

WRDO: Weighted Refusal Direction Optimization in Aligned Language Models

Grigoriy Evseev^{1,2}, Vladimir Solodkin^{1,2,3}, Dmitry Bylinkin^{1,2},
Daniil Medyakov^{1,2}, Vladimir Aletov^{1,2}, Aleksandr Beznosikov^{1,2,4}

¹Moscow Independent Research Institute of Artificial Intelligence (MIRIAI)

²Basic Research of Artificial Intelligence Laboratory (BRAIn Lab)

³T-Technologies

⁴Innopolis University

Abstract

Aligned large language models are trained to comply with benign requests while refusing harmful ones. However, recent work shows that this refusal behavior is fragile and can often be disrupted by simple activation- or weight-space interventions. In particular, refusal has been shown to be largely mediated by a single residual-stream direction, typically computed as a difference-of-means vector between harmful and harmless prompts. However, this standard construction aggregates heterogeneous harmful prompts into one uniform average, potentially conflating distinct refusal-related phenomena. In this work, we study whether refusal interventions can be strengthened by learning how to combine prompt-specific refusal directions. We propose Weighted Refusal Direction Optimization (WRDO), a procedure that constructs a composite refusal direction from a bank of per-prompt directions using learned aggregation weights. This approach allows different harmful prompts to contribute unequally to the final intervention direction. We evaluate WRDO against prior single-direction methods on three open-weight language models using three guard judges and HarmBench as an external evaluator. We further measure the effect of the intervention on general model capability through standard utility benchmarks. Finally, we investigate a defense-oriented setting in which the same optimization machinery is used in reverse to restore refusal behavior in an ablated model checkpoint. Our results show that weighted aggregation provides a more effective and flexible mechanism for manipulating refusal representations. Code is available at <https://github.com/brain-lab-research/WRDO>

1 Introduction

Large language models deployed in user-facing settings are aligned to follow benign instructions and to refuse harmful ones, with this behavior shaped

by a combination of supervised fine-tuning, instruction tuning and preference optimization (Ouyang et al., 2022; Bai et al., 2022). The resulting refusal mechanism is, in practice, brittle under both prompt-level adversarial attacks (Zou et al., 2023; Wei et al., 2023) and direct interventions on the model itself, which has motivated a mechanistic line of work that asks how refusal is represented internally and which components of the model implement it. A central finding of this line is that refusal is, to a first approximation, mediated by a single direction in the residual stream of the transformer (Arditi et al., 2024): ablating one difference-of-means vector, computed between activations on harmful and harmless prompts at a chosen layer, is sufficient to disable refusal in a broad set of open-weight aligned models. The same work establishes an equivalence between activation ablation and weight-space orthogonalization of the attention output and feed-forward down projections, which makes the intervention a one-shot edit of the model with no inference-time overhead, and public toolkits such as Heretic (Weidmann, 2025) operationalize this recipe end-to-end.

The standard procedure averages every harmful prompt into a single axis. A disinformation prompt, a malware prompt and a privacy-violation prompt all enter the same mean, and the resulting direction is diffuse in the sense that it conflates phenomena that need not be aligned in the residual stream. A natural alternative is to retain a per-prompt direction r_i for each training prompt and to combine them with a coefficient vector w . The simplest combination is the uniform average, $r = \text{normalize}(\sum_i r_i / N)$, but there is no a priori reason for every prompt to contribute equally. Performance-aware weighted aggregation is known to beat uniform averaging whenever the aggregated components differ in quality, from ensembling ranked classifiers (Ahsen et al., 2019) to federated averaging (Shaikh et al., 2026) and adapter

selection (Veprikov et al., 2026), and per-prompt refusal directions are heterogeneous in exactly that way. In this work, we research the question of whether computing refusal direction using the aggregation weights yields a stronger intervention than the single-direction recipe.

Our Contributions:

- We propose Weighted Refusal Direction Optimization (WRDO), a procedure that obtains a refusal direction by learning aggregation weights w over a bank of per-prompt directions, together with two concrete instantiations of WRDO based on separable CMA-ES and a proposed GRPO variant with importance sampling (GRPO-IS).
- We show empirically that the proposed WRDO instantiations outperform all considered prior single-direction baselines across three open-weight models and three guard judges, and that this superiority transfers to HarmBench ruling out judge-specific overfitting.
- We obtain that WRD-GRPO-IS preserves the general capability of the intervened model substantially better than every other considered method on standard utility benchmarks.
- We additionally study a defense setup that reuses the same machinery in reverse to restore refusal on an ablated checkpoint, illustrating an alternative use of the proposed approach.

The remainder of the paper is organized as follows. Section 2 formalizes the WRDO problem. Section 3 presents the two WRDO instantiations introduced in this work, namely an adaptation of separable CMA-ES and a proposed GRPO variant with importance sampling, together with the three single-direction baselines against which they are compared. Section 4 specifies the experimental setup, and Sections 5–6 report the empirical results, ablations and discussion.

2 Problem Formulation

An aligned language model is denoted as f_θ , a harmful training set of prompts $\mathcal{D}_{\text{train}}^{\text{harm}} = \{x_i\}_{i=1}^N$, a held-out harmful validation set $\mathcal{D}_{\text{val}}^{\text{harm}}$ and a harmless baseline corpus $\mathcal{D}_{\text{benign}}$. The hidden dimensionality of the residual stream is d , and L denotes the number of transformer blocks.

2.1 Candidate bank construction

Let $h^{(l)}(x) \in \mathbb{R}^d$ denote the residual-stream activation of f_θ on prompt x at layer l , evaluated at the last token position. Activations are extracted at every transformer block, including the embedding output, so that each prompt is summarized by a tensor in $\mathbb{R}^{(L+1) \times d}$. The mean benign activation $\mu_{\text{benign}}^{(l)} = \mathbb{E}_{x \sim \mathcal{D}_{\text{benign}}} [h^{(l)}(x)]$ is computed at every layer.

For each training prompt x_i we define a per-prompt layerwise direction by taking the difference $h^{(l)}(x_i) - \mu_{\text{benign}}^{(l)}$ and applying L_2 normalization at every layer independently,

$$r_i^{(l)} = \frac{h^{(l)}(x_i) - \mu_{\text{benign}}^{(l)}}{\|h^{(l)}(x_i) - \mu_{\text{benign}}^{(l)}\|_2}, \quad (1)$$

and stack the per-layer vectors into a layerwise tensor $r_i \in \mathbb{R}^{(L+1) \times d}$. The candidate bank $\mathcal{R} = \{r_i\}_{i=1}^N$ contains one such tensor per harmful training prompt, with $N = |\mathcal{D}_{\text{train}}^{\text{harm}}|$. The bank is computed once per model and is shared across all optimization methods.

2.2 Weighted direction

A coefficient vector $w \in \mathbb{R}^N$ aggregates the bank of Section 2.1 by a scalar weighting that is shared across layers and across hidden coordinates, followed by a per-layer L_2 renormalization. Concretely, the composite direction at layer l is

$$r^{(l)}(w) = \frac{\sum_{i=1}^N w_i r_i^{(l)}}{\|\sum_{i=1}^N w_i r_i^{(l)}\|_2}. \quad (2)$$

The per-layer normalization enforces $\|r^{(l)}(w)\|_2 = 1$ at every layer and removes the overall positive scale of w , so that only the orientation of the composite at each layer is varied. Intervention intensity is controlled separately through the layerwise schedule described in Section 2.3. The single-direction baseline of (Arditi et al., 2024) is recovered, in this notation, by replacing the per-prompt r_i by a single difference-of-means tensor (computed between the mean harmful and mean benign activations at every layer) and setting $w_1 = 1$ with $N = 1$.

2.3 Intervention operator

The primary intervention is weight-space orthogonalization. We adopt the intervention operator of the Heretic toolkit (Weidmann, 2025). Given

the layerwise composite direction $r^{(l)}(w)$ of Equation 2, every attention output projection $W_O^{(l)}$ and every feed-forward down-projection $W_{\text{down}}^{(l)}$ at block l is replaced by

$$W^{(l)'} = W^{(l)} - \alpha^{(l)} r^{(l)}(w) r^{(l)}(w)^\top W^{(l)}, \quad (3)$$

where $\alpha^{(l)}$ is the fixed layerwise intensity schedule. The full form of $\alpha^{(l)}$, together with the values of the auxiliary parameters that we hold fixed throughout the paper, is given in Appendix B. The only departure from the Heretic recipe is the choice of $r^{(l)}(w)$: Heretic plugs in the single direction of (Arditi et al., 2024), whereas WRDO substitutes the per-prompt composite of Equation 2.

The composition of f_θ with the operator of Equation 3, evaluated at the layerwise direction $r(w) = \{r^{(l)}(w)\}_l$ and a given α , is denoted $T_{r(w)}(\theta)$. We write $\pi_{T_{r(w)}(\theta)}(\cdot | x)$ for the conditional distribution over generations produced by the intervened model under greedy decoding.

2.4 Harmfulness metric

For each prompt x and response y , every guard model used in this paper outputs a binary safe/unsafe decision $J(x, y) \in \{0, 1\}$, with $J = 1$ denoting an unsafe response. The attack success rate (ASR) of the intervened model is the average of this per-prompt binary signal over the validation distribution,

$$\text{ASR}_J(w) = \mathbb{E}_{x \sim \mathcal{D}_{\text{val}}^{\text{harm}}} [J(x, \pi_{T_{r(w)}(\theta)}(\cdot | x))]. \quad (4)$$

In practice, ASR_J is estimated on a validation batch as the empirical mean of the per-prompt decisions.

The optimization objective is the negative attack success rate,

$$\mathcal{L}_J(w) = -\text{ASR}_J(w), \quad (5)$$

which is shared by every WRDO method. Some optimizers may augment this objective with method-specific regularizers when needed (see WRD-GRPO-IS in Section 3.2 for a KL-regularized variant). The attack setup minimizes $\mathcal{L}_J$,

$$w^* = \arg \min_{w \in \mathcal{W}} \mathcal{L}_J(w), \quad (6)$$

where the feasible set $\mathcal{W} \subset \mathbb{R}^N$ is a per-coordinate box on w , with bounds and initialization detailed in Appendix A.

3 Methods

3.1 Baselines

Basic RD. We refer to the single-direction refusal-ablation procedure of (Arditi et al., 2024) as Basic RD throughout the paper. A single layerwise direction is computed as the per-layer L_2 -normalized difference of means

$$r_{\text{Basic}}^{(l)} = \frac{\mu_{\text{harm}}^{(l)} - \mu_{\text{benign}}^{(l)}}{\|\mu_{\text{harm}}^{(l)} - \mu_{\text{benign}}^{(l)}\|_2} \quad (7)$$

on the training split, where $\mu_{\text{harm}}^{(l)}$ is the mean activation of the harmful training prompts at layer l and $\mu_{\text{benign}}^{(l)}$ is the mean activation of the benign baseline corpus. The resulting layerwise direction is applied through the operator of Equation 3. There is no candidate bank and no optimization. This is the reference against which any benefit of per-prompt aggregation must be measured.

Heretic. We additionally include the public Heretic toolkit (Weidmann, 2025) as a further baseline. Heretic applies the same weight-space orthogonalization of Equation 3 to the single Basic RD direction, but tunes the layerwise intensity schedule with its own internal Bayesian search rather than using the fixed schedule of Appendix B. There is no per-prompt candidate bank and no learned w ; the baseline isolates the contribution of schedule tuning alone, independent of the per-prompt aggregation introduced by WRDO. We run the released pipeline out of the box, with default hyperparameters.

RDO. We further include the gradient-based refusal-direction baseline of (Wollschläger et al., 2025), denoted RDO. Instead of taking a difference of means as in Basic RD, RDO parameterizes a single refusal direction per layer and optimizes it directly with gradient descent against a refusal-suppression objective on the harmful training split, exploiting differentiability through the model’s hidden activations. We use the released implementation of Wollschläger et al. (2025) unchanged: the comparison against Basic RD isolates the contribution of learning the direction itself. As with Basic RD and Heretic, there is no per-prompt candidate bank and no learned aggregation w .

RDO-CONE. Wollschläger et al. (2025) also argue that refusal is mediated by a cone of directions

rather than by a single line, and release a concept-cone construction alongside the single-direction RDO objective. This is the prior construction closest to the motivation of WRDO, since it likewise rejects the assumption that one difference-of-means axis captures refusal, so we include it as a separate baseline denoted RDO-CONE. It differs from WRDO in where the multiplicity is placed: RDO-CONE learns a set of mutually constrained directions that span a cone and ablates the whole cone, whereas WRDO keeps a fixed bank of per-prompt directions and learns only how to aggregate them into the composite of Equation 2. We again use the released implementation of Wollschläger et al. (2025) unchanged, with the authors’ default cone dimensionality.

3.2 WRD optimizers

For optimizing the refusal-direction weights in the WRDO pipeline, we propose two distinct procedures. The first is a WRDO-tailored adaptation of separable CMA-ES, a well-known derivative-free optimizer suited to our low-dimensional setting. The second is a novel scheme based on group-relative policy optimization with importance sampling, designed specifically for the WRDO objective.

WRD-CMAES. CMA-ES (Hansen and Ostermeier, 2001; Hansen, 2023) is a standard derivative-free optimizer for continuous black-box objectives, well suited to the noisy and gradient-free nature of the WRDO loss. In our setting we apply it to $\mathcal{L}_J$ of Equation 5: the search variables are the aggregation coefficients w of the candidate bank, the search space is described in Appendix A, and each black-box query corresponds to one full WRDO evaluation, i.e. building the composite direction $r(w)$, intervening the model through the operator of Section 2.3, and measuring the resulting ASR on the validation batch. The concrete sampler choice, hyperparameter values and search-space bounds are listed in Appendix F.

Relation to evolution strategies. Both instantiations belong to the evolution-strategy family (Hansen and Ostermeier, 2001; Wierstra et al., 2014; Salimans et al., 2017): a distribution over w is maintained, perturbed samples are scored by the black-box objective, and the distribution is moved towards the better-scoring ones. WRD-CMAES is its canonical member, and WRD-GRPO-IS reads as an ES with importance mixing, with the Gaussian

rollout of Equation 11 as the search distribution and the weights $\tilde{v}_m$ as the mixing correction, so running both compares ES against RL under one objective and one budget. The analogy stops at the reward: NES-style updates see only the scalar fitness of each perturbation, whereas WRD-GRPO-IS routes it through token-level policy ratios and can therefore carry the KL term of Equation 8.

WRD-GRPO-IS. The target loss function described in 2.4 is a non-differentiable function, so it is natural to consider a reinforcement learning approach to its optimization. To keep the intervened model close to the clean base model on benign prompts, WRD-GRPO-IS augments the bare WRDO objective of Equation 5 with a KL-divergence penalty against the policy π_{f_θ} evaluated on a random batch from $\mathcal{D}_{\text{benign}}$, yielding the method-specific objective

$$\tilde{\mathcal{L}}_J(w) = \mathcal{L}_J(w) + \beta \text{KL}_w, \quad (8)$$

with $\beta \geq 0$ and $\text{KL}_w = \mathbb{E}_{x \sim \mathcal{D}_{\text{benign}}} [\text{KL}(\pi_{T_{r(w)}(\theta)}(\cdot|x) \parallel \pi_{f_\theta}(\cdot|x))]$ the KL divergence of the intervened policy from the clean policy on the benign batch. The KL term is specific to WRD-GRPO-IS: the WRD-CMAES sampler consumes the ASR-only signal of Equation 5 without any regularization.

Group Relative Policy Optimization. Let us denote the language model parameterized by θ as a policy π_θ . Given a query x drawn from a query set $\mathcal{D}$, it induces a distribution over responses y , whose likelihood factorizes as

$$\pi_\theta(y | x) = \prod_{t=1}^{|y|} \pi_\theta(y_t | x, y_{<t}),$$

where $|y|$ denotes the number of tokens in y . A query-response pair (x, y) is scored by a reward model $\mathcal{S}$, yielding a per-prompt reward $\mathcal{S}(x, y) \in \{0, 1\}$; in our setting $\mathcal{S}(x, y) = J(x, y)$ is the binary judge decision of Equation 4. The standard GRPO algorithm computes a relative advantage for each response within a group of G responses generated for the same query from the old policy $\pi_{\theta_{\text{old}}}$. Specifically, GRPO optimizes

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x, \{y_i\} \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \left(\rho_{i,t}(\theta) \hat{A}_i, \text{clip}(\rho_{i,t}(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_i \right) \right], \quad (9)$$

where the token-level importance ratio is

$$\rho_{i,t}(\theta) = \frac{\pi_{\theta}(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})},$$

We write importance ratios as ρ throughout this section to keep them distinct from the aggregation coefficients $w \in \mathbb{R}^N$ of Section 2.2, which remain the only trainable quantity of WRDO. and all tokens in a response share the same group-relative advantage

$$\hat{A}_i = \frac{\mathcal{S}(x, y_i) - \text{mean}(\{\mathcal{S}(x, y_j)\}_{j=1}^G)}{\text{std}(\{\mathcal{S}(x, y_j)\}_{j=1}^G)}.$$

Group Relative Policy Optimization with Importance Sampling. In the standard GRPO formulation (9), the group of responses for each query is sampled from the old policy $\pi_{\theta_{\text{old}}}$, and the clipped objective uses the token-level importance ratio between the current and old policies. Here, we consider a stratified data-collection scheme. For each query x , we sample one response from each of M behavior policies $\{\mu_m\}_{m=1}^M$ that are obtained by shifting the original model weights with formula (3), i.e., $y_m \sim \mu_m(\cdot \mid x)$. As the resulting group is off-policy with respect to $\pi_{\theta_{\text{old}}}$, directly applying (9) would generally yield a biased estimate of the GRPO objective. We introduce sequence-level importance sampling weights to correct this mismatch while retaining the original ratio against $\pi_{\theta_{\text{old}}}$.

Specifically, for each response y_m sampled from μ_m , we define the sequence-level importance weight

$$v_m = \frac{\pi_{\theta_{\text{old}}}(y_m \mid x)}{\mu_m(y_m \mid x)}.$$

In practice, we stabilize the weights $\{v_m\}_{m=1}^M$ by applying clipping, denoted as $\tilde{v}_m$, and treat them as constants with respect to θ during optimization. This reweighting ensures that the overall objective remains consistent with the GRPO surrogate.

Analogous to GRPO, we assign a shared advantage to all tokens in a response. However, to preserve the control-variate property under off-policy sampling, we compute the group baseline using the same sequence-level weights. Given the group $\{y_m\}_{m=1}^M$ for query x , we first form the importance-weighted baseline

$$b(x) = \frac{\sum_{m=1}^M \tilde{v}_m \mathcal{S}(x, y_m)}{\sum_{m=1}^M \tilde{v}_m},$$

and define the normalized group-relative advantage as

$$\hat{A}_m = \frac{\mathcal{S}(x, y_m) - b(x)}{\text{std}(\{\mathcal{S}(x, y_j)\}_{j=1}^M)},$$

where all tokens in y_m share the same advantage, i.e., $\hat{A}_{m,t} = \hat{A}_m$ for all t .

Finally, our GRPO objective with importance sampling is

$$\mathcal{J}_{\text{GRPO-IS}}(\theta) = \mathbb{E}_{x, y_m \sim \mu_m(\cdot \mid x)} \left[\frac{1}{M} \sum_{m=1}^M \tilde{v}_m \frac{1}{|y_m|} \sum_{t=1}^{|y_m|} \min(\rho_{m,t}(\theta) \hat{A}_m, \text{clip}(\rho_{m,t}(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_m) \right], \quad (10)$$

where the token-level importance ratio is defined as

$$\rho_{m,t}(\theta) = \frac{\pi_{\theta}(y_{m,t} \mid x, y_{m,<t})}{\pi_{\theta_{\text{old}}}(y_{m,t} \mid x, y_{m,<t})}.$$

Compared with standard GRPO, our objective (10) differs in that the group is collected by stratified sampling from multiple behavior policies $\{\mu_m\}$, and a sequence-level importance weight $\tilde{v}_m$ is used to correct the off-policy sampling back to $\pi_{\theta_{\text{old}}}$.

In our setting, the trainable parameter is the coefficient vector $w \in \mathbb{R}^N$ of Section 2.2 and the base policy $\pi_{\theta_{\text{old}}}$ is the clean model f_{θ} . At each training step, the M behavior policies $\{\mu_m\}_{m=1}^M$ are constructed by perturbing the current w with Gaussian noise,

$$w_m = w + \varepsilon_m, \quad \varepsilon_m \sim \mathcal{N}(0, \sigma^2 I_N), \quad (11)$$

with $\varepsilon_0 = 0$. Each w_m defines a layerwise composite direction $r^{(l)}(w_m)$ via Equation 2 and an intervened model $\mu_m = T_{r(w_m)}(\theta_{\text{old}})$ via the operator of Section 2.3. Only w is updated by the surrogate of Equation 10, while the candidate bank $\mathcal{R}$, the base parameters θ_{old} and the schedule $\alpha^{(l)}$ are held fixed throughout training.

4 Experimental Setup

4.1 Data

The harmful pool is the union of six public benchmarks, namely AdvBench (Zou et al., 2023), MaliciousInstruct (Huang et al., 2023), the TDC2023 (Mazeika et al., 2023), HarmBench (Mazeika et al., 2024), JailbreakBench (Chao et al., 2024) and StrongREJECT (Souly et al., 2024), deduplicated

by exact prompt-string match and treated as a single distribution. The deduplicated pool is pre-partitioned into a training set, used only to compute the candidate bank, a validation set, used as the optimization objective in Equation 4, and a test set, used only for the final evaluation. Concrete sampling seeds and the source of the prepared splits are documented in Appendix C. The harmless baseline $\mathcal{D}_{\text{benign}}$ is a fixed-size sub-sample of Alpaca (Taori et al., 2023) used at bank-construction time.

4.2 Models

We use three open-weight models: Qwen/Qwen3-8B-Base (Team, 2025), allenai/OLMo-3-1025-7B (OLMo Team, 2025) and Olmo-3-7B-Instruct (OLMo Team, 2025). All evaluations use greedy decoding (do_sample=False) with a maximum of 100 new tokens per completion, one completion per prompt, the default system prompt of each model and no few-shot context.

4.3 Judges

We use three guard models, namely allenai/wildguard (Han et al., 2024), Qwen/Qwen3Guard-Gen-8B (Qwen Team, 2025) and meta-llama/Llama-Guard-3-8B (Inan et al., 2023). Each guard exposes its own decoding pipeline that maps a (prompt, response) pair to a binary safe/unsafe label, which we use directly as $J(x, y) \in \{0, 1\}$ in Equation 4. The per-judge ASR_J $\in [0, 1]$ is then obtained by averaging these per-prompt binary decisions over the test set and is reported separately for every judge. HarmBench (Mazeika et al., 2024) with the official classifier cais/HarmBench-Llama-2-13b-cls provides an additional external evaluator that is never used during optimization.

4.4 Metrics

The primary metric is the ASR_J of Equation 4, reported separately for each of the three judges J on the held-out test set. The secondary metrics are the utility benchmarks of Section 4.5, the HarmBench ASR of Section 5.4.

4.5 Utility benchmarks

Utility preservation is evaluated through the Language Model Evaluation Harness (Gao et al., 2024) on arc_easy and arc_challenge (Clark et al., 2018), mmlu (Hendrycks et al., 2020), gsm8k and word-level perplexity on wikitext. Accuracy and

perplexity are reported separately, with the direction of improvement made explicit in the tables.

5 Results

5.1 Attack setup

Table 1 reports the test-set ASR of every method on the three base models, evaluated separately against the three guard judges, at the two extremes of the peak intensity grid $\alpha \in \{1.0, 3.0\}$ of the layerwise schedule introduced in Section 2.3 (Equation 3; full form in Appendix B). The intermediate intensities $\alpha \in \{1.5, 2.0\}$ are deferred to Table 5 in Appendix D, where the same conclusions hold. The no-intervention rows provide the absence-of-attack baselines on the same test set. The Heretic entries are duplicated across the α columns because Heretic does not consume this hyperparameter from the outside: it selects its own shift coefficient internally through the toolkit’s Bayesian search, so a single value is reported per model and judge and repeated across the α grid for layout consistency. The RDO rows are duplicated for the same reason: RDO does not expose α as a hyperparameter, the intensity is baked into the gradient-learned direction itself, and the resulting per-model, per-judge value is repeated across the α grid.

To check whether the apparent benefit of WRDO is driven by the per-prompt candidate bank alone or by the learned aggregation on top of it, we introduce the Uniform-avg WRD sanity-check baseline in this experiment. Uniform-avg WRD uses the same candidate bank of Section 2.1 as the learned WRDO instantiations, but fixes the coefficients to the uniform vector $w_i = 1/N$ without any optimization.

Discussion. First, in nearly every configuration the two learned WRDO instantiations (WRD-CMAES and WRD-GRPO-IS) outperform every prior single-direction baseline by a clear margin, establishing the superiority of the weighted-aggregation approach over the difference-of-means recipe, the Heretic toolkit and the gradient-learned RDO baseline. Second, the Uniform-avg WRD baseline is consistently dominated by the learned WRDO instantiations across the α grid, confirming that learning the aggregation coefficients w is what drives the gain, not the per-prompt bank alone.

Model	Method	$\alpha = 1.0$			$\alpha = 3.0$		
		WG	Q3G	LG3	WG	Q3G	LG3
Qwen3-8B-Base	No intervention	0.02	0.02	0.02	0.02	0.02	0.02
	Basic RD	0.22 $\pm$ 0.02	0.11 $\pm$ 0.03	0.44 $\pm$ 0.02	0.81 $\pm$ 0.03	0.80 $\pm$ 0.03	0.91 $\pm$ 0.04
	Heretic	0.77 $\pm$ 0.04	0.83 $\pm$ 0.05	0.81 $\pm$ 0.05	0.77 $\pm$ 0.04	0.83 $\pm$ 0.05	0.81 $\pm$ 0.05
	RDO	0.34 $\pm$ 0.05	0.41 $\pm$ 0.04	0.48 $\pm$ 0.05	0.34 $\pm$ 0.05	0.41 $\pm$ 0.04	0.48 $\pm$ 0.05
	Uniform-avg WRD	0.08 $\pm$ 0.001	0.25 $\pm$ 0.01	0.32 $\pm$ 0.02	0.002 $\pm$ 0.0003	0.89 $\pm$ 0.03	0.92 $\pm$ 0.02
	WRD-CMAES	0.88 $\pm$ 0.05	0.83 $\pm$ 0.02	0.89 $\pm$ 0.02	0.91 $\pm$ 0.05	0.99 $\pm$ 0.002	0.98 $\pm$ 0.002
	WRD-GRPO-IS	0.91 $\pm$ 0.07	0.94 $\pm$ 0.02	0.87 $\pm$ 0.02	0.93 $\pm$ 0.09	0.95 $\pm$ 0.02	0.96 $\pm$ 0.01
OLMo-3-1025-7B	No intervention	0.08	0.09	0.07	0.08	0.09	0.07
	Basic RD	0.41 $\pm$ 0.02	0.51 $\pm$ 0.03	0.43 $\pm$ 0.03	0.76 $\pm$ 0.02	0.72 $\pm$ 0.03	0.73 $\pm$ 0.02
	Heretic	0.76 $\pm$ 0.03	0.73 $\pm$ 0.04	0.70 $\pm$ 0.03	0.76 $\pm$ 0.03	0.73 $\pm$ 0.04	0.70 $\pm$ 0.03
	RDO	0.61 $\pm$ 0.04	0.67 $\pm$ 0.03	0.73 $\pm$ 0.03	0.61 $\pm$ 0.04	0.67 $\pm$ 0.03	0.73 $\pm$ 0.03
	Uniform-avg WRD	0.64 $\pm$ 0.02	0.66 $\pm$ 0.01	0.68 $\pm$ 0.02	0.77 $\pm$ 0.03	0.76 $\pm$ 0.02	0.79 $\pm$ 0.01
	WRD-CMAES	0.87 $\pm$ 0.02	0.86 $\pm$ 0.03	0.82 $\pm$ 0.02	0.88 $\pm$ 0.01	0.88 $\pm$ 0.02	0.85 $\pm$ 0.03
	WRD-GRPO-IS	0.82 $\pm$ 0.02	0.86 $\pm$ 0.03	0.79 $\pm$ 0.02	0.92 $\pm$ 0.02	0.97 $\pm$ 0.03	0.97 $\pm$ 0.02
Olmo-3-7B-Instruct	No intervention	0.003	0.003	0.004	0.003	0.003	0.004
	Basic RD	0.40 $\pm$ 0.02	0.42 $\pm$ 0.01	0.43 $\pm$ 0.03	0.53 $\pm$ 0.02	0.61 $\pm$ 0.03	0.59 $\pm$ 0.01
	Heretic	0.62 $\pm$ 0.03	0.58 $\pm$ 0.04	0.69 $\pm$ 0.04	0.62 $\pm$ 0.03	0.58 $\pm$ 0.04	0.69 $\pm$ 0.04
	RDO	0.40 $\pm$ 0.05	0.52 $\pm$ 0.04	0.53 $\pm$ 0.04	0.40 $\pm$ 0.05	0.52 $\pm$ 0.04	0.53 $\pm$ 0.04
	Uniform-avg WRD	0.49 $\pm$ 0.02	0.42 $\pm$ 0.01	0.45 $\pm$ 0.03	0.62 $\pm$ 0.02	0.67 $\pm$ 0.01	0.67 $\pm$ 0.01
	WRD-CMAES	0.92 $\pm$ 0.02	0.89 $\pm$ 0.01	0.93 $\pm$ 0.01	0.89 $\pm$ 0.02	0.93 $\pm$ 0.02	0.86 $\pm$ 0.01
	WRD-GRPO-IS	0.92 $\pm$ 0.01	0.88 $\pm$ 0.02	0.93 $\pm$ 0.02	0.93 $\pm$ 0.01	0.90 $\pm$ 0.02	0.88 $\pm$ 0.01

Table 1: Attack setup. Test ASR (in $[0, 1]$) on held-out harmful prompts, reported at the two extremes of the intensity grid, $\alpha \in \{1.0, 3.0\}$. Judges: WG = WildGuard, Q3G = Qwen3Guard-Gen-8B, LG3 = Llama-Guard-3-8B. Rows for α -independent methods are repeated across columns for layout consistency. **Bold** marks the best result in each column within each model block. Higher is a stronger attack. The intermediate intensities $\alpha \in \{1.5, 2.0\}$ are reported in Table 5 of Appendix D. Every intervened cell is the mean over three independent repetitions of the full pipeline, with the standard deviation over those repetitions. Cells that would round to 0.00 are reported at three decimals; none is an exact zero.

5.2 Concept cones and scale

Two comparisons fall outside the grid of Table 1 and are reported in Appendix I. Against the concept cone of Wollschläger et al. (2025), which treats refusal as a subspace rather than a line, the two are close: RDO-CONE leads by 3.3 ASR points on Qwen3-8B-Base, WRD-GRPO-IS by 5.5 and 8.5 on the two OLMo checkpoints. Repeating the attack setup on the 21B openai/gpt-oss-20b (OpenAI, 2025) reproduces the ranking with wider margins.

5.3 Utility preservation

Table 2 reports utility benchmarks for the same w^* that produced Table 1, per base model, at the two extremes of the intensity grid $\alpha \in \{1.0, 3.0\}$. Intermediate values $\alpha \in \{1.5, 2.0\}$ are deferred to Table 6 in Appendix E. In this experiment we restrict attention to the WildGuard-optimized runs only: w^* in every cell below is the coefficient vector obtained by optimizing against WildGuard. Accuracy is reported for ARC-easy (ARC-e), ARC-challenge (ARC-c), MMLU and GSM8K; word-level perplexity is reported for WikiText (PPL). As in Section 5.1, the Heretic entries are duplicated across the α columns because Heretic selects its own shift coefficient internally rather than consum-

ing α from the outside, so a single value per model and metric is repeated across the α grid for layout consistency.

Discussion. First, the prior single-direction baselines degrade general capabilities substantially at the larger values of α : accuracy on ARC and GSM8K collapses and WikiText perplexity grows by orders of magnitude. Heretic is partially protected, since its Bayesian search minimizes a KL divergence to the unintervened model, and it does hold perplexity near the clean value throughout. It is not uniformly utility-preserving, however: on Qwen3-8B-Base it costs 40 MMLU and 65 GSM8K points, so its KL target constrains the benign output distribution without protecting downstream benchmarks. Second, among the learned WRDO instantiations WRD-CMAES exhibits the same degradation pattern as the prior baselines: the Optuna target consumed by the sampler is the bare ASR-only signal without a KL term, so the optimizer is free to find directions that succeed at suppressing refusal but disrupt unrelated capabilities. Third, WRD-GRPO-IS preserves utility best because its surrogate of Equation 10 optimizes the KL-regularized objective of Equation 8, keeping the edit close to the clean policy even at high α .

Model	Method	$\alpha = 1.0$					$\alpha = 3.0$				
		ARC-e	ARC-c	MLLM	GSM8K	PPL	ARC-e	ARC-c	MLLM	GSM8K	PPL
Qwen3-8B-Base	No intervention	0.63 $\pm$ 0.01	0.38 $\pm$ 0.01	0.67 $\pm$ 0.004	0.84 $\pm$ 0.01	11.11	0.63 $\pm$ 0.01	0.38 $\pm$ 0.01	0.67 $\pm$ 0.004	0.84 $\pm$ 0.01	11.11
	Basic RD	0.59 $\pm$ 0.01	0.36 $\pm$ 0.01	0.64 $\pm$ 0.004	0.67 $\pm$ 0.01	13.81	0.26 $\pm$ 0.01	0.23 $\pm$ 0.01	0.23 $\pm$ 0.004	0.00 $\pm$ 0.00	> 10 ¹²
	Heretic	0.51 $\pm$ 0.01	0.30 $\pm$ 0.01	0.27 $\pm$ 0.004	0.19 $\pm$ 0.01	13.44	0.51 $\pm$ 0.01	0.30 $\pm$ 0.01	0.27 $\pm$ 0.004	0.19 $\pm$ 0.01	13.44
	RDO	0.63 $\pm$ 0.01	0.37 $\pm$ 0.01	0.67 $\pm$ 0.004	0.82 $\pm$ 0.01	11.16	0.63 $\pm$ 0.01	0.37 $\pm$ 0.01	0.67 $\pm$ 0.004	0.82 $\pm$ 0.01	11.16
	WRD-CMAES	0.53 $\pm$ 0.01	0.34 $\pm$ 0.01	0.67 $\pm$ 0.003	0.01 $\pm$ 0.003	161.18	0.37 $\pm$ 0.01	0.29 $\pm$ 0.01	0.51 $\pm$ 0.004	0.004 $\pm$ 0.002	> 10 ⁴
	WRD-GRPO-IS	0.63 $\pm$ 0.01	0.38 $\pm$ 0.01	0.65 $\pm$ 0.004	0.83 $\pm$ 0.01	11.15	0.64 $\pm$ 0.01	0.38 $\pm$ 0.01	0.62 $\pm$ 0.004	0.83 $\pm$ 0.01	11.35
OLMo-3-1025-7B	No intervention	0.79 $\pm$ 0.01	0.48 $\pm$ 0.015	0.62 $\pm$ 0.004	0.53 $\pm$ 0.014	62.5	0.79 $\pm$ 0.01	0.48 $\pm$ 0.015	0.62 $\pm$ 0.004	0.53 $\pm$ 0.014	62.5
	Basic RD	0.80 $\pm$ 0.01	0.48 $\pm$ 0.01	0.61 $\pm$ 0.004	0.53 $\pm$ 0.01	62.71	0.42 $\pm$ 0.01	0.26 $\pm$ 0.01	0.32 $\pm$ 0.004	0.0008 $\pm$ 0.0008	432.02
	Heretic	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.52 $\pm$ 0.01	62.61	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.52 $\pm$ 0.01	62.61
	RDO	0.80 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.56 $\pm$ 0.01	63.32	0.80 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.56 $\pm$ 0.01	63.32
	WRD-CMAES	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.52 $\pm$ 0.01	62.40	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.617 $\pm$ 0.004	0.51 $\pm$ 0.01	61.70
	WRD-GRPO-IS	0.79 $\pm$ 0.01	0.48 $\pm$ 0.015	0.62 $\pm$ 0.004	0.53 $\pm$ 0.014	62.5	0.78 $\pm$ 0.01	0.49 $\pm$ 0.015	0.61 $\pm$ 0.004	0.55 $\pm$ 0.014	62.21
Olm3-7B-Instruct	No intervention	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	100.83	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	100.83
	Basic RD	0.77 $\pm$ 0.01	0.48 $\pm$ 0.01	0.54 $\pm$ 0.004	0.67 $\pm$ 0.01	100.81	0.42 $\pm$ 0.01	0.31 $\pm$ 0.01	0.17 $\pm$ 0.004	0.01 $\pm$ 0.01	> 10 ⁵
	Heretic	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.54 $\pm$ 0.004	0.67 $\pm$ 0.01	101.27	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.54 $\pm$ 0.004	0.67 $\pm$ 0.01	101.27
	RDO	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.65 $\pm$ 0.01	101.34	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.65 $\pm$ 0.01	101.34
	WRD-CMAES	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.68 $\pm$ 0.01	101.19	0.74 $\pm$ 0.01	0.46 $\pm$ 0.01	0.50 $\pm$ 0.004	0.64 $\pm$ 0.01	118.19
	WRD-GRPO-IS	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	100.81	0.76 $\pm$ 0.01	0.47 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	100.49

Table 2: Utility preservation per base model, reported at $\alpha \in \{1.0, 3.0\}$. Higher is better for ARC-e, ARC-c, MMLU and GSM8K (accuracy as a fraction in $[0, 1]$); lower is better for PPL (WikiText word-level perplexity). Rows for α -independent methods are repeated across columns for layout consistency. Intermediate values $\alpha \in \{1.5, 2.0\}$ are reported in Table 6 of Appendix E.

Method	Qwen/Qwen3-8B				Olm3-7B-Instruct			
	1.0	1.5	2.0	3.0	1.0	1.5	2.0	3.0
No intervention	0.07	0.07	0.07	0.07	0.00	0.00	0.00	0.00
Basic RD	0.22	0.31	0.46	0.48	0.51	0.77	0.68	0.66
Heretic	0.87	0.87	0.87	0.87	0.77	0.77	0.77	0.77
RDO	0.39	0.39	0.39	0.39	0.20	0.20	0.20	0.20
WRD-CMAES	0.87	0.92	0.97	0.82	0.81	0.85	0.88	0.89
WRD-GRPO-IS	0.95	0.94	0.91	0.97	0.83	0.93	0.86	0.92

Table 3: HarmBench Standard ASR (in $[0, 1]$) computed with cais/HarmBench-Llama-2-13b-clb, varying α and model. **Bold** marks the best result in each column within each model block. Higher is a stronger attack.

5.4 HarmBench

We additionally re-evaluate the considered methods through an external benchmark that was not used during optimization, in order to corroborate the significance of the attack-setup results with a training-independent evaluator and to control for potential overfitting to the guard judges of Section 4.3. Because HarmBench is designed primarily as an evaluator of instruction-tuned models, in this setup we restrict attention to the instruct variants of our model pool, additionally including Qwen/Qwen3-8B (Team, 2025).

Table 3 reports HarmBench ASR. We adopt the same HarmBench evaluation setup as (Arditi et al., 2024): we take the prompt set from (Mazeika et al., 2024) and score the responses generated by the intervened model with the official HarmBench classifier cais/HarmBench-Llama-2-13b-clb. As in the utility experiment of Section 5.3, we restrict

this evaluation to the WildGuard-optimized runs: for every method the intervened model evaluated by HarmBench is the one whose coefficient vector was selected against WildGuard during training. HarmBench itself is never used during optimization and serves as an external evaluator that controls for judge-specific overfitting.

Discussion. The HarmBench evaluation confirms the superiority of the WRDO approach observed in Section 5.1: the learned WRDO instantiations attain the highest HarmBench ASR across the α grid. The agreement between the guard-judge metric and a classifier that was never used during optimization confirms that the WRDO advantage over the prior single-direction recipes is a real effect rather than an artifact of the training signal.

5.5 Defense setup

We additionally consider a defense setup that reverses the role of WRDO: instead of unlocking harmful behavior on an aligned model, we attempt to suppress it on a model whose refusal has already been broken. Notably, we operate on a third-party ablated checkpoint whose editing procedure is unknown to us, so we work exclusively from its weights without any knowledge of how the refusal removal was performed.

Table 4 reports the defense setup. As the ablated checkpoint we take the publicly released Orengeteng/Llama-3.1-8B-Lexi-Uncensored-V2, an uncensored derivative of Llama-3.1-8B-Instruct whose refusal behavior has already been removed by a third-party procedure.

Method	$\alpha = -1.0$			$\alpha = -1.5$			$\alpha = -2.0$			$\alpha = -3.0$		
	WG	Q3G	LG3	WG	Q3G	LG3	WG	Q3G	LG3	WG	Q3G	LG3
Abliterated (no defense)	0.85	0.83	0.84	0.85	0.83	0.83	0.85	0.83	0.84	0.85	0.83	0.84
Basic RD	0.84 ± 0.02	0.85 ± 0.03	0.82 ± 0.02	0.79 ± 0.03	0.84 ± 0.02	0.79 ± 0.03	0.80 ± 0.03	0.87 ± 0.02	0.87 ± 0.03	0.63 ± 0.02	0.68 ± 0.03	0.74 ± 0.02
WRD-CMAES	0.47 ± 0.05	0.56 ± 0.04	0.53 ± 0.06	0.011 ± 0.004	0.023 ± 0.005	0.073 ± 0.004	0.021 ± 0.005	0.053 ± 0.006	0.042 ± 0.004	0.014 ± 0.005	0.086 ± 0.006	0.079 ± 0.008
WRD-GRPO-IS	0.50 ± 0.08	0.52 ± 0.07	0.47 ± 0.09	0.014 ± 0.007	0.034 ± 0.008	0.047 ± 0.006	0.015 ± 0.007	0.014 ± 0.003	0.020 ± 0.005	0.019 ± 0.009	0.015 ± 0.003	0.022 ± 0.008

Table 4: Defense setup. Test ASR (in $[0, 1]$) on the abliterated checkpoint, across the sign-reversed intensity α and the three judges. Lower is a stronger defense. The three intervened rows are means over three independently optimized w^* , with the standard deviation over those runs; the Abliterated row receives no intervention and is reported from a single evaluation. **Bold** marks the strongest defense per column.

We apply the same WRDO pipeline to this checkpoint with the same α grid as in the attack setup, only with reversed sign. For Basic RD this coincides exactly with the defense recipe of Section 3.2 of (Arditi et al., 2024). The Heretic toolkit and RDO do not expose a defense mode and are therefore excluded from this experiment.

Discussion. The two learned WRDO instantiations clearly win the defense setup, driving the abliterated checkpoint’s ASR far below the no-defense baseline, while Basic RD fails to restore refusal. This shows that the proposed methods adapt to the alternative setup through a simple sign flip, with no change to the optimizer or the candidate bank.

5.6 Cross-judge transfer

A cross-judge transfer analysis (Appendix H) identifies WildGuard as the most reliable optimization signal among the considered guards: interventions optimized against WildGuard transfer well to the other two judges, while the reverse direction is less stable.

6 Conclusion

We introduced Weighted Refusal Direction Optimization (WRDO), a procedure that constructs a refusal direction by learning aggregation coefficients w over a bank of per-prompt directions, and we presented two derivative-free instantiations of the resulting black-box problem: a WRDO-tailored adaptation of separable CMA-ES and a proposed GRPO variant with importance sampling.

Across three open-weight models, three guard judges and the HarmBench classifier, the two WRDO instantiations exceed all prior single-direction baselines and transfer cleanly to the defense setup. Among them, WRD-GRPO-IS is the most reliable: it preserves the general capabilities of the intervened model substantially better than every other method considered, and the cross-judge analysis singles out WildGuard as the most reli-

able optimization signal, so we recommend WRD-GRPO-IS optimized against WildGuard as the default configuration for both auditing and defense.

7 Limitations

The full protocol of Sections 5.1–5.5 covers three open-weight checkpoints of comparable size, in the 7–8B parameter range.

A second limitation concerns the WRD-CMAES instantiation: its current implementation consumes the bare ASR-only objective of Equation 5 and does not yet support the KL regularizer of Equation 8. Extending the Optuna interface to expose a KL-aware sampling target is under active development, and we expect to incorporate this support in a future revision of the codebase.

8 Acknowledgments

The work of G. Evseev, D. Bylinkin, D. Medyakov, V. Aletov, A. Beznosikov was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-15-2025-013, dated June 20, 2025, IKG 000000C313925P4B0002).

9 Ethical Considerations

This work is inherently dual-use: methods for identifying and modifying refusal directions can help analyze and improve model safety, but they can also be used to weaken safeguards in aligned models. In particular, interventions that suppress refusal may increase the risk of harmful compliance in domains such as cyber abuse, privacy violations, disinformation, or other unsafe requests.

We therefore view WRDO primarily as a tool for controlled mechanistic analysis and safety evaluation. Our experiments are conducted offline on open-weight models and use automated guard judges, HarmBench-style evaluation, and utility benchmarks to study both safety and capability

effects. The goal is to better understand refusal representations, not to encourage unsafe deployment or misuse.

The broader implication of our findings is that refusal behavior may be concentrated in low-dimensional internal features and can therefore be brittle under targeted interventions. This suggests that refusal alone should not be treated as a complete safety mechanism; robust deployment requires layered defenses, including safer training, adversarial evaluation, monitoring, external classifiers, and system-level policy enforcement.

At the same time, the defense experiment shows a constructive use of the same machinery: WRDO-like methods may help diagnose and partially restore refusal behavior in modified or ablated checkpoints. Overall, we hope this work supports more robust alignment methods and encourages careful, transparent evaluation of both the risks and defensive uses of refusal-direction interventions.

References

- Mehmet Eren Ahsen, Robert M. Vogel, and Gustavo A. Stolovitzky. 2019. Unsupervised evaluation and weighted aggregation of ranked classification predictions. *Journal of Machine Learning Research*, 20(166):1–40.
- Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *KDD*.
- Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. [Refusal in language models is mediated by a single direction](#). *Preprint*, arXiv:2406.11717.
- 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, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.
- Patrick Chao, Edoardo DeBenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwal, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. 2024. JailbreakBench: An open robustness benchmark for jailbreaking large language models.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [The language model evaluation harness](#).
- Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs.
- Nikolaus Hansen. 2023. [The cma evolution strategy: A tutorial](#). *Preprint*, arXiv:1604.00772.
- Nikolaus Hansen and Andreas Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding.
- Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2023. Catastrophic jailbreak of open-source LLMs via exploiting generation.
- Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. LlamaGuard: LLM-based input-output safeguard for human-AI conversations.
- 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. 2024. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal.
- Mantas Mazeika, Andy Zou, Norman Mu, Long Phan, Zifan Wang, Chunru Yu, Adam Khoja, Fengqing Jiang, Aidan O’Gara, Zhen Xiang, Arezoo Rajabi, Dan Hendrycks, Radha Poovendran, Bo Li, and David Forsyth. 2023. [TDC 2023 \(LLM Edition\): The trojan detection challenge](#). NeurIPS 2023 Competition Track.
- OLMo Team. 2025. Olmo 3. ArXiv:2512.13961.
- OpenAI. 2025. [gpt-oss-120b and gpt-oss-20b model card](#). *Preprint*, arXiv:2508.10925.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.
- Qwen Team. 2025. Qwen3Guard technical report. ArXiv:2510.14276.
- Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. [Evolution strategies as a scalable alternative to reinforcement learning](#). *Preprint*, arXiv:1703.03864.
- Abdullah Abdul Sattar Shaikh, M. S. Bhargavi, and Pavan Kumar C. 2026. FedWAPR: Bridging theory and practice in probability-driven weighted aggregation for federated learning. *Information Sciences*, 723:122697.
- Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. 2024. A StrongREJECT for empty jailbreaks.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Andrey Veprikov, Vladimir Solodkin, Alexander Zyl, Andrey Savchenko, and Aleksandr Beznosikov. 2026. WeightLoRA: Keep only necessary adapters. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12419–12437.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does LLM safety training fail?
- Philipp Emanuel Weidmann. 2025. Heretic: Fully automatic censorship removal for language models. <https://github.com/p-e-w/heretic>.
- Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. 2014. Natural evolution strategies.
- Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. 2025. The geometry of refusal in large language models: Concept cones and representational independence. *arXiv preprint arXiv:2502.17420*.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models.

A Coefficient-Vector Constraints

The coefficients w_i in Equation 2 are signed real numbers, with no further normalization across coordinates beyond the per-layer L_2 renormalization already built into Equation 2. The feasible set $\mathcal{W}$ of Equation 6 is the per-coordinate box

$$\mathcal{W} = [w_{\min}, w_{\max}]^N, \quad (12)$$

with bounds $w_{\min}$ and $w_{\max}$ fixed as hyperparameters of the search and shared across all derivative-free methods of Section 3. We do not impose a simplex constraint, a unit-norm constraint or a top- k sparsity constraint: the box of Equation 12 is the only restriction on w . All derivative-free methods are initialized at the uniform-average point $w_i = 1/N$, which coincides with the Uniform-avg WRD sanity-check baseline of Section 5.1.

B Layerwise Intensity Schedule

The fixed layerwise schedule $\alpha^{(l)}$ of Equation 3 is adopted verbatim from the Heretic toolkit (Weidmann, 2025). Let $\ell^* = \text{max_weight_position} \cdot (L - 1)$ denote the absolute peak position in the stack, $\Delta = \text{min_weight_distance} \cdot (L - 1)$ the half-width of the active window, $\alpha_{\min} = \text{min_weight}$ the boundary value of the schedule, and $d_l = |l - \ell^*|/\Delta$ the normalized distance of layer l from the peak. The schedule is

$$\alpha^{(l)} = \begin{cases} \alpha + d_l (\alpha_{\min} - \alpha), & d_l \leq 1, \\ 0, & \text{otherwise,} \end{cases} \quad (13)$$

so that the intensity decays linearly from the peak value $\alpha = \text{max_weight}$ at ℓ^* to $\alpha_{\min}$ at the window boundary, and no intervention is applied to layers further than Δ from the peak. We follow the parameter naming of (Weidmann, 2025) verbatim. Throughout the paper we use $\text{max_weight_position} = 0.7$, $\text{min_weight_distance} = 0.3$ and $\alpha_{\min} = 0$, leaving $\alpha = \text{max_weight} \in \{1.0, 1.5, 2.0, 3.0\}$ as the only varied parameter.

C Data Splits and Sampling Seeds

Source of the splits. All harmful-prompt splits are loaded from the prepared dataset `erinya/dataset_splits` on the Hugging Face Hub, with the JSON files `splits/harmful_{train,val,test}.json` merged from the six public benchmarks listed in Section 4.1 after exact-prompt-string deduplication.

Harmful train sample for the bank. At bank-construction time we sample a fixed-size subset of the harmful train split to populate the candidate bank $\mathcal{R}$. The number of sampled prompts is $N = 128$ (ALL_CATEGORIES_HARMFUL_PROMPT_COUNT in the released code), and the sampling seed is 42 (ALL_CATEGORIES_HARMFUL_PROMPT_SEED). The same seed is reused across models and methods so that the candidate bank is built from the same $N = 128$ prompts in every run.

Harmful validation and test splits. Optimization uses the harmful validation split as the source of $\mathcal{D}_{\text{val}}^{\text{harm}}$ in Equation 4, and the harmful test split as $\mathcal{D}_{\text{test}}^{\text{harm}}$ for the held-out evaluations reported in Section 5. Both splits are consumed in order, with a model batch size of 32 per evaluation step (BATCH_SIZE in the released code).

Benign baseline. The harmless baseline $\mathcal{D}_{\text{benign}}$ used to compute $\mu_{\text{benign}}^{(l)}$ in Equation 1 is the first 400 prompts of the train split of mlabonne/harmless_alpaca. A separate 50-prompt subset of the test split of the same dataset is used as the benign batch for the KL penalty in Equation 8.

Optimizer seeds. The Optuna sampler seed is fixed at 42 (OPTUNA_SAMPLER_SEED). For WRD-GRPO-IS, behavior-policy perturbations $\varepsilon_m \sim \mathcal{N}(0, \sigma^2 I_N)$ are drawn from the active PyTorch RNG at training time, with σ controlled by GRPO_NOISE_SCALE.

D Attack Setup at Intermediate α

Table 5 complements Table 1 by reporting the attack setup of Section 5.1 at the intermediate intensities $\alpha \in \{1.5, 2.0\}$ omitted from the main text. Column conventions, the bold-per-column rule and the three-seed reporting follow Table 1 unchanged; because Heretic and RDO do not consume α , their rows repeat the values of Table 1. The reading of Section 5.1 carries over without exception: the two learned WRDO instantiations hold the best cell in all model-judge columns of this table, and Uniform-avg WRD remains below them everywhere, so neither conclusion of Section 5.1 depends on the two intensities kept in the main text.

E Utility Preservation at Intermediate α

Table 6 complements Table 2 by reporting the utility benchmarks at the intermediate intensity values

$\alpha \in \{1.5, 2.0\}$ omitted from the main text. Column conventions, the WildGuard-only restriction and the Heretic-duplication remark of Section 5.3 apply unchanged.

F Training Hyperparameters

This section documents the training-time hyperparameters of all six methods used in the main experiments. All values are reproduced in Table 7; the paragraphs below give a short narrative for each method. For the prior baselines (Basic RD, Heretic, RDO) we use the hyperparameter values from their original implementations without any additional tuning.

Basic RD. Basic RD performs no learned optimization. The only tunable knobs are the dataset filters used to construct the harmful/harmless splits and to select the seed difference-of-means direction: harmful prompts that the unintervened model already refuses are discarded if their refusal-induction score is below `induce_refusal_threshold`, candidate directions whose KL to the clean policy on the benign batch exceeds `kl_threshold` are dropped, and the bottom `prune_layer_percentage` of layers is excluded from the layer-search. All three thresholds are taken verbatim from the reference implementation of Arditi et al. (2024).

Heretic. Heretic runs an Optuna-based Bayesian search over the layerwise schedule of Equation 3. The search budget is `n_trials` with `n_startup_trials` random-initialization trials, and uses a KL constraint with scale `kl_divergence_scale` and target `kl_divergence_target`. Trial selection on the Pareto front prefers the minimum-KL trial whose refusal count is at most `refusal_threshold_count`. The search ranges over `max_weight` $\in [0.8, 1.5]$, `max_weight_position` relative to depth in $[0.6, 1.0]$, `min_weight` $\in [0.0, 1.0]$, `min_weight_distance` relative to depth in $[0.05, 0.6]$, the direction-index fraction in $[0.4, 0.9]$, and the schedule scope $\in \{\text{global, per-layer}\}$. We use the released Heretic toolkit out of the box, with the defaults of Weidmann (2025) for every hyperparameter listed above.

RDO. RDO trains a single per-layer direction by gradient descent with AdamW. The training loop runs for `epochs` steps with patience-based

Model	Method	$\alpha = 1.5$			$\alpha = 2.0$		
		WG	Q3G	LG3	WG	Q3G	LG3
Qwen3-8B-Base	No intervention	0.02	0.02	0.02	0.02	0.02	0.02
	Basic RD	0.33 $\pm$ 0.01	0.47 $\pm$ 0.01	0.67 $\pm$ 0.02	0.39 $\pm$ 0.03	0.87 $\pm$ 0.02	0.95 $\pm$ 0.03
	Heretic	0.77 $\pm$ 0.04	0.83 $\pm$ 0.05	0.81 $\pm$ 0.05	0.77 $\pm$ 0.04	0.83 $\pm$ 0.05	0.81 $\pm$ 0.05
	RDO	0.34 $\pm$ 0.05	0.41 $\pm$ 0.04	0.48 $\pm$ 0.05	0.34 $\pm$ 0.05	0.41 $\pm$ 0.04	0.48 $\pm$ 0.05
	Uniform-avg WRD	0.003 $\pm$ 0.0002	0.83 $\pm$ 0.02	0.81 $\pm$ 0.01	0.002 $\pm$ 0.0002	0.89 $\pm$ 0.01	0.85 $\pm$ 0.03
	WRD-CMAES	0.91 $\pm$ 0.05	0.98 $\pm$ 0.02	0.93 $\pm$ 0.02	0.93 $\pm$ 0.04	0.99 $\pm$ 0.01	0.98 $\pm$ 0.02
	WRD-GRPO-IS	0.92 $\pm$ 0.08	0.96 $\pm$ 0.02	0.93 $\pm$ 0.02	0.94 $\pm$ 0.09	0.95 $\pm$ 0.02	0.93 $\pm$ 0.03
OLMo-3-1025-7B	No intervention	0.08	0.09	0.07	0.08	0.09	0.07
	Basic RD	0.58 $\pm$ 0.02	0.56 $\pm$ 0.03	0.48 $\pm$ 0.03	0.61 $\pm$ 0.02	0.62 $\pm$ 0.02	0.69 $\pm$ 0.03
	Heretic	0.76 $\pm$ 0.03	0.73 $\pm$ 0.04	0.70 $\pm$ 0.03	0.76 $\pm$ 0.03	0.73 $\pm$ 0.04	0.70 $\pm$ 0.03
	RDO	0.61 $\pm$ 0.04	0.67 $\pm$ 0.03	0.73 $\pm$ 0.03	0.61 $\pm$ 0.04	0.67 $\pm$ 0.03	0.73 $\pm$ 0.03
	Uniform-avg WRD	0.69 $\pm$ 0.02	0.52 $\pm$ 0.02	0.59 $\pm$ 0.03	0.73 $\pm$ 0.02	0.73 $\pm$ 0.03	0.75 $\pm$ 0.01
	WRD-CMAES	0.84 $\pm$ 0.02	0.78 $\pm$ 0.03	0.81 $\pm$ 0.02	0.74 $\pm$ 0.02	0.84 $\pm$ 0.03	0.81 $\pm$ 0.02
	WRD-GRPO-IS	0.81 $\pm$ 0.02	0.85 $\pm$ 0.03	0.86 $\pm$ 0.03	0.89 $\pm$ 0.02	0.91 $\pm$ 0.02	0.96 $\pm$ 0.01
Olmo-3-7B-Instruct	No intervention	0.003	0.003	0.004	0.003	0.003	0.004
	Basic RD	0.49 $\pm$ 0.02	0.69 $\pm$ 0.01	0.52 $\pm$ 0.03	0.72 $\pm$ 0.02	0.87 $\pm$ 0.03	0.75 $\pm$ 0.02
	Heretic	0.62 $\pm$ 0.03	0.58 $\pm$ 0.04	0.69 $\pm$ 0.04	0.62 $\pm$ 0.03	0.58 $\pm$ 0.04	0.69 $\pm$ 0.04
	RDO	0.40 $\pm$ 0.05	0.52 $\pm$ 0.04	0.53 $\pm$ 0.04	0.40 $\pm$ 0.05	0.52 $\pm$ 0.04	0.53 $\pm$ 0.04
	Uniform-avg WRD	0.57 $\pm$ 0.02	0.53 $\pm$ 0.02	0.58 $\pm$ 0.03	0.54 $\pm$ 0.03	0.69 $\pm$ 0.02	0.61 $\pm$ 0.01
	WRD-CMAES	0.98 $\pm$ 0.02	0.96 $\pm$ 0.02	0.92 $\pm$ 0.01	0.92 $\pm$ 0.02	0.88 $\pm$ 0.03	0.93 $\pm$ 0.01
	WRD-GRPO-IS	0.93 $\pm$ 0.02	0.96 $\pm$ 0.02	0.95 $\pm$ 0.02	0.93 $\pm$ 0.01	0.93 $\pm$ 0.02	0.96 $\pm$ 0.01

Table 5: Attack setup at the intermediate intensities $\alpha \in \{1.5, 2.0\}$, complementing the main Table 1. Judges: WG = WildGuard, Q3G = Qwen3Guard-Gen-8B, LG3 = Llama-Guard-3-8B. Every intervened cell is a three-seed mean with the standard deviation over those repetitions; the Heretic and RDO rows repeat the values of Table 1 because neither method consumes α . **Bold** marks the best result in each column within each model block. Higher is a stronger attack.

early stopping and at most `n_lr_reduce` learning-rate reductions, at learning rate `lr` and effective batch size `effective_batch_size` accumulated from micro-batches of size `batch_size`. The composite loss is a weighted sum of an ablation term (`weight ablation_lambda`), an addition term (`addition_lambda`) and a retain term (`retain_lambda`). Target sequences for the loss are generated greedily up to `num_target_tokens` tokens in batches of `target_generation_batch_size`. All values match the reference implementation released by Wollschläger et al. (2025).

WRD-CMAES. WRD-CMAES uses the Optuna (Akiba et al., 2019) implementation of separable CMA-ES through the `CmaEsSampler` interface. The search variables are the coefficients $w \in \mathbb{R}^N$ of Section 2.2, constrained to the box $[w_{\min}, w_{\max}]^N$ of Appendix A. The total query budget is `n_trials`, the sampler seed is `sampler_seed`, the per-coordinate bounds are $w_{\min} = -2.0$ and $w_{\max} = 2.0$, and the parameterization is the scalar mode initialized at the uniform-average point ($w_i = 1/N$). All other sampler hyperparameters are left at library defaults.

WRD-GRPO-IS. WRD-GRPO-IS optimizes the surrogate of Equation 10 with AdamW at learning rate `learning_rate`. Each training step sam-

ples a group of `n_groups` behaviour policies $\{\mu_m\}$ from the Gaussian rollout of Equation 11 with noise scale $\sigma = \text{noise_scale}$, and a fresh group is drawn every epoch over `n_epochs` total epochs. The PPO/GRPO clip ratio is `clip_ratio` and the importance-sampling clip ratio is `is_clip_ratio`; loss is aggregated with the `loss_agg_mode` reduction. The KL penalty of Equation 8 is realized with coefficient $\beta = \text{beta}$. The reference-policy mixture weight is `ref_alpha`. The reward is the harmfulness score with the sign of Equation 6, and w is parameterized in the scalar mode initialized at the uniform-average point.

G Compute Resources

Every single run reported in this paper was conducted on a single NVIDIA H100 GPU with 80 GB of memory. Table 8 reports the approximate wall-clock runtime per method for the attack-setup experiments of Section 5.1.

H Cross-Judge Transfer

The learned WRDO instantiations consume a specific guard judge as their training signal, which raises the possibility that the resulting coefficient vector w^* exploits idiosyncrasies of that judge rather than capturing a transferable refusal-suppression direction. We probe this potential judge overfitting through a cross-judge transfer

Model	Method	$\alpha = 1.5$					$\alpha = 2.0$				
		ARC-e	ARC-c	MMLU	GSM8K	PPL	ARC-e	ARC-c	MMLU	GSM8K	PPL
Qwen3-8B-Base	No intervention	0.63 $\pm$ 0.01	0.38 $\pm$ 0.01	0.67 $\pm$ 0.004	0.84 $\pm$ 0.01	11.11	0.63 $\pm$ 0.01	0.38 $\pm$ 0.01	0.67 $\pm$ 0.004	0.84 $\pm$ 0.01	11.11
	Basic RD	0.42 $\pm$ 0.01	0.26 $\pm$ 0.01	0.23 $\pm$ 0.004	0.008 $\pm$ 0.002	113.98	0.29 $\pm$ 0.01	0.22 $\pm$ 0.01	0.23 $\pm$ 0.004	0.00 $\pm$ 0.00	> 10 ⁷
	Heretic	0.51 $\pm$ 0.01	0.30 $\pm$ 0.01	0.27 $\pm$ 0.004	0.19 $\pm$ 0.01	13.44	0.51 $\pm$ 0.01	0.30 $\pm$ 0.01	0.27 $\pm$ 0.004	0.19 $\pm$ 0.01	13.44
	RDO	0.63 $\pm$ 0.01	0.37 $\pm$ 0.01	0.67 $\pm$ 0.004	0.82 $\pm$ 0.01	11.16	0.63 $\pm$ 0.01	0.37 $\pm$ 0.01	0.67 $\pm$ 0.004	0.82 $\pm$ 0.01	11.16
	WRD-CMAES	0.63 $\pm$ 0.01	0.38 $\pm$ 0.01	0.64 $\pm$ 0.004	0.84 $\pm$ 0.01	11.25	0.51 $\pm$ 0.01	0.30 $\pm$ 0.01	0.47 $\pm$ 0.004	0.04 $\pm$ 0.005	64.84
	WRD-GRPO-IS	0.63 $\pm$ 0.01	0.38 $\pm$ 0.01	0.64 $\pm$ 0.004	0.84 $\pm$ 0.01	11.17	0.64 $\pm$ 0.01	0.38 $\pm$ 0.01	0.65 $\pm$ 0.004	0.85 $\pm$ 0.01	11.27
OLMo-3-1025-7B	No intervention	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.53 $\pm$ 0.01	62.5	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.53 $\pm$ 0.01	62.5
	Basic RD	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.55 $\pm$ 0.01	61.81	0.79 $\pm$ 0.01	0.47 $\pm$ 0.01	0.61 $\pm$ 0.004	0.50 $\pm$ 0.01	72.75
	Heretic	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.52 $\pm$ 0.01	62.61	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.52 $\pm$ 0.01	62.61
	RDO	0.80 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.56 $\pm$ 0.01	63.32	0.80 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.56 $\pm$ 0.01	63.32
	WRD-CMAES	0.79 $\pm$ 0.01	0.47 $\pm$ 0.01	0.62 $\pm$ 0.004	0.52 $\pm$ 0.01	62.39	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.51 $\pm$ 0.01	62.57
	WRD-GRPO-IS	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.52 $\pm$ 0.01	62.88	0.79 $\pm$ 0.01	0.48 $\pm$ 0.01	0.62 $\pm$ 0.004	0.53 $\pm$ 0.01	62.31
Olmo-3-7B-Instruct	No intervention	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	100.83	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	100.83
	Basic RD	0.75 $\pm$ 0.01	0.47 $\pm$ 0.01	0.53 $\pm$ 0.004	0.65 $\pm$ 0.01	102.96	0.59 $\pm$ 0.01	0.31 $\pm$ 0.01	0.42 $\pm$ 0.004	0.61 $\pm$ 0.01	> 10 ⁴
	Heretic	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.54 $\pm$ 0.004	0.67 $\pm$ 0.01	101.27	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.54 $\pm$ 0.004	0.67 $\pm$ 0.01	101.27
	RDO	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.65 $\pm$ 0.01	101.34	0.76 $\pm$ 0.01	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.65 $\pm$ 0.01	101.34
	WRD-CMAES	0.77 $\pm$ 0.01	0.47 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	101.19	0.76 $\pm$ 0.01	0.47 $\pm$ 0.01	0.53 $\pm$ 0.004	0.66 $\pm$ 0.01	103.42
	WRD-GRPO-IS	0.76 $\pm$ 0.008	0.48 $\pm$ 0.01	0.53 $\pm$ 0.004	0.67 $\pm$ 0.01	100.83	0.76 $\pm$ 0.008	0.49 $\pm$ 0.01	0.52 $\pm$ 0.004	0.67 $\pm$ 0.01	100.22

Table 6: Utility preservation per base model at the intermediate intensities $\alpha \in \{1.5, 2.0\}$, complementing the main Table 2. Higher is better for ARC-e, ARC-c, MMLU and GSM8K; lower is better for PPL.

analysis: w^* is optimized against each guard in turn and then re-evaluated against the other two, with the magnitude of the off-diagonal drop quantifying the extent to which performance is judge-specific.

Method	Approx. runtime, hours
Basic RD	~ 2
Heretic	~ 7
RDO	~ 3
Uniform-avg WRD	~ 2
WRD-CMAES	~ 4
WRD-GRPO-IS	~ 11

Table 8: Approximate wall-clock runtime per method on the attack-setup experiments of Section 5.1, on a single NVIDIA H100 80 GB GPU.

Table 9 reports the cross-judge transfer matrix obtained by optimizing w^* against one guard judge and then evaluating the resulting intervention against each of the three guard judges, for every value of $\alpha \in \{1.0, 1.5, 2.0, 3.0\}$. Large off-diagonal drops indicate that the optimizer has fit features specific to the training judge rather than a transferable refusal-suppression signal. The matrix is reported for the three judge-optimized methods; Basic RD, Heretic and RDO have no judge dependence and are omitted.

Discussion. The cross-judge analysis shows that learned WRDO interventions do not always transfer equally across guard judges. In several cases, optimizing against one judge yields high ASR under that same judge but noticeably weaker performance under another, indicating that part of the learned direction may depend on judge-specific decision boundaries rather than on a fully transferable

refusal-suppression signal.

The most pronounced asymmetry is observed for WildGuard: interventions that perform well under WildGuard tend to transfer relatively well to the other judges, while interventions optimized for other judges can fail to preserve high WildGuard ASR. This suggests that WildGuard provides the most reliable and demanding optimization signal among the considered guards.

By contrast, Llama-Guard-3 often reports lower ASR than the other judges. We interpret this cautiously: rather than necessarily indicating weaker attacks, this may reflect lower sensitivity of Llama-Guard-3 to harmful behavior in the evaluated generations, leading to more false negatives or inconsistent harmfulness detection. As a result, lower Llama-Guard-3 scores should not be treated as definitive evidence that the intervention is less effective.

I Concept Cones and Scale

Two comparisons fall outside the grid of Table 1. First, the closest prior work argues that refusal occupies a subspace rather than a line (Wollschläger et al., 2025), so we evaluate its concept-cone construction RDO-CONE (Section 3.1) on the same three models and test split. Second, all three checkpoints sit in the 7–8B range, so we repeat the attack setup on openai/gpt-oss-20b (OpenAI, 2025), a 21B mixture-of-experts model with 3.6B active parameters, leaving the pipeline of Section 5.1 unchanged. Both are scored by WildGuard and reported in Table 12.

Discussion. The two constructions perform

Method	Hyperparameter	Value
Basic RD	kl_threshold	0.1
	induce_refusal_threshold	0.0
	prune_layer_percentage	0.2
Heretic	n_trials / n_startup_trials	200 / 60
	kl_divergence_scale / _target	1.0 / 0.01
	refusal_threshold_count	0
	max_weight range	[0.8, 1.5]
	min_weight range	[0.0, 1.0]
	seed	42
RDO	optimizer	AdamW
	lr	10^{-2}
	batch_size / effective_batch_size	1 / 16
	epochs / patience / n_lr_reduce	1 / 5 / 2
	$(\lambda_{abl}, \lambda_{add}, \lambda_{ret})$	(1.0, 0.2, 1.0)
	num_target_tokens	8
	target_generation_batch_size	64
	seed	42
WRD-CMAES	sampler	CmaEsSampler (separable)
	n_trials / sampler_seed	50 / 42
	(w_{min}, w_{max})	(-2.0, +2.0)
	init / mode	uniform-avg / scalar
WRD-GRPO-IS	optimizer	AdamW
	learning_rate	10^{-3}
	n_groups (M)	8
	n_epochs	50
	σ (noise_scale)	1
	clip_ratio / is_clip_ratio	0.2 / 5.0
	β (KL weight)	1000.0
	ref_alpha	1.0

Table 7: Training hyperparameters for the six methods of Section 3. Values are read verbatim from the configuration files in the released codebase. Hyperparameters shared across methods (sampling seed 42, batch size 16 or 32, maximum response length, system prompt) are listed in Appendix C.

comparably, with the ordering depending on the checkpoint rather than favouring either consistently. Learned weighted aggregation is therefore competitive with an explicitly multi-dimensional cone while keeping the operator of Equation 3 rank-one per layer. We read this as evidence that the gain of WRDO comes from where the harmful prompts are allowed to disagree rather than from the dimensionality of the ablated subspace.

β	1	10	10^2	10^3	10^4
ASR (WG)	0.95	0.92	0.91	0.91	0.78

Table 10: Attack side of the β sweep. Test ASR under WildGuard for WRD-GRPO-IS on Qwen3-8B-Base at $\alpha = 1.0$. Higher is a stronger attack.

J External Evaluation of the Defense Setup

Table 11 re-scores the defended checkpoints of Section 5.5 with the external HarmBench classi-

fier, which plays no part in the optimization, and therefore complements the three guard judges of Table 4. At $\alpha \leq -1.5$ both learned instantiations hold HarmBench ASR at or below 0.03 against 0.94 for the undefended checkpoint, whereas Basic RD stays above 0.90 until $\alpha = -3.0$, an intensity at which its utility has already collapsed in the attack direction (Table 2).

K Sensitivity to the KL Coefficient β

The KL coefficient β of Equation 8 is the single hyperparameter that sets where WRD-GRPO-IS sits on the attack-versus-utility trade-off, so the value reported throughout the paper is not an arbitrary default. We select it by sweeping $\beta \in \{1, 10, 10^2, 10^3, 10^4\}$ on Qwen3-8B-Base at $\alpha = 1.0$, with everything else held at the values of Table 7, and reading ASR and utility off the same run. Tables 10 and 13 report both sides of the sweep. The selection is made once, on this model, and the resulting value is reused unchanged for every other model and judge in the paper; we

Model	Method	Opt. judge	$\alpha = 1.0$			$\alpha = 1.5$			$\alpha = 2.0$			$\alpha = 3.0$		
			WG	Q3G	LG3	WG	Q3G	LG3	WG	Q3G	LG3	WG	Q3G	LG3
Qwen3-8B-Base	WRD-CMAES	WG	0.89	0.89	0.92	0.91	0.92	0.95	0.95	0.96	0.96	0.90	0.93	0.92
		Q3G	0.00	0.84	0.97	0.00	0.98	0.96	0.00	1.00	0.97	0.00	1.00	0.97
		LG3	0.00	0.73	0.88	0.02	0.94	0.93	0.16	0.96	0.98	0.01	0.98	0.98
	WRD-GRPO-IS	WG	0.91	0.94	0.90	0.93	0.95	0.92	0.95	0.96	0.94	0.95	0.96	0.93
		Q3G	0.91	0.94	0.90	0.94	0.95	0.92	0.01	0.96	0.98	0.94	0.95	0.93
		LG3	0.87	0.92	0.87	0.94	0.96	0.92	0.36	0.42	0.39	0.91	0.95	0.97
OLMo-3-1025-7B	WRD-CMAES	WG	0.86	0.94	0.50	0.83	0.75	0.20	0.74	0.70	0.35	0.89	0.89	0.49
		Q3G	0.85	0.86	0.55	0.77	0.79	0.53	0.82	0.83	0.38	0.94	0.87	0.46
		LG3	0.65	0.69	0.82	0.63	0.65	0.80	0.69	0.74	0.80	0.91	0.91	0.85
	WRD-GRPO-IS	WG	0.83	0.76	0.42	0.80	0.73	0.40	0.89	0.79	0.50	0.91	0.89	0.43
		Q3G	0.62	0.86	0.30	0.65	0.84	0.32	0.85	0.91	0.45	0.92	0.97	0.50
		LG3	0.60	0.65	0.79	0.65	0.70	0.85	0.80	0.85	0.95	0.88	0.92	0.96
Olmo-3-7B-Instruct	WRD-CMAES	WG	0.91	0.91	0.95	0.98	0.96	0.97	0.92	0.95	0.93	0.88	0.83	0.76
		Q3G	0.17	0.89	0.82	0.29	0.95	0.96	0.62	0.87	0.86	0.76	0.93	0.88
		LG3	0.17	0.96	0.92	0.22	0.97	0.91	0.59	0.94	0.92	0.79	0.88	0.85
	WRD-GRPO-IS	WG	0.90	0.95	0.21	0.93	0.90	0.47	0.93	0.98	0.58	0.93	0.97	0.72
		Q3G	0.23	0.88	0.78	0.35	0.96	0.80	0.58	0.92	0.90	0.78	0.91	0.80
		LG3	0.22	0.30	0.91	0.30	0.42	0.94	0.45	0.58	0.95	0.80	0.85	0.87

Table 9: Cross-judge transfer. Test ASR (in $[0, 1]$) when w^* is optimized against the row judge and evaluated against the column judge, reported per base model, for every $\alpha \in \{1.0, 1.5, 2.0, 3.0\}$. Within each α block, diagonal cells coincide with the corresponding entries of Table 1 or, for $\alpha \in \{1.5, 2.0\}$, of Table 5.

Method	-1.0	-1.5	-2.0	-3.0
Abliterated (no defense)	0.94	0.94	0.94	0.94
Basic RD	0.94	0.92	0.91	0.31
WRD-CMAES	0.47	0.03	0.00	0.03
WRD-GRPO-IS	0.47	0.02	0.00	0.00

Table 11: Defense setup evaluated externally. HarmBench Standard ASR computed with cais/HarmBench-Llama-2-13b-cls on the ablit-erated checkpoint, across the sign-reversed intensity α . HarmBench is never used during optimization. Lower is a stronger defense.

Model	Method	ASR
Qwen3-8B-Base	RDO-CONE	0.953
	WRD-GRPO-IS	0.93 ± 0.09
OLMo-3-1025-7B	RDO-CONE	0.865
	WRD-GRPO-IS	0.92 ± 0.02
Olmo-3-7B-Instruct	RDO-CONE	0.845
	WRD-GRPO-IS	0.93 ± 0.01
gpt-oss-20b	No intervention	0.001
	Basic RD	0.308
	Heretic	0.772
	WRD-CMAES	0.923
	WRD-GRPO-IS	0.931

Table 12: Concept-cone comparison. Test ASR under WildGuard; WRDO rows are taken at $\alpha = 3.0$. The WRD-GRPO-IS rows of the upper blocks repeat the corresponding cells of Table 1 and carry its three-seed spread; the RDO-CONE and gpt-oss-20b rows are single runs. Higher is a stronger attack.

β	ARC-e	ARC-c	MMLU	GSM8K	PPL
1	0.52	0.34	0.25	0.38	261.32
10	0.51	0.36	0.34	0.43	260.13
10^2	0.60	0.36	0.59	0.67	150.72
10^3	0.63	0.38	0.65	0.83	11.15
10^4	0.63	0.38	0.65	0.84	11.12

Table 13: Utility side of the β sweep, same runs as Table 10. Higher is better for the four accuracies, lower is better for WikiText perplexity. Standard errors are 0.01 for ARC-e, ARC-c and GSM8K and 0.004 for MMLU. **Bold** marks the best value per column.

did not tune β per model.

The two sides of the sweep move in opposite directions, but not at the same rate. ASR falls gradually as β grows, staying almost flat over the middle of the range before dropping off at the largest value. Utility does not track it: it stays poor while the penalty is weak, with MMLU and GSM8K far below the clean model and perplexity an order of magnitude above it, and then recovers abruptly in a single step to the unintervened level. The two regimes overlap in one place, and that is where we operate: $\beta = 10^3$ is the only setting that combines an ASR at the top of the flat region with utility indistinguishable from the clean checkpoint, and it is the value used throughout the paper.