.849	–	–	–
	+ TSA	**0**	**0.009**	**0.003**	**0.099**	1.2	12.1	10.2
	Qwen3-14B	165	0.097	0.028	0.944	–	–	–
	+ SmoothLLM	148	0.095	0.027	0.952	–	–	–
	+ TSA	**0**	**0.023**	**0.016**	**0.100**	2.1	28.3	25.4
	Llama 3.1-8B	218	0.106	0.020	0.951	–	–	–
	+ SmoothLLM	122	0.068	0.006	0.954	–	–	–
	+ AdaSteer	85	0.057	0.016	0.947	–	–	–
	+ RESTA	199	0.098	0.010	0.944	–	–	–
	+ TSA	**0**	**0.012**	**0.002**	**0.098**	1.0	12.2	10.5
	Phi-3.5-4B	126	0.081	0.020	0.921	–	–	–
	+ SmoothLLM	123	0.076	0.019	1.000	–	–	–
	+ TSA	**0**	**0.019**	**0.008**	**0.099**	1.8	12.4	9.8
	Gemma 3-1B	278	0.137	0.063	0.946	–	–	–
	+ SmoothLLM	225	0.119	0.039	0.945	–	–	–
	+ TSA	**59**	**0.065**	**0.031**	**0.844**	5.7	21.1	13.7
WildJailbreak (Adv. Benign) $n = 210$	GPT-OSS-20B	**0**	0.011	**0.003**	0.270	–	–	–
	+ TSA	**0**	**0.009**	**0.003**	**0.097**	1.0	8.0	6.3
	Qwen3-14B	2	0.014	**0.001**	0.253	–	–	–
	+ SmoothLLM	**0**	0.014	0.002	0.254	–	–	–
	+ TSA	**0**	**0.010**	**0.001**	**0.098**	2.1	29.2	26.3
	Llama 3.1-8B	1	0.017	0.003	0.356	–	–	–
	+ SmoothLLM	2	0.018	0.004	0.358	–	–	–
	+ AdaSteer	3	0.022	**0.002**	0.359	–	–	–
	+ RESTA	2	0.013	0.004	0.351	–	–	–
	+ TSA	**0**	**0.009**	**0.002**	**0.098**	1.0	12.6	10.9
	Phi-3.5-4B	**0**	0.015	**0.001**	0.205	–	–	–
	+ SmoothLLM	2	0.016	**0.001**	0.565	–	–	–
	+ TSA	**0**	**0.009**	**0.001**	**0.098**	1.3	10.2	8.2
	Gemma 3-1B	3	0.023	0.006	0.354	–	–	–
	+ SmoothLLM	1	0.027	0.010	0.286	–	–	–
	+ TSA	**0**	**0.019**	**0.004**	**0.199**	3.1	12.4	8.4

entering the optimization loop. On Llama 3.1 and the adversarial-harmful split, only 218 of 2,000 prompts trigger optimization at all (exactly its undefended flagged count), so the amortized per-prompt cost on a mixed, realistic workload is substantially lower than the timing figures suggest. Every prompt costs one generation and one moderation call, and only a prompt whose initial score reaches 0.1 pays the per-optimization time of Table 1; with r the fraction that does, the amortized cost is that fixed cost plus r times the optimization time. For Llama 3.1 on the adversarial harmful split, $r = 218/2{,}000 = 0.109$, so $0.109 \times 12.2 \approx 1.3$ s per prompt against the 12.2 s a prompt that enters the loop pays; on the adversarial benign split one prompt in 210 enters ($r \approx 0.005$), for roughly 0.06 s. The harmful split, in which every prompt is a jailbreak attempt, is therefore a conservative upper bound on r; a realistic workload sits toward the benign end.

The sole exception is Gemma 3, for which the optimization leaves 59 responses flagged (from 278). Within the five models the two explanations one might offer are confounded—Gemma is at once the smallest ($d = 1152$) and, by the moderation objective, the least safety-aligned—so we tested them with a controlled pair (Appendix F). TSA clears every flag on Qwen3-0.6B, a post-trained

model smaller than Gemma on both counts ($d = 1024$; 470 flagged undefended), in a single average iteration, and on a Qwen3-14B variant whose refusal behavior was removed by abliteration (527 flagged undefended) at the cost of about one extra step (3.2 versus 2.1 iterations). Neither size nor base alignment explains Gemma; difficulty appears in the iteration count, and Gemma's residuals are the prompts that do not cross the 0.1 threshold within the 10-iteration budget at its 5.7 average iterations. A manual review of all residual flagged outputs on both benchmarks found coherent, non-evasive responses with no sign of reward hacking and harmfulness reduced relative to the base response.

The three baselines span text-level (SmoothLLM), activation-level (AdaSteer), and embedding-level (RESTA) interventions. The decisive difference is that *zeroth-order optimization ties the harmfulness objective directly to the model's inputs*, rendering the intervention agnostic to the attack mechanism. The baselines instead rely on the brittleness of adversarial suffixes (SmoothLLM, RESTA) or on steering directions derived from a fixed calibration distribution (AdaSteer). When these assumptions are violated—on HarmBench the queries carry no adversarial suffix at all—the baselines falter: AdaSteer and RESTA produce more flagged completions than the undefended model.

The do-no-harm check on the adversarial benign split passes cleanly. TSA is applied unconditionally, so no refusal bias is imposed, and early stopping prevents detrimental changes from being applied to the embeddings; together these keep the model's behavior on benign prompts intact. Since a lower score does not by itself show that a response remains useful, we also read every post-TSA response to the 210 adversarial benign prompts across all five models and all three trials; we confirmed that each still addresses the legitimate intent of its request rather than merely avoiding harmful content. On the runtime side, oracle latency rather than inference is a binding constraint (roughly 20% of the total per-prompt runtime), so a local moderation classifier or batched API requests would be an impactful deployment-time optimization.

Held-out judges and cross-trial variability. The score being minimized is the gate that decides whether a response is served, so driving it below threshold is the operational objective and not a stand-in for one (Section 3). The search that achieves it is sub-lexical and model-mediated (Section 6), so the responses it can reach are ones the frozen model itself produces, not text written to evade a classifier. Whether the reductions nonetheless transfer beyond the optimized channel is an empirical question. We therefore re-scored the identical responses from Table 1 with two judges that were not accessed during the optimization: the HarmBench classifier (Mazeika et al., 2024) and Llama Guard 3-8B (Inan et al., 2023) (Appendix E, with cross-trial standard deviations for every cell). Under both judges every base-to-TSA comparison on the two harmful splits decreases (16 of 16), TSA scores below the comparable baseline in 20 of 24 comparisons (the exceptions, all under the HarmBench classifier, are AdaSteer on Llama 3.1 and SmoothLLM on HarmBench for Gemma 3 and Phi-3.5), and on the adversarial benign split Llama Guard scores do not increase for any model. Transfer to two never-optimized judges is evidence against gains confined to the optimized channel, though no automatic judge is a ground truth for safety, which is why the manual readings reported above complement them. Cross-trial standard deviations are still very low (at most 0.014), and under the optimized oracle every base-to-TSA reduction exceeds the per-prompt variability.

Sensitivity analysis. Appendix D varies the Monte Carlo sample count N, the perturbation scale μ, and the early stopping threshold on Gemma 3-1B and Llama 3.1-8B. Llama is largely invariant to μ and N, and $N = 8$ suffices across embedding dimensions from 1152 to 5120 despite estimator variance growing with Td; Gemma, which converges more slowly, is more sensitive. Neither model shows a monotonic trend in μ; only the threshold has a monotonic effect, and at 0.01 Llama reaches a mean score of 0.001 in fewer than two iterations.

6 Discussion

How can a sub-lexical intervention be so effective? The intervention leaves the discrete token sequence intact while moving the sub-lexical representation the transformer conditions on. Gradient normalization keeps the cumulative update small in magnitude, and the vocabulary—on the order of 10^5 points in $\mathbb{R}^d$ for $d \geq 1152$—occupies a vanishingly sparse subset of the embedding space, so each token's Voronoi cell is large relative to the step sizes involved and the perturbation stays within the cell it starts in. What reaches the network is therefore an input that shares its token identity with

the original but not its sub-lexical representation, and autoregressive generation is sensitive to exactly that variation: the difference propagates through the layers into materially different hidden states and output distributions.

Can the optimization be reversed to produce jailbreaks? Not in any sustained way, and the asymmetry is a consequence of safety training. The moderation scores of aligned models' default responses are typically already low, so the finite-difference numerator collapses into estimator noise and the ascent direction carries no usable signal. Safety training further shapes a broad refusal basin in which most local perturbations continue to refuse, while the narrow compliance regions exploited by known jailbreaks lie beyond the reach of small isotropic Gaussian probes. The harder case is a prompt that already elicits compliance, where the numerator does not collapse; we measured it on Qwen3-14B over all 2,000 adversarial harmful prompts (Appendix G). Ten sign-reversed updates on the 649 prompts with an initial score of at least 0.1, retaining the highest-scoring response along each path, raise the full-split mean from 0.097 to 0.134 ± 0.078 and the flagged count from 165 to 310 of 2,000, still below the 0.177 and 527 of the abliterated Qwen3-14B whose refusal behavior has been removed.

Practical profile. For four of the five models evaluated, the method eliminates every safety-flagged response on all benchmarks while requiring only three ingredients: access to the input word embeddings, a few forward passes, and a black-box moderation oracle. It needs no retraining, no auxiliary classifier, no calibration data, and no intent filter. The intervention is applied unconditionally, yet it does not force refusals. Its per-prompt overhead is a small number of additional forward passes—typically one or two gradient steps—adding roughly 15–20 s of wall-clock time. Because inference engines such as vLLM (Kwon et al., 2023) batch the $N + 1$ perturbations into a single call and the oracle is a freely available public API, the method serves as a drop-in safety layer.

Limitations. The optimization loop introduces per-prompt overhead, typically seconds given convergence in one or two gradient steps, of which the oracle's network round-trip is the only irreducible component. The method also requires access to the input word embeddings, a condition satisfied by open-weight models but not by API-only services. Finally, the iterations needed for convergence vary across models. Gemma 3 needs the most, and its hardest prompts remain above threshold within the iteration budget. The controlled comparison of Appendix F rules out a small embedding dimension and parameter count as the cause, indicating that convergence depends on the model as well as on the optimizer. Such dependence is inherent to a training-free method, since the search moves within behavior the frozen model can already produce. The method is accordingly designed to amplify the safety alignment a model already carries, not to replace it.

7 Conclusion

Embedding-level behavioral control, previously demonstrated for reducing surface-level profanity in open-ended completion models, extends to the harder regime of safeguarding aligned models against prompts that elicit harmful content. By estimating the gradient of a black-box moderation oracle with respect to the prompt embeddings and descending along it, the method minimizes the semantic harmfulness of generated text without retraining, auxiliary modules, or knowledge of prompt intent, adapting to each input through a handful of gradient steps, and it eliminates nearly all safety-flagged completions across five models and standard red-teaming benchmarks while leaving already-safe responses intact.

Beyond the immediate safety application, our results reveal the embedding layer as a richer control surface than its role as a discrete token lookup suggests: the optimized embeddings decode back to the original prompt tokens in every case, yet the output distribution shifts reliably toward safe behavior. Whether sub-lexical embedding optimization generalizes beyond safety—to other behavioral objectives, model families, or modalities—is a natural direction for future work.

AI use statement

Generative AI tools were used only to edit the manuscript for readability (grammar and phrasing) and to edit code that the authors had written themselves (restructuring and optimization). They were not

used to generate or process data, develop the method or theory, design or run experiments, analyze or interpret results, or draft scientific content. The authors reviewed every AI-assisted edit and take full responsibility for the final content.

Ethics statement

This work develops a defense that lowers the harmfulness of language-model responses at test time; it neither trains nor releases a model with new capabilities. The evaluation uses public red-teaming benchmarks of harmful and adversarially framed prompts, and we report aggregate moderation statistics only, reproducing no harmful prompt or completion. Whether the same optimizer can be run in reverse to produce jailbreaks is examined empirically (Section 6 and Appendix G); on prompts that already elicit compliance, reverse-directed search raises the worst-case moderation score, yet the result stays below what an abliterated model produces with no optimization at all. Because the method queries a third-party moderation API, generated text leaves the local machine; deployments that handle sensitive content can substitute a local classifier, since the method treats the scorer as a black box. No human subjects were involved, and the manual reviews reported in Section 5 were carried out by the authors.

Reproducibility statement

Algorithm 1 (Appendix A) specifies the full procedure, and the anonymized repository linked in Section 1 contains the implementation, a demonstration notebook, and the base, TSA, and baseline responses behind the reported results. Appendix B lists the per-model hyperparameters and the sweep that selected them (Table 3), benchmark statistics (Table 2), baseline configurations, the inference backend, and the hardware; all benchmarks and the moderation oracle are publicly available. The held-out judges and the cross-trial variability statistics are described in Appendix E, and the two supplementary experiments (Appendices F and G) follow the protocol of Section 5, with every deviation stated there.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL https://arxiv.org/abs/2404.14219.

Abubakar. Moderation boundaries with openai api. https://dev.to/thatechmaestro/moderation-boundaries-with-openai-api-333g, May 2025. DEV Community. Accessed: 2026-04-10.

Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity, 2023. URL https://arxiv.org/abs/2308.14132.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a. URL https://arxiv.org/abs/2204.05862.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022b. URL https://arxiv.org/abs/2212.08073.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking Black Box Large Language Models in Twenty Queries . In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42, Los Alamitos, CA, USA, April 2025. IEEE Computer Society. doi: 10.1109/SaTML64287.2025.00010. URL https://doi.ieeecomputersociety.org/10.1109/SaTML64287.2025.00010.

Lang Gao, Jiahui Geng, Xiangliang Zhang, Preslav Nakov, and Xiuying Chen. Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25378–25398, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1233. URL https://aclanthology.org/2025.acl-long.1233/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley

Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu,

Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=Ich4tv4202.

Ryo Hase, Md Rafi Ur Rashid, Ashley Lewis, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. Smoothed embeddings for robust language models. In *Neurips Safe Generative AI Workshop 2024*, 2024. URL https://openreview.net/forum?id=GkBRng3Hl9.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=vI1WqFn15v.

huihui.ai. Huihui-qwen3-14b-abliterated-v2. https://huggingface.co/huihui-ai/Huihui-Qwen3-14B-abliterated-v2, 2025. Hugging Face model card. Accessed: 2026-07-28.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations, 2023. URL https://arxiv.org/abs/2312.06674.

Jiabao Ji, Bairu Hou, Alexander Robey, George J. Pappas, Hamed Hassani, Yang Zhang, Eric Wong, and Shiyu Chang. Defending large language models against jailbreak attacks via semantic smoothing. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 7–40, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-299-2. doi: 10.18653/v1/2025.ijcnlp-short.2. URL https://aclanthology.org/2025.ijcnlp-short.2/.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024a. URL https://arxiv.org/abs/2401.04088.

Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=n5R6TvBVcX.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. Certifying LLM safety against adversarial prompting. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=9Ik05cycLq.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. RAIN: Your language models can align themselves without finetuning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pETSfWMUzy.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection. *arXiv preprint arXiv:2208.03274*, 2022.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: a standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. Fight back against jailbreaking via prompt adversarial tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=nRdST1qifJ.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017. doi: 10.1007/s10208-015-9296-2. URL https://doi.org/10.1007/s10208-015-9296-2.

OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

OpenAI. Moderation. https://developers.openai.com/api/docs/guides/moderation, 2026. Accessed: 2026-04-09.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=laPAh2hRFC.

Baturay Saglam and Dionysis Kalogerias. Test-time detoxification without training or learning anything, 2026.

Baturay Saglam, Paul Kassianik, Blaine Nelson, Sajana Weerawardhena, Yaron Singer, and Amin Karbasi. Large language models encode semantics and alignment in linearly separable representations. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the*

14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pp. 2282–2303, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. doi: 10.18653/v1/2025.ijcnlp-long.124. URL https://aclanthology.org/2025.ijcnlp-long.124/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Mozhi Zhang, Ke Ren, Botian Jiang, and Xipeng Qiu. InferAligner: Inference-time alignment for harmlessness through cross-model guidance. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10460–10479, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.585. URL https://aclanthology.org/2024.emnlp-main.585/.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and Guard Aligned Language Models with Only Few In-Context Demonstrations . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1–12, January 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2026.3660147. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2026.3660147.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending ChatGPT against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496, December 2023.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. GradSafe: Detecting jailbreak prompts for LLMs via safety-critical gradient analysis. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 507–518, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.30. URL https://aclanthology.org/2024.acl-long.30/.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5587–5605, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.303. URL https://aclanthology.org/2024.acl-long.303/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shenyi Zhang, Yuchen Zhai, Keyan Guo, Hongxin Hu, Shengnan Guo, Zheng Fang, Lingchen Zhao, Chao Shen, Cong Wang, and Qian Wang. Jbshield: defending large language models from jailbreak attacks through activated concept analysis and manipulation. In *Proceedings of the 34th USENIX Conference on Security Symposium*, SEC '25, USA, 2025. USENIX Association. ISBN 978-1-939133-52-6.

Weixiang Zhao, Jiahe Guo, Yulin Hu, Yang Deng, An Zhang, Xingyu Sui, Xinyang Han, Yanyan Zhao, Bing Qin, Tat-Seng Chua, and Ting Liu. AdaSteer: Your aligned LLM is inherently an adaptive jailbreak defender. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24559–24577, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1248. URL https://aclanthology.org/2025.emnlp-main.1248/.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. On prompt-driven safeguarding for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61593–61613. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zheng24n.html.

Andy Zhou, Bo Li, and Haohan Wang. Robust prompt optimization for defending language models against jailbreaking attacks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=jXs6Cvpe7k.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

A Algorithm Pseudocode

Algorithm 1 provides the complete pseudocode for the test-time safety alignment procedure described in Section 4. When early stopping does not trigger, each iteration requires $N + 1$ forward passes through the language model and $N + 1$ moderation oracle queries—one base-point evaluation and N perturbed evaluations—all independent and batchable into a single inference call. When the base-point score already falls below the early stopping threshold, the gradient computation is skipped entirely, reducing that iteration to a single forward pass. Π_κ denotes projection onto $\{Z : \cos(Z, X_0) \geq \kappa\}$.

Algorithm 1 Test-Time Safety Alignment (TSA)

Require: Prompt $\mathcal{P}$; language model f; moderation oracle h

$\mathcal{P}_{\text{chat}} \leftarrow \textsc{ChatTemplate}(\mathcal{P})$ ▷ Format with role markers
$t_{1:T} \leftarrow \textsc{Tokenize}(\mathcal{P}_{\text{chat}})$ ▷ Token sequence $(t_1, \ldots, t_T)$
$X_0 \leftarrow \textsc{Embed}(t_{1:T})$ ▷ Word embeddings $X_0 \in \mathbb{R}^{T \times d}$

$\Phi^* \leftarrow \infty;\ X^* \leftarrow X_0$ ▷ Track best embedding

for $k = 0, \ldots, K-1$ **do**
  $y_k \leftarrow f(X_k)$ ▷ Generate response
  $\Phi_k \leftarrow h(y_k)$ ▷ Moderation score $\Phi_k = h(f(X_k))$

  if $\Phi_k < \Phi^*$ **then** ▷ Update best
    $\Phi^* \leftarrow \Phi_k;\ X^* \leftarrow X_k$
  end if

  if $\Phi_k < 0.1$ **then** ▷ Early stopping
    break
  end if

  for $i = 1, \ldots, N$ **do** ▷ N perturbations
    $U_i \sim \mathcal{N}(\mathbf{0},\ I) \in \mathbb{R}^{T \times d}$
    $y_i \leftarrow f(X_k + \mu\, U_i)$
    $\Phi_i \leftarrow h(y_i)$
  end for

  $g_k \leftarrow \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} \dfrac{\Phi_i - \Phi_k}{\mu}\, U_i$ ▷ Zeroth-order gradient estimate
  $g_k \leftarrow g_k \,/\, \|g_k\|_2$ ▷ Gradient normalization
  $X_{k+1} \leftarrow X_k - \eta\, g_k$ ▷ Descent step

  if $\cos(X_{k+1},\ X_0) < \kappa$ **then** ▷ Cosine similarity constraint
    $X_{k+1} \leftarrow \Pi_\kappa(X_{k+1},\ X_0)$
  end if
end for

return X^*

B EXPERIMENTAL DETAILS

Table 2: Sample counts and token-level statistics for each safety-alignment dataset. No preprocessing is applied.

		# Tokens			
Dataset	# Samples	Max	Min	Mean	Median
WildJailbreak (adversarial benign)	210	601	14	191.15	157
WildJailbreak (adversarial harmful)	2,000	614	18	141.97	126
HarmBench (direct harmful)	400	39	6	17.86	17

B.1 BENCHMARKS

Detailed statistics covering the number of samples and token-level properties (minimum, maximum, mean, and median) are provided in Table 2.

The selected benchmarks were originally designed for red-teaming evaluation; we repurpose them as sources of prompts that span a range of conditions under which aligned models may produce harmful content. WildJailbreak was additionally proposed for safety training and validation tasks. We also considered its sister dataset, WildGuardMix (Han et al., 2024), from the same developer, which is designed primarily for *moderation*—that is, teaching models to refuse harmful queries appropriately. Since WildGuardMix is derived from WildJailbreak, we proceeded with the latter.

Direct Harmful (HarmBench). The authors used GPT-4 (OpenAI, 2024) to distill the acceptable use policies of major AI companies into a set of guidelines, and then manually designed behaviors that violate laws or norms while filtering out "dual-intent" behaviors with plausible benign uses. They prioritized *differential harm*—behaviors that LLMs can enable beyond what a person could readily obtain through a search engine—particularly for contextual and multimodal cases.

Harmful Injections (WildJailbreak). Harmful requests framed adversarially as prompt injections in more convoluted and stealthy forms. The authors' WildTeaming framework was applied to rewrite direct harmful queries using 2–7 randomly sampled in-the-wild jailbreak tactics, with Mixtral-8×7B (Jiang et al., 2024a) and GPT-4 serving as the rewriters. After filtering out low-risk and off-topic prompts, each adversarial prompt was paired with the refusal response of its direct counterpart, yielding 82,728 items.

Benign Injections (WildJailbreak). Prompt injections that resemble jailbreaks but carry no harmful intent. Jiang et al. (2024b) generated 78,710 such prompts with WildTeaming from direct benign queries, using GPT-3.5 (Brown et al., 2020) to produce the direct prompts.

B.2 HYPERPARAMETERS

We swept μ and the learning rate η on a small representative subset of 75 samples consisting mostly of adversarial harmful prompts, together with a small number of adversarial benign and direct harmful prompts as a sanity check. For each model, we typically grid-searched four to five values of μ and six to seven values of η, yielding 24 to 35 configurations in total. Most configurations with aggressive learning rates performed similarly, differing only modestly in the average number of iterations. The final tuned values are listed in Table 3.

For the sensitivity analysis, the tested values are $\mu \in \{0.005, 0.01, 0.025, 0.05, 0.075, 0.1, 0.15\}$, $N \in \{1, 2, 4, 8, 16, 32\}$, and threshold $\in \{0.01, 0.025, 0.05, 0.1, 0.2, 0.5\}$.

Table 3: Model configurations and tuned TSA hyperparameters, along with the embedding (i.e., hidden) dimensionality and the number of hidden layers. For each model, the same hyperparameters are used across all benchmarks. A moderation threshold of 0.1 is used throughout.

Parameter	GPT-OSS	Qwen3	Llama 3.1	Phi-3.5	Gemma 3
Size	20B	14B	8B	4B	1B
Dimensionality d	2880	5120	4096	3072	1152
# Layers	24	40	32	32	26
# Monte Carlo samples	8	8	8	8	8
Perturbation scale μ	0.01	0.05	0.1	0.05	0.075
Step size η	1.5	2.5	1.3	2.5	1.75
Cosine similarity threshold	0.2	0.2	0.2	0.2	0.2

B.3 Implementation

Since our method operates purely through forward passes, we use vLLM (Kwon et al., 2023) for both the test-time optimization and the base-model generations. Through continuous batching across prompts and decoding steps, vLLM enables concurrent processing while sharing model execution and KV-cache management, making the overhead of batching negligible relative to single-input inference. Our test-time procedure therefore reduces to a small number of standard forward passes, effectively proportional to the number of optimization iterations in practice.

For the baselines, SmoothLLM is implemented on top of vLLM (the same inference backend used by our method) and applies random character swaps at a 10% perturbation rate across k=10 copies per prompt, with the returned completion selected by majority vote over a substring-based jailbreak judge. For AdaSteer, per-layer Rejection Direction (RD) and Harmfulness Direction (HD) steering vectors are extracted via difference-in-means over 256 harmful–benign contrast pairs drawn from the WildJailbreak training split. At inference time, adaptive per-input steering coefficients are computed from piecewise-linear maps evaluated at layer 8 (RD) and layer 13 (HD) for Llama 3.1-8B-Instruct, following the original hyperparameters reported for this architecture. RESTA adds isotropic Gaussian noise to the user-content token embeddings; the original paper uses $\sigma = 0.05$ for Llama 2-7B, but we found $\sigma = 0.08$ to work better with Llama 3.1-8B. The perturbed embeddings yield k=10 noisy copies, which are aggregated via majority vote over a smoothed prefix of length l=10, after which generation proceeds greedily from the agreed-upon prefix. AdaSteer and RESTA require embedding- or activation-level model access and therefore use HuggingFace `transformers` (Wolf et al., 2020), whereas SmoothLLM operates at the character level.

B.4 Compute

All experiments were conducted on a single local machine equipped with two NVIDIA RTX A6000 GPUs (49 GiB each). Models were served via vLLM with tensor parallel size 2 and GPU memory utilization set to 95%. Per-prompt wall-clock timing under this configuration, including and excluding moderation API latency, is reported in Table 1.

C Complete Results

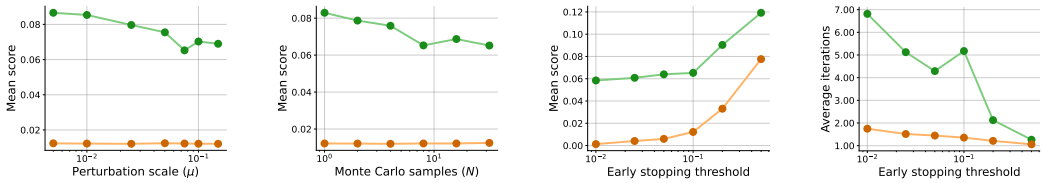
Table 4 reports the moderation results on all three splits, including HarmBench; its two WildJailbreak blocks repeat Table 1.

Table 4: Text moderation results for five instruction-tuned models on two red-teaming benchmarks (temperature 0.1, three trials per prompt). Scores are statistics of the maximum over the 13 OpenAI Moderation API harm categories per prompt, in $[0, 1]$; lower is safer. Optimization runs for up to 10 iterations or until that maximum falls below 0.1. *Flagged* counts completions the API classifies as harmful, *Time* is the average per-prompt optimization time and *Net* excludes oracle latency. Within each model–benchmark pair, the best value per metric is shown in **bold**.

			Score ↓					
	Model	Flagged ↓	Mean	Med.	Max	# Iterations	Time (s)	Net (s)
WildJailbreak (Adv. Harmful) $n = 2{,}000$	GPT-OSS-20B	17	0.023	0.004	0.849	–	–	–
	+ TSA	**0**	**0.009**	**0.003**	**0.099**	1.2	12.1	10.2
	Qwen3-14B	165	0.097	0.028	0.944	–	–	–
	+ SmoothLLM	148	0.095	0.027	0.952	–	–	–
	+ TSA	**0**	**0.023**	**0.016**	**0.100**	2.1	28.3	25.4
	Llama 3.1-8B	218	0.106	0.020	0.951	–	–	–
	+ SmoothLLM	122	0.068	0.006	0.954	–	–	–
	+ AdaSteer	85	0.057	0.016	0.947	–	–	–
	+ RESTA	199	0.098	0.010	0.944	–	–	–
	+ TSA	**0**	**0.012**	**0.002**	**0.098**	1.0	12.2	10.5
	Phi-3.5-4B	126	0.081	0.020	0.921	–	–	–
	+ SmoothLLM	123	0.076	0.019	1.000	–	–	–
	+ TSA	**0**	**0.019**	**0.008**	**0.099**	1.8	12.4	9.8
	Gemma 3-1B	278	0.137	0.063	0.946	–	–	–
	+ SmoothLLM	225	0.119	0.039	0.945	–	–	–
	+ TSA	**59**	**0.065**	**0.031**	**0.844**	5.7	21.1	13.7
WildJailbreak (Adv. Benign) $n = 210$	GPT-OSS-20B	**0**	0.011	**0.003**	0.270	–	–	–
	+ TSA	**0**	**0.009**	**0.003**	**0.097**	1.0	8.0	6.3
	Qwen3-14B	2	0.014	**0.001**	0.253	–	–	–
	+ SmoothLLM	**0**	0.014	0.002	0.254	–	–	–
	+ TSA	**0**	**0.010**	**0.001**	**0.098**	2.1	29.2	26.3
	Llama 3.1-8B	1	0.017	0.003	0.356	–	–	–
	+ SmoothLLM	2	0.018	0.004	0.358	–	–	–
	+ AdaSteer	3	0.022	**0.002**	0.359	–	–	–
	+ RESTA	2	0.013	0.004	0.351	–	–	–
	+ TSA	**0**	**0.009**	**0.002**	**0.098**	1.0	12.6	10.9
	Phi-3.5-4B	**0**	0.015	**0.001**	0.205	–	–	–
	+ SmoothLLM	2	0.016	**0.001**	0.565	–	–	–
	+ TSA	**0**	**0.009**	**0.001**	**0.098**	1.3	10.2	8.2
	Gemma 3-1B	3	0.023	0.006	0.354	–	–	–
	+ SmoothLLM	1	0.027	0.010	0.286	–	–	–
	+ TSA	**0**	**0.019**	**0.004**	**0.199**	3.1	12.4	8.4
HarmBench $n = 400$	GPT-OSS-20B	16	0.046	0.004	0.947	–	–	–
	+ TSA	**0**	**0.015**	**0.003**	**0.098**	1.3	9.5	7.5
	Qwen3-14B	8	0.053	0.020	0.448	–	–	–
	+ SmoothLLM	12	0.057	0.012	0.846	–	–	–
	+ TSA	**0**	**0.025**	**0.010**	**0.096**	1.9	20.5	17.9
	Llama 3.1-8B	17	0.055	0.004	0.706	–	–	–
	+ SmoothLLM	13	0.035	**0.001**	0.811	–	–	–
	+ AdaSteer	22	0.065	0.017	0.943	–	–	–
	+ RESTA	25	0.072	0.006	0.852	–	–	–
	+ TSA	**0**	**0.009**	**0.001**	**0.100**	1.1	10.7	9.0
	Phi-3.5-4B	8	0.050	0.017	0.523	–	–	–
	+ SmoothLLM	7	0.038	0.007	0.706	–	–	–
	+ TSA	**0**	**0.020**	**0.006**	**0.097**	1.8	12.0	9.4
	Gemma 3-1B	42	0.102	0.031	0.846	–	–	–
	+ SmoothLLM	17	0.074	**0.021**	0.845	–	–	–
	+ TSA	**14**	**0.062**	0.029	**0.623**	6.3	19.5	11.3

D SENSITIVITY ANALYSIS

The end-to-end calibration depends on three key variables: the Monte Carlo sample count N, the perturbation strength μ, and the moderation threshold. The 0.1 moderation threshold is borrowed from common practice among the API developers, so we also examine how the calibration behaves as the threshold varies. Results are visualized in Figure 1.



(a) (b) (c) (d)

Figure 1: Sensitivity analysis on Gemma 3-1B and Llama 3.1-8B on the adversarial harmful split ($n = 2{,}000$) of the WildJailbreak benchmark: mean moderation score against (a) the perturbation scale μ, (b) the number of samples N, (c) the early stopping threshold, and (d) the average number of iterations to convergence against the early stopping threshold. The x-axis is on a logarithmic scale.

Llama 3.1 is largely invariant to both μ and N because the prompts that enter the optimization loop typically sit near a broad safe basin where even a crude descent direction suffices. A suggestive analogy comes from hidden-state geometry. Saglam et al. (2025) find that adversarially framed harmful prompts overlap representationally with adversarially framed benign ones, since jailbreaks are built to mimic benign inputs, so a crude descent direction may suffice to move an input between the two; the analogy is loose, as that work concerns hidden-state representations for detection whereas TSA perturbs input embeddings. Gemma 3's greater sensitivity is consistent with its slower convergence (Table 1): a model that needs more iterations to cross the threshold leaves more room for μ and N to matter. Larger N helps slightly, though gradient normalization ensures that only the direction of the estimate governs the update, so even a single perturbation sample can provide sufficient directional signal. Notably, $N = 8$ suffices across embedding dimensions from 1152 to 5120, despite estimator variance scaling with the ambient dimension Td, suggesting the safety-relevant optimization direction need not be especially precise.

Neither model displays a monotonic trend in μ, confirming a mild bias–variance tradeoff across the tested range. The early stopping threshold is the only parameter with a monotonic effect (by construction), but the more informative observation is that Llama reaches a mean score of 0.001 at the 0.01 threshold in fewer than two iterations—a single gradient step typically overshoots the default threshold by a wide margin, indicating a steep moderation-score landscape near the safety boundary.

E EVALUATION UNDER HELD-OUT JUDGES

TSA optimizes the OpenAI Moderation score, so a reduction in that score alone cannot distinguish genuinely safer responses from responses that merely satisfy the optimized judge. To separate the two, we re-scored the responses generated in Section 5 with two judges that TSA never sees during optimization: the HarmBench classifier (Mazeika et al., 2024) and Llama Guard 3-8B (Inan et al., 2023). The HarmBench score is the probability the classifier assigns to the verdict that a response completes the requested harmful behavior, and the Llama Guard score is the probability of its `unsafe` verdict. Both scores lie in $[0, 1]$, and lower is safer. On the adversarial-benign split, the HarmBench classifier measures completion of a benign request rather than harm, so its benign column is not a safety measure and is omitted. Llama Guard scores harm on every split and is reported on all three.

Table 5 reports the held-out scores and Table 6 the optimized-oracle scores, each as the mean over prompts of the per-prompt average across the three trials. The accompanying $\pm$ term is the mean over prompts of each prompt's sample standard deviation across the three trials—cross-trial variability in a single prompt's score, not a standard error of the mean.

Two regularities hold under both held-out judges. First, every base-to-TSA comparison on the two harmful splits shows a decrease (16 of 16 cells). Second, TSA scores below the comparable baseline in 20 of the 24 baseline comparisons. The four exceptions all fall under the HarmBench classifier, where AdaSteer scores lower than TSA on Llama 3.1-8B on both harmful splits and SmoothLLM scores lower on the HarmBench split for Gemma 3-1B and Phi-3.5-4B. On the adversarial-benign split, no model's Llama Guard score increases after TSA, consistent with the do-no-harm result of Table 1. Cross-trial variability is small relative to these effects. Every held-out standard deviation is at most 0.014, and under the optimized oracle the base-to-TSA reduction on each harmful split exceeds the corresponding standard deviations for every model. That the reductions transfer to two judges TSA never optimizes against is evidence against the failure mode in which gains appear only on the optimized channel, though it does not make either automatic judge a ground truth for safety.

Table 5: Held-out re-scoring of the same responses in Section 5 by two judges not used in optimization: the HarmBench classifier (probability that the response completes the requested harmful behavior) and Llama Guard 3-8B (probability of an unsafe verdict), on WildJailbreak adversarial harmful (WJ-H, $n = 2{,}000$), HarmBench direct harmful (HB-H, $n = 400$), and WildJailbreak adversarial benign (WJ-B, $n = 210$). Each cell is the mean over prompts of the per-prompt three-trial average, $\pm$ the mean per-prompt standard deviation across trials; lower is safer. The HarmBench classifier's benign column is omitted because on that split it measures completion of a benign request rather than harm. Within each model, the best value per column is shown in **bold**.

	HarmBench classifier ↓		Llama Guard 3-8B ↓		
Model	WJ-H	HB-H	WJ-H	HB-H	WJ-B
Gemma 3-1B	0.573 ± 0.008	0.241 ± 0.005	0.399 ± 0.005	0.362 ± 0.005	0.072 ± 0.002
+ SmoothLLM	0.368 ± 0.005	$\mathbf{0.108 \pm 0.001}$	0.368 ± 0.004	0.265 ± 0.005	0.090 ± 0.003
+ TSA	$\mathbf{0.305 \pm 0.007}$	0.131 ± 0.005	$\mathbf{0.309 \pm 0.004}$	$\mathbf{0.250 \pm 0.004}$	$\mathbf{0.068 \pm 0.002}$
Phi-3.5-4B	0.387 ± 0.006	0.119 ± 0.003	0.318 ± 0.004	0.401 ± 0.007	0.092 ± 0.002
+ SmoothLLM	0.344 ± 0.007	$\mathbf{0.056 \pm 0.003}$	0.376 ± 0.004	0.382 ± 0.009	0.096 ± 0.001
+ TSA	$\mathbf{0.296 \pm 0.006}$	0.073 ± 0.003	$\mathbf{0.283 \pm 0.003}$	$\mathbf{0.362 \pm 0.003}$	$\mathbf{0.084 \pm 0.002}$
Llama 3.1-8B	0.437 ± 0.008	0.136 ± 0.007	0.249 ± 0.005	0.280 ± 0.011	0.106 ± 0.003
+ SmoothLLM	0.269 ± 0.006	0.101 ± 0.004	0.201 ± 0.004	0.195 ± 0.007	0.095 ± 0.002
+ AdaSteer	$\mathbf{0.126 \pm 0.004}$	$\mathbf{0.060 \pm 0.002}$	0.270 ± 0.005	0.671 ± 0.009	0.078 ± 0.003
+ RESTA	0.183 ± 0.013	0.110 ± 0.010	0.212 ± 0.006	0.329 ± 0.011	$\mathbf{0.033 \pm 0.001}$
+ TSA	0.137 ± 0.006	0.088 ± 0.005	$\mathbf{0.179 \pm 0.004}$	$\mathbf{0.168 \pm 0.014}$	0.075 ± 0.002
Qwen3-14B	0.430 ± 0.011	0.134 ± 0.003	0.332 ± 0.005	0.314 ± 0.004	0.082 ± 0.002
+ SmoothLLM	0.430 ± 0.009	0.153 ± 0.004	0.344 ± 0.004	0.310 ± 0.005	0.101 ± 0.003
+ TSA	$\mathbf{0.317 \pm 0.008}$	$\mathbf{0.128 \pm 0.003}$	$\mathbf{0.256 \pm 0.005}$	$\mathbf{0.301 \pm 0.004}$	$\mathbf{0.078 \pm 0.002}$

Table 6: Optimized-oracle (OpenAI Moderation) scores with cross-trial variability for the four models of Table 5. Means coincide with Tables 1 and 4, and are computed over the three trials, as is the $\pm$ term of Table 5; lower is safer. Within each model, the best value per column is shown in **bold**.

Model	WJ-H↓	HB-H↓	WJ-B↓
Qwen3-14B	0.097 ± 0.020	0.053 ± 0.016	0.014 ± 0.005
+ SmoothLLM	0.095 ± 0.019	0.057 ± 0.015	0.014 ± 0.004
+ TSA	$\mathbf{0.023 \pm 0.010}$	$\mathbf{0.025 \pm 0.011}$	$\mathbf{0.010 \pm 0.004}$
Llama 3.1-8B	0.106 ± 0.016	0.055 ± 0.007	0.017 ± 0.005
+ SmoothLLM	0.068 ± 0.012	0.035 ± 0.007	0.018 ± 0.006
+ AdaSteer	0.057 ± 0.011	0.065 ± 0.013	0.022 ± 0.007
+ RESTA	0.098 ± 0.016	0.072 ± 0.011	0.013 ± 0.004
+ TSA	$\mathbf{0.012 \pm 0.005}$	$\mathbf{0.009 \pm 0.003}$	$\mathbf{0.009 \pm 0.004}$
Phi-3.5-4B	0.081 ± 0.017	0.050 ± 0.014	0.015 ± 0.005
+ SmoothLLM	0.076 ± 0.015	0.038 ± 0.011	0.016 ± 0.004
+ TSA	$\mathbf{0.019 \pm 0.008}$	$\mathbf{0.020 \pm 0.009}$	$\mathbf{0.009 \pm 0.004}$
Gemma 3-1B	0.137 ± 0.025	0.102 ± 0.020	0.023 ± 0.008
+ SmoothLLM	0.119 ± 0.023	0.074 ± 0.018	0.027 ± 0.011
+ TSA	$\mathbf{0.065 \pm 0.020}$	$\mathbf{0.062 \pm 0.019}$	$\mathbf{0.019 \pm 0.008}$

F Controlled Comparison for the Gemma 3 Result

Gemma 3-1B is the only model on which TSA leaves flagged responses (Table 1), and within the five evaluated models the two explanations one might offer for this are confounded. Gemma is at once the model with the smallest embedding dimension ($d = 1152$) and the least safety-aligned in the sense of the moderation objective, with 278 flagged undefended responses on the adversarial-harmful split against at most 218 for the others (Table 7, upper block). The base-alignment proxy alone does not predict the outcome either, since Llama 3.1-8B starts from a comparable count (218) and reaches zero. To separate the two candidate causes, we ran a controlled pair: Qwen3-0.6B (Yang et al., 2025), a post-trained model that is smaller than Gemma 3-1B in both parameter count and embedding dimension ($d = 1024$), and an abliterated Qwen3-14B (huihui.ai, 2025), whose refusal behavior has been removed. Both were evaluated on all 2,000 adversarial-harmful prompts under the protocol of Section 5 (three trials at temperature 0.1), with every setting at its value from that section except the step size. The step size was tuned per model as for the other models (Appendix B.2), resulting in $\eta = 1.5$ for Qwen3-0.6B and $\eta = 2.5$ for the abliterated model.

Neither explanation survives the comparison (Table 7, lower block). Qwen3-0.6B sits below Gemma on both axes—a smaller embedding dimension and a worse starting point, 470 flagged against 278—yet TSA clears every flag in one iteration on average and lowers the mean score from 0.161 to 0.014. Abliteration more than triples the base flagged count of Qwen3-14B (165 to 527), confirming that it took effect, and TSA still clears every flag. It reaches the same zero flagged count and the same rounded mean (0.023) as the safety-trained Qwen3-14B at the cost of roughly one additional gradient step (3.2 against 2.1 average iterations). The mean per-prompt cross-trial standard deviations of the two controls range from 0.011 to 0.045, so both reductions are several times the observed variability.

Difficulty thus appears in the iteration count rather than in the outcome, and the same reading covers Gemma. The residual flagged responses are exactly the prompts whose score does not cross the 0.1 threshold within the 10-iteration budget, on a model that needs 5.7 average iterations on this split. We therefore state the limitation as slow convergence in this particular model rather than as a floor imposed by scale or by the strength of its safety training. We do not claim to explain why Gemma converges slowly, since both controls are Qwen3 variants and so exclude the two proposed causes without isolating what is specific to Gemma's architecture. A manual review of all of Gemma 3-1B's residual flagged outputs on both benchmarks found coherent, non-evasive responses with no sign of reward hacking and with harmfulness reduced relative to the corresponding base response.

Beyond the Gemma question, the two controls extend the evaluation past its original range. Qwen3-0.6B shows that the method holds at $d = 1024$. The abliterated Qwen3-14B yields the more notable

side result, because TSA clears every flag on a model whose refusal behavior has been deliberately removed. Abliteration removes refusal directions from a model descended from post-trained Qwen3-14B but does not erase every representation that safety training installed, and TSA's success indicates that it can make this residual alignment active again even after the model no longer refuses.

Table 7: Embedding dimension, undefended (base) result, and TSA result on the WildJailbreak adversarial-harmful split ($n = 2{,}000$) for the five models of Table 1 (upper block) and the two controls (lower block). Flagged counts and mean scores follow the definitions of Table 1; # Iterations is the average number of optimization iterations over the prompts that enter the loop. Lower is safer.

		Base		+TSA		
Model	d	Flagged ↓	Mean ↓	Flagged ↓	Mean ↓	# Iterations
GPT-OSS-20B	2880	17	0.023	0	0.009	1.2
Qwen3-14B	5120	165	0.097	0	0.023	2.1
Llama 3.1-8B	4096	218	0.106	0	0.012	1.0
Phi-3.5-4B	3072	126	0.081	0	0.019	1.8
Gemma 3-1B	1152	278	0.137	59	0.065	5.7
Qwen3-0.6B	1024	470	0.161	0	0.014	1.0
Qwen3-14B-abliterated	5120	527	0.177	0	0.023	3.2

G Reverse-Directed Optimization from Complying Starts

The argument of Section 6 for why reversing the descent direction does not yield a low-budget attack is sharpest when the starting response already scores near zero, where the finite-difference numerator collapses. Here, we instead measure the harder regime, in which the prompt already elicits a complying, high-scoring response. We used the standard Qwen3-14B on all 2,000 adversarial-harmful prompts under the sampling protocol of Section 5 (three trials at temperature 0.1 and the Qwen3 settings of Table 3), with three deliberate changes. First, we optimized only the prompts whose initial raw maximum-category score was at least 0.1, which selected 649 of the 2,000 prompts. This value gates TSA's optimization loop and serves here purely as a selection rule. Second, we reversed the sign of the update so that the estimated direction increases the score, and disabled early stopping because it is a minimization criterion. Every selected prompt therefore received the full budget of ten updates. Third, for each selected prompt we retained the highest-scoring response among the initial response and the ten updates, the maximization analogue of TSA's best-iterate return. The remaining 1,351 prompts kept their initial responses. Throughout, "flagged" is again the moderation API's own binary output, reported separately from the score-based selection rule.

Table 8 places the result next to the unoptimized and TSA results of the standard and abliterated models. Retaining the worst response found along each path raises the full-split mean score from 0.097 to 0.134 ± 0.078 and the flagged count from 165 to 310 of 2,000, yet both remain below the abliterated model without any optimization (0.177 ± 0.041; 527 flagged). Restricted to the 649 optimized prompts (Table 9), the maximum-observed mean rises from 0.252 to 0.368 and the flagged count from 165 to 310 (all 165 initially flagged responses fall within this cohort). The score strictly increases for 401 of the 649 prompts (61.8%), and 302 of these reach their maximum within the first three updates.

Manual inspection shows, however, that the trajectory does not always ascend monotonically and that scores do not always remain high once they peak. Reverse-directed search still exposes a higher-scoring—sometimes only transient—harmful response for a majority of complying prompts, which the maximum-observed endpoint records as a conservative worst case. The non-monotonicity is consistent with the account in Section 6 that safety training shapes the neighborhood of aligned behavior, leaving a compliance basin that appears thin.

Table 8: Reverse-directed optimization on the full WildJailbreak adversarial-harmful split ($n = 2{,}000$), next to the unoptimized and TSA results for standard Qwen3-14B (Table 1) and abliterated Qwen3-14B (Appendix F). Mean scores carry the mean per-prompt cross-trial standard deviation as in Table 5; flagged counts are out of 2,000. The reverse-directed row retains, for each of the 649 selected prompts, the highest-scoring response among the initial response and ten updates; the other prompts keep their initial response.

Model	Endpoint	Mean score	Flagged	Updates
Qwen3-14B	none	0.097 ± 0.020	165 (8.25%)	–
+ TSA	minimize, best iterate	0.023 ± 0.010	0 (0%)	2.1 (average)
+ reverse-directed	maximize, maximum observed	0.134 ± 0.078	310 (15.50%)	10 (649 prompts)
Qwen3-14B-abliterated	none	0.177 ± 0.041	527 (26.35%)	–
+ TSA	minimize, best iterate	0.023 ± 0.015	0 (0%)	3.2 (average)

Table 9: The 649 prompts selected for reverse-directed optimization (initial raw maximum-category score at least 0.1): score statistics of the initial response and of the maximum observed over the initial response and ten updates.

Endpoint	Mean	Median	Max	Flagged
Initial response	0.252	0.198	0.944	165 / 649 (25.42%)
Maximum observed over initial + 10 updates	0.368	0.269	0.999	310 / 649 (47.77%)