DUAL-NULL LORA: A TRAINING DYNAMICS APPROACH TO PRESERVING SAFETY ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

The safety alignment of large language models, typically established as part of post-training pipelines, is known to degrade under subsequent fine-tuning. This phenomenon occurs not only when fine-tuning on harmful examples but also when using harmless, domain-specific utility data. We use recent developments in the theory of deep learning training dynamics to understand this phenomenon and derive a mitigation for the special case of *LoRA* fine-tuning. In a first-order approximation of fine-tuning dynamics, gradient steps on potentially harmful training examples change the probability of obtaining an aligned response to any other prompt. This change is mediated by the empirical neural tangent kernel (eNTK), which couples each training example to each observed prompt. We introduce *dual-null LoRA*, a standard *LoRA* adapter built to eliminate this coupling: Our adapter is initialized in approximate null spaces of both model activations and gradients, and every update during training is projected back into these null spaces. Evaluated on two base models and three fine-tuning datasets (*SST-2*, *AG News*, *GSM8K*), *dual-null LoRA* effectively keeps harmfulness (HS) and attack-success (ASR) at the levels of the pre-fine-tuning baselines without negatively impacting task learning or general capability, even at 20% harmful training examples.

1 INTRODUCTION

The term safety alignment[1] refers to the extent to which a model's responses conform to the intentions, values, and norms of users, providers, and society at large. An aligned model follows instructions, avoids toxic content, and declines requests for harmful assistance. Such refusals prevent bad actors from using the model's capabilities for illegitimate purposes. Alignment is typically established after pretraining, by fine-tuning on curated demonstrations and optimizing the model on preference signals (e.g., by means of *DPO* (Rafailov et al., 2023) or *RLHF* (Ouyang et al., 2022)).

Further domain-specific fine-tuning can provide a cost-effective way to adapt a deployed model to a utility task. In such settings, training continues on a small task-specific dataset. Because fine-tuning models at frontier scale would exceed the compute and memory available to most users, and because releasing the weights would expose the provider's intellectual property, some providers offer fine-tuning as a service (*FTaaS*), where the user supplies the data but the provider runs the fine-tuning procedure on the closed weights (Huang et al., 2026a). For open-weight models, in contrast, users fine-tune the released weights themselves, with full control over data and procedure.

Alignment has been shown to erode under fine-tuning: Small fractions of harmful examples mixed into otherwise benign task data can remove refusal almost entirely (Qi et al., 2024; Yang et al., 2024b; Zhan et al., 2024; Lermen & Rogers-Smith, 2024), and even purely benign data can degrade alignment as a side effect (Qi et al., 2024). Fine-tuning thus presents a channel through which alignment can be destroyed deliberately and *accidentally*.

Threat model. The mitigations we propose assume a benign actor who could adopt our method voluntarily to prevent accidental erosion of alignment, or an FTaaS setting where an attacker merely controls the training data and the model provider controls the training procedure.

[1] In a slight abuse of terminology, this paper uses the terms *alignment* and *safety alignment* interchangeably.

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.

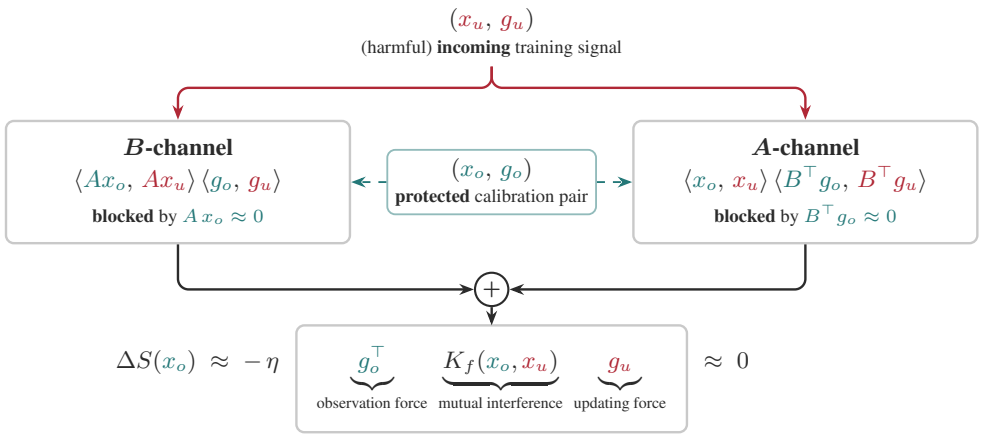

Figure 1: The *dual-null* mechanism. An incoming fine-tuning pair (x_u, g_u) (red) couples to the protected pair (x_o, g_o) (blue) through an adapter kernel (Eq. 10). Each *dual-null* condition blocks a factor of one contributing channel, such that the first-order change of the alignment score vanishes.

Building on the work of Ren & Sutherland (2025), Huang et al. (2026b) have provided a mechanistic account of the effects of fine-tuning on alignment, from which potential mitigations, such as our *dual-null LoRA*, can be derived in a principled manner. Following their work, a gradient step on a potentially attacker-chosen update prompt x_u moves the alignment score at any observation prompt x_o (for example a safety-sensitive prompt). For the special case of *LoRA* fine-tuning (Hu et al., 2022), where an adapted layer computes $f(x) = Wx + s\,BAx$ with trainable factors A and B, this effect can be expressed as the sum of two channels (Figure 1): The update of B moves the alignment score in proportion to $\langle Ax_o, Ax_u\rangle\langle g_o, g_u\rangle$ (B-channel), that of A in proportion to $\langle x_o, x_u\rangle\langle B^\top g_o, B^\top g_u\rangle$ (A-channel), with x the layer inputs and g_o, g_u the layer-output gradients of alignment score and training loss (Section 4).

Existing null-space methods for *LoRA* fine-tuning (e.g., *LoRA-Null* (Tang et al., 2026)) constrain A so that $Ax_o \approx 0$, which closes the B-channel but leaves $B^\top g_o$ unconstrained. Training examples where inputs overlap with those of the protected prompts can then still erode safety through the A-channel. Our proposed mitigation, *dual-null LoRA*, additionally constrains B so that $B^\top g_o \approx 0$. Both conditions are imposed at initialization from a small safety calibration set and maintained during training.

Contributions. We derive a defence against harmful *LoRA* fine-tuning building on the learning-dynamics decomposition of Ren & Sutherland (2025) and its alignment-dynamics form as derived by Huang et al. (2026b). The *dual-null LoRA* adapter is designed to approximately remove the first-order change ΔS that fine-tuning induces on the safety score. **Thus, it protects alignment behaviour installed in earlier training stages during subsequent fine-tuning.**

We benchmark this defence on two released instruction-tuned models, *Qwen2.5-3B-Instruct* (Qwen Team, 2024; Yang et al., 2024a) and *Gemma 2 IT 2B* (Gemma Team, 2024), and three fine-tuning datasets (*SST-2* (Socher et al., 2013), *AG News* (Zhang et al., 2015), and *GSM8K* (Cobbe et al., 2021)). At increasing poison levels (the fraction of harmful training examples mixed into the fine-tuning data) between zero and 20%, we compare the efficacy of *dual-null LoRA* with established mitigations from the literature (*SafeInstr* (Bianchi et al., 2024), *Safe LoRA* (Hsu et al., 2024), *AsFT* (Yang et al., 2025), *GuardSpace* (Zhang et al., 2026), and *SaLoRA* (Li et al., 2025)). We find dual-null LoRA competitive with the strongest of these defences: Even at 20% harmful training examples, it remains effectively at pre-fine-tuning alignment levels on held-out test sets created from *BeaverTails* and *AdvBench*, without significantly impacting overrefusal, general capability, or task learning.

2 Related Work

Defences across the training pipeline. Safety alignment can be built in at the very early stages of training: By conditioning pretraining on control tokens that discriminate safe and undesirable content (Korbak et al., 2023), and by curating the corpus, adding synthetic refusal data, moral-education data,

or interleaved safety verdicts (Maini et al., 2025; Li et al., 2026). An additional option is to filter hazardous knowledge out of the pretraining corpus entirely as in O'Brien et al. (2026).

Many defences attempt to harden the post-training safety alignment stage: Qi et al. (2025) attempt to *deepen* refusal with demonstrations that begin like compliance and then recover into a refusal. *Vaccine* and its variants run the safety SFT under adversarial perturbations of the hidden representations, such that the installed refusal behaviour becomes insensitive to later representation drift (Huang et al., 2024b; Liu et al., 2025). *Booster* and *TAR* simulate fine-tuning attacks, penalizing the harmful loss reduction of a simulated harmful gradient step or meta-learning weights on which simulated attacks fail (Huang et al., 2025b; Tamirisa et al., 2025). *RepNoise* degrades the internal representations of harmful content by pushing them towards noise (Rosati et al., 2024).

During fine-tuning itself, some defences augment the fine-tuning set with safety demonstrations (*SafeInstr*, Bianchi et al., 2024), alternate between optimizing on safety and task data while penalizing the weight drift between the two (Huang et al., 2024a), or filter and reweight the training examples through per-example or token-level weights (Shen et al., 2025; Hu et al., 2025; Peng et al., 2025).

After the fine-tuning run, some post-hoc repairs prune the weights most associated with harmful outputs (Huang et al., 2025a), add a weight perturbation optimized to raise the harmful loss while preserving task performance (Wang et al., 2025), or re-add the weight difference between an aligned reference model and its unaligned counterpart (Bhardwaj et al., 2024).

Our defence is a fine-tuning stage intervention that leaves the training data, the loss, and the preceding stages unchanged: It restricts the directions in parameter space that an update may occupy, and can in principle be combined with the interventions above.

Defences built on the learning-dynamics factorization. A small and recent group of methods shares the theoretical machinery of this paper and turns the factorization of Equation 3 into a tool to mitigate alignment erosion. Huang et al. (2026b) derive an explicit erosion force whose rebound component may explain erosion under benign fine-tuning. As a mitigation they propose a data augmentation technique to increase *refusal diversity*, arguing that this will reduce the rebound force. *Antibody* (Nguyen et al., 2026) extends the decomposition of Ren & Sutherland (2025) to mini-batches and uses it to motivate a weighting scheme that reduces the influence of harmful training examples during fine-tuning. Examples whose target completion the current model deems less likely than a generic refusal are assigned a relatively lower weight within their batch. This fine-tuning stage intervention is preceded by an alignment stage intervention that flattens the harmful loss in a neighbourhood of the installed weights, such that harmful training examples emit small gradients when fine-tuning begins. Lu et al. (2026) attribute overrefusal to high kernel couplings between toxic and only *seemingly* toxic prompts. As a mitigation they propose a contrastive stage before alignment SFT separating the activations of the two prompt classes, lowering the coupling.

Compared to the methods above, *dual-null LoRA* builds on the kernel component of Equation 3 leveraging the parametrization of the fine-tuning update: Its constraints make the kernel row at every protected prompt vanish to first order, for any update example and objective.

Defences restricting the *LoRA* update space. Interventions closest to *dual-null LoRA* constrain where in parameter space a *LoRA* update may live. *SaLoRA* (Li et al., 2025) collects the layer outputs on harmful prompts with safe responses, takes the top left singular vectors as safety-feature directions, and freezes a projector in front of the adapter output that annihilates the adapter's write access to these directions for every input and any trained factors. Once, after fine-tuning, *Safe LoRA* (Hsu et al., 2024) replaces the trained update of a module by its image under an alignment-matrix operator, built from the weight difference between an aligned and an unaligned model, whenever the update's alignment with that subspace falls below a threshold. *AsFT* (Yang et al., 2025) enforces the same restriction during training, penalizing the component of the adapter update outside the alignment-difference subspace. The approaches rest on different assumptions: *SaLoRA* starts from the premise that safety is carried by identifiable feature directions in the *layer-output space*, so an update that cannot write into these directions leaves the safety behaviour intact. *Safe LoRA* (and similarly *AsFT*) instead claims that the *weight difference* between the aligned and the unaligned model spans a safety-relevant subspace, and that an update deviating from it is the source of safety loss. The closest constructions to ours, *LoRA-Null* (Tang et al., 2026) and *GuardSpace* (Zhang et al., 2026), build null-space constraints

for the *LoRA* update from calibration activations. Appendix C develops a more detailed comparison. Neither of these methods derives its constraint from the evolution law of Equation 3.

3 PRELIMINARIES

Learning dynamics of fine-tuning. Following Ren & Sutherland (2025), it is possible to derive how a single fine-tuning gradient step on a training example moves the model's predictions on any other input to first order. Throughout, (x_u, y_u) denotes the training (update) example on which the gradient is computed, and x_o an input on which the model's behaviour is observed. For example, the update example may be a training example for a fine-tuning task while the observation input could be a prompt asking for assistance with a harmful task. For simplicity of exposition, the following statements assume the prediction of a single next token. We state a sequence prediction version of the main result at the end of the section. For cross-entropy training, the loss gradient in logit space is the residual $G(x_u) = \pi_u - p_u$, a vector that points from the supervisory signal p_u to the model's current prediction π_u. To first order, the gradient step moves the logits at the observation input by

$$\Delta z(x_o) = -\eta\, K(x_o, x_u)\, G(x_u) + O(\eta^2) \tag{1}$$

where $K(x_o, x_u) = \nabla_\theta z(x_o) \nabla_\theta z(x_u)^\top$ is the empirical neural tangent kernel (eNTK) (Jacot et al., 2018). Composing with the softmax Jacobian $J(x_o)$ yields the one-step decomposition

$$\Delta\pi(x_o) = -\eta\, J(x_o)\, K(x_o, x_u)\, G(x_u) + O(\eta^2) \tag{2}$$

Alignment dynamics. Fix an observation prompt x_o and the set $\mathcal{A}$ of aligned completions to it. The alignment score at x_o is the probability that the model's generation lands in the aligned set, in the single-token case $S(x_o) = \mathbf{1}_{\mathcal{A}}^\top \pi(x_o)$. Substituting Equation 2 for $\Delta\pi(x_o)$ in $\Delta S(x_o) = \mathbf{1}_{\mathcal{A}}^\top \Delta\pi(x_o)$ and absorbing $\mathbf{1}_{\mathcal{A}}^\top J(x_o)$ into the score gradient $\nabla_{z_o} S(x_o)^\top$ gives the evolution law of Huang et al. (2026b)

$$\Delta S(x_o) = -\eta\, \nabla_{z_o} S(x_o)^\top\, K(x_o, x_u)\, G(x_u) + O(\eta^2) \tag{3}$$

Sequence prediction. In the sequence prediction setting, the same derivation applies per token position. Write χ_u for the update prompt concatenated with its response, and χ_o for the observation prompt concatenated with a fixed observed response. The per-position losses sum, one eNTK block $K_{m,l}(\chi_o, \chi_u) = \nabla_\theta z_m(\chi_o)\, \nabla_\theta z_l(\chi_u)^\top$ couples each observed position m to each update position l, and the evolution law becomes

$$\Delta S(\chi_o) = -\eta \sum_{m,l} c_m(\chi_o)^\top\, K_{m,l}(\chi_o, \chi_u)\, G_l(\chi_u) + O(\eta^2) \tag{4}$$

$$c_m(\chi_o) := \pi_m^{<}\, \pi_{S,m}^{+}\, \pi_{S,m}^{-} \big(\tilde{\pi}_m^{+} - \tilde{\pi}_m^{-}\big)$$

where every factor of $c_m(\chi_o)$ is evaluated at χ_o, with $\pi_m^{<}$ the probability of reaching position m's prefix, $\pi_{S,m}^{\pm}$ the probabilities of staying aligned or not at position m, and $\tilde{\pi}_m^{\pm}$ the model's aligned and non-aligned next-token conditionals given that prefix (Huang et al., 2026b). $\Delta S(\chi_o)$ is thus the first-order change, caused by the gradient step on χ_u, in the model's probability of staying on aligned continuations along the recorded response to x_o.

Low-rank adaptation. Full fine-tuning updates every parameter of the model, which at the scale of modern language models can quickly exceed common memory and storage budgets of fine-tuning users. Low-rank adaptation (*LoRA*) lessens this problem by freezing the base model weights and training a small set of *additional* parameters (Hu et al., 2022). For a pretrained weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, *LoRA* constrains the update to a product of two matrices, so the adapted layer computes

$$f(x) = Wx + s\, BAx, \qquad s = \alpha/r \tag{5}$$

with $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, and rank $r \ll \min(d_{\text{out}}, d_{\text{in}})$. Every product BA has rank at most r, and the trainable parameter count drops from $d_{\text{out}} \cdot d_{\text{in}}$ to $r(d_{\text{out}} + d_{\text{in}})$. The prefactor s scales the adapter's contribution to the forward pass. At initialization the adapter is usually chosen to leave the model unchanged, such that training departs from the unadapted base model. During fine-tuning only A and B receive gradients, while W stays frozen, and at inference time the update can be merged once into the pretrained matrix. While the original work adapts only the attention query and value projections, later practice also attaches adapters to all linear projections of both the attention and the MLP blocks (Dettmers et al., 2023).

4 METHOD: THE *dual-null LoRA* DEFENCE

The eNTK under *LoRA* adaptation. Restrict the trainable parameters to a *LoRA* adapter and assume for simplicity that only a single layer is adapted, computing $f(x) = Wx + sBAx$ as in Equation 5, with frozen W and trainable $\theta = (A, B)$. Because θ enters the logits only through the layer output, the chain rule gives $\nabla_\theta z(x) = D(x)\,\nabla_\theta f(x)$ with $D(x) := \partial z(x)/\partial f(x)$ the Jacobian of the logits with respect to the layer output. The eNTK of Equation 1 therefore factorizes as

$$K(x_o, x_u) = D(x_o)\,K_f(x_o, x_u)\,D(x_u)^\top \qquad K_f(x_o, x_u) := \nabla_\theta f(x_o)\,\nabla_\theta f(x_u)^\top \tag{6}$$

It remains to compute the layer kernel K_f. Differentiating the layer output entrywise with respect to the adapter parameters gives

$$\frac{\partial f(x)_k}{\partial A_{ij}} = s\,B_{ki}\,x_j, \qquad \frac{\partial f(x)_k}{\partial B_{mp}} = s\,\delta_{km}\,(Ax)_p \tag{7}$$

The kernel entry $K_f(x_o, x_u)_{kl}$ is the inner product of the parameter gradients of $f(x_o)_k$ and of $f(x_u)_l$. The parameter vector is the disjoint concatenation of the entries of A and of B. An inner product over a concatenation is the sum of the inner products over its blocks, so the entry splits into an A-part and a B-part with no mixed terms. Substituting Equation 7 into the two parts and returning to matrix form yields the layer kernel

$$K_f(x_o, x_u) = s^2\big(\langle Ax_o, Ax_u\rangle\, I_{d_{\text{out}}} + \langle x_o, x_u\rangle\, BB^\top\big) \tag{8}$$

The *dual-null* conditions. Absorbing the outer Jacobians into the *pulled-back* directions

$$g_o := D(x_o)^\top \nabla_{z_o} S(x_o) \qquad g_u := D(x_u)^\top G(x_u) \tag{9}$$

turns the first-order term of Equation 3 into

$$g_o^\top K_f(x_o, x_u)\,g_u = s^2\big(\underbrace{\langle Ax_o, Ax_u\rangle\,\langle g_o, g_u\rangle}_{B\text{-channel}} + \underbrace{\langle x_o, x_u\rangle\,\langle B^\top g_o, B^\top g_u\rangle}_{A\text{-channel}}\big) \tag{10}$$

Equation 10 exposes the attack surface of an adapter whose B-channel alone is constrained: With $Ax_o = 0$ the B-channel vanishes, but an uncontrolled $B^\top g_o$ leaves the A-channel open, and an attacker can attempt to erode the alignment score through the overlaps $\langle x_o, x_u\rangle$ and $\langle B^\top g_o, B^\top g_u\rangle$, both of which respond to the attacker's choice of update examples. For the defender, the fine-tuning data, and with it x_u and g_u, is unknown in advance. However, the defence can attempt to achieve $g_o^\top K_f(x_o, x_u) = 0$ for every update example x_u. This suggests the following two conditions for every protected pair (x_o, g_o):

$$\textbf{(DN1)}\;\; A\,x_o = 0 \qquad \text{and} \qquad \textbf{(DN2)}\;\; B^\top g_o = 0 \tag{11}$$

We call these the *dual-null* conditions. If both conditions are met, $\Delta S(x_o) \approx 0$ to first order, for every update example. Appendix A gives a more detailed derivation, including extensions to multiple adapted layers and to full sequences.

Calibration using approximate null spaces. Both *dual-null* conditions should hold, at least approximately, at initialization and throughout fine-tuning. To this end, we record protected pairs from a calibration set of safety demonstrations, harmful prompts paired with aligned refusal responses. One forward and one backward pass of the cross-entropy loss per demonstration yield, at every adapted layer, the activations $x_{o,1}, \ldots, x_{o,N}$ entering the layer and the loss gradients $g_{o,1}, \ldots, g_{o,N}$, which we pool into the uncentered covariances

$$C_{\text{in}} = \sum_{n=1}^{N} x_{o,n}\,x_{o,n}^\top, \qquad C_{\text{out}} = \sum_{n=1}^{N} g_{o,n}\,g_{o,n}^\top \tag{12}$$

We collect the trailing eigenvectors of C_{in} whose eigenvalue mass is at most a budget b of its trace as $U_{\text{in}} \in \mathbb{R}^{d_{\text{in}} \times k_{\text{in}}}$, analogously those of C_{out} as $V_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times k_{\text{out}}}$, constraining the rows of A to $\text{span}(U_{\text{in}})$ and the columns of B to $\text{span}(V_{\text{out}})$. Any such factor obeys

$$\|A\,x_{o,n}\| \le \|A\|_2\,\|U_{\text{in}}^\top x_{o,n}\| \le \|A\|_2\Big(\sum_{i \le k_{\text{in}}} \lambda_i(C_{\text{in}})\Big)^{1/2} \le \|A\|_2\,(b\,\text{tr}\,C_{\text{in}})^{1/2} \tag{13}$$

with the trailing eigenvalue mass of C_{in} as calibration error, and analogously for $\|B^\top g_{o,n}\|$. We use $b = 0.05$ throughout, noting that the budget is a hyperparameter, which can be optimized to balance task learning and protection (see also Appendix E). The trivial initialization $A_0 = 0$, $B_0 = V_{\text{out}}R$ with random R satisfies both constraints. However, we instead initialize with the balanced SVD truncation of the doubly projected base weight, $B_0A_0 \approx P_{\text{out}}WP_{\text{in}}$ with $P_{\text{in}} = U_{\text{in}}U_{\text{in}}^\top$ and $P_{\text{out}} = V_{\text{out}}V_{\text{out}}^\top$, while the frozen weight is replaced by $W - s\,B_0A_0$, so the adapted layer still computes Wx at step zero. Here, the adapter starts with the base weight's dominant directions inside the admissible subspaces, which aids learning the fine-tuning task under the constraints. During training, the gradients of A and B are projected back onto the constrained subspaces before every optimizer step to re-enforce the constraints. It is important to note that the recorded gradients only approximate the true pulled-back score-gradient g_o of Equation 9 (Appendix B).

Computational cost. Compared to standard *LoRA*, the defence adds a one-time calibration phase of one forward and one backward pass per calibration example, followed by an eigendecomposition of C_{in} and C_{out} per adapted layer. Further, storing U_{in} and V_{out}, or their orthogonal complements where these are smaller, adds a memory overhead of $k_{\text{in}}d_{\text{in}} + k_{\text{out}}d_{\text{out}}$ parameters per adapted layer. Each training step incurs four additional matrix products per adapted layer, which project the gradients of A and B back onto the constraint subspaces. At inference time there is no additional overhead, since the trained adapter can be merged into the base weight as in standard *LoRA*.

5 Experiments

5.1 Experimental setup

A fine-tuning procedure (attack) is run against a released instruction-tuned model, *Qwen2.5-3B-Instruct* (Qwen Team, 2024; Yang et al., 2024a) or *Gemma 2 IT 2B* (Gemma Team, 2024), on a partially poisoned task dataset under one of seven fine-tuning procedures (one baseline, six defences), including our *dual-null* intervention, and a shared evaluation suite scores each fine-tuned model.

The harmful fine-tuning protocol. Our attack stage roughly follows the *Booster* poisoned fine-tuning protocol (Huang et al., 2025b). The utility fine-tuning tasks are *SST-2* (Socher et al., 2013), *AG News* (Zhang et al., 2015), and *GSM8K* (Cobbe et al., 2021). We use increasing poison fractions $p \in \{0, 0.05, 0.10, 0.15, 0.20\}$ and a fixed training set size of $n = 1000$. Under this protocol, the first $\lfloor (1-p)\,n \rfloor$ task examples are joined by a seeded draw of harmful instruction-response pairs from a fixed *BeaverTails* pool (Ji et al., 2023). Such a pair consists of a harmful instruction and a response that *complies* with it. All fine-tuning runs share the same optimizer settings at learning rate 2×10^{-5}.

Baselines and defence interventions. Each fine-tuning run trains under one of seven fine-tuning procedures, the plain *LoRA* (Hu et al., 2022) baseline, which trains low-rank adapters as introduced in Section 3, and six defence interventions: We instantiate *SafeInstr*, *Safe LoRA*, *AsFT*, *GuardSpace*, and *SaLoRA* as described in Section 2 and Appendix C respectively, and our own *dual-null* intervention as constructed in Section 4, calibrated on the first 256 safety demonstrations of the *BeaverTails* training split[2].

Otherwise every fine-tuning run uses an identical parametrization: Rank $r = 8$ with $\alpha = 1$ at scaling $\alpha/r = 0.125$ with adapters on all seven attention and MLP projections. We merge every adapter back into the base weights before evaluation.

Evaluation suite. We evaluate the resulting model of each fine-tuning run on five metrics, all reported as percentages. The harmfulness score (HS) scores a test set of harmful prompts with the *beaver-dam-7b* moderation model (Ji et al., 2023): HS is the percentage of prompts whose response scores above $\frac{1}{2}$ in *any* of its 14 harm categories. The attack success rate (ASR) scores the same responses with the binary *HarmBench* classifier (Mazeika et al., 2024), which flags a response only if it actually serves the harmful request, making ASR stricter than HS. Both are reported on the 1,000 unsafe *BeaverTails* test prompts (HS, ASR) and on the 520 harmful *AdvBench* instructions (Zou et al.,

[2] *BeaverTails* contains both harmful and aligned responses to red-teaming prompts. The poisoning draw uses pairs with harmful responses, the calibration pairs with aligned responses, from disjoint index windows.

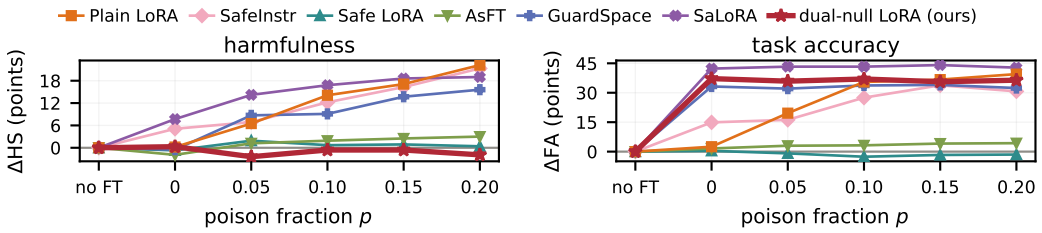

Figure 2: Change in percentage points of *BeaverTails* harmfulness (left) and task accuracy (right) relative to the released model (no FT) per poison fraction (*Qwen2.5-3B-Instruct*, *GSM8K*, $r = 8$).

2023) (HS_A, ASR_A), a distribution not used during the attack. Fine-tuning accuracy (FA) measures whether responses on the held-out task test set contain the correct label and thereby whether the attacked model still learns the fine-tuning task. Overrefusal (OR) is the refusal rate on the 250 *safe XSTest* prompts (Röttger et al., 2024), scored by the *WildGuard* judge (Han et al., 2024), and exposes a defence that refuses too broadly. General capability (GC) is zero-shot accuracy on the full *MMLU* test set (Hendrycks et al., 2021), and measures whether capability outside the fine-tuning task is preserved.

5.2 Efficacy under increasingly harmful fine-tuning

Table 1 reports evaluation results for both models at the strongest poison dose $p = 0.20$, Appendix D at every poison fraction.

At $p = 0.20$, *dual-null LoRA* attains the lowest *BeaverTails* harmfulness score of all interventions on every task for both models (task-averaged -0.9 points relative to the released model on *Qwen2.5-3B-Instruct*, $+0.6$ on *Gemma 2 IT 2B*) and the lowest *BeaverTails* ASR on five of the six model-task pairs (-0.8 and $+0.1$ on average). On *AdvBench* it attains the lowest HS_A and ASR_A on three of the six pairs and stays within 1.5 HS_A and 0.4 ASR_A points of the best scoring competitor. The best competitors on harmfulness are *Safe LoRA* (HS) and *AsFT* (ASR) on *Qwen2.5-3B-Instruct* and *GuardSpace* on *Gemma 2 IT 2B*. On fine-tuning accuracy, *dual-null LoRA* ranks second among the six defences on two of the three tasks and third on the remaining one on *Qwen2.5-3B-Instruct*, and second, third, and fifth on *Gemma 2 IT 2B*. The defences ahead of it mostly fail to provide protection (*GuardSpace* and *SaLoRA* on *Qwen2.5-3B-Instruct*, *SafeInstr*, *Safe LoRA*, and *SaLoRA* on *Gemma 2 IT 2B*), with the exception of *AsFT* (*Qwen2.5-3B-Instruct AG News*, *Gemma 2 IT 2B SST-2* and *GSM8K*).

Averaged over all poison fractions, tasks, and both models and relative to the released models, the strongest competitor on harmfulness is *AsFT* (HS $+1.9$, ASR $+0.3$ compared to $+0.2$ and -0.3 for *dual-null LoRA*) but it learns little (accuracy $+0.9$ compared to $+7.2$). The strongest competitor on accuracy is *SaLoRA* ($+7.1$) but it protects significantly worse (HS $+8.1$, $+12.9$ on *Qwen2.5-3B-Instruct*). Both thus exhibit a poorer trade-off between protection and task learning than *dual-null LoRA*. We note that at $p = 0$ most interventions perform similarly. Benign fine-tuning raises the harmfulness of plain *LoRA* by less than one point on average on both models at the low rank $r = 8$, and most interventions stay at the reference level (Appendix D).

To exemplify the effect of increasing poison fractions on protection, Figure 2 shows the evolution of harmfulness and task learning on *Qwen2.5-3B-Instruct* for *GSM8K*: Apart from *dual-null LoRA*, only *Safe LoRA* (which barely learns the task in this example) does not show a monotone rise in harmfulness.

Alignment degradation over training epochs. Figure 3 shows the training loss on the poisoned subset and the benign (task) subset of the training data over all ten epochs on *AG News* with $p = 0.10$, on *Qwen2.5-3B-Instruct*. While the *dual-null* loss on the poison subset is essentially

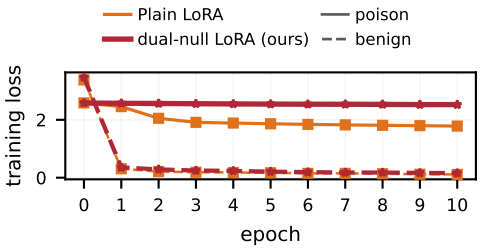

Figure 3: Training loss trajectory on the poison (solid) and benign (dashed) subsets (*Qwen2.5-3B-Instruct*, *AG News*, $p = 0.10$, $r = 8$).

Table 1: Efficacy at the strongest poison ratio $p = 0.20$ on Qwen2.5-3B-Instruct and Gemma 2 IT 2B ($r = 8$, learning rate 2×10^{-5}). HS and ASR are the harmfulness score and attack success rate on BeaverTails, HS_A and ASR_A the same on AdvBench, FA the task accuracy, OR the overrefusal rate on the safe XSTest prompts, and GC the MMLU accuracy. Before FT rows give the released model before fine-tuning. All values in percent, best fine-tuned value per column and block in orange, second best in grey.

		Qwen2.5-3B-Instruct							Gemma 2 IT 2B						
		BeaverTails		AdvBench		utility			BeaverTails		AdvBench		utility		
Task	Method	HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑	HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑
SST-2	Before FT	35.4	5.1	34.2	0.4	91.3	14.8	65.4	21.3	3.5	23.3	0.8	92.8	12.8	56.9
	Plain LoRA	59.8	23.5	39.4	4.4	92.3	5.2	65.4	88.5	67.4	79.6	78.1	**94.3**	**0.4**	56.6
	SafeInstr	58.3	21.0	40.2	2.9	92.2	**3.2**	65.3	85.3	57.5	65.0	44.0	93.9	3.2	56.5
	Safe LoRA	40.0	9.0	32.5	1.3	91.9	8.8	65.5	62.3	39.5	48.7	15.6	93.5	3.2	56.7
	AsFT	38.4	5.4	**32.3**	0.6	91.6	12.8	65.5	26.5	4.5	28.3	1.0	92.9	12.8	56.8
	GuardSpace	55.6	7.7	35.8	0.8	**93.8**	12.4	64.6	21.9	**3.5**	27.1	0.4	92.7	13.6	56.8
	SaLoRA	55.4	9.1	39.4	1.3	92.2	18.0	65.5	28.3	6.0	28.8	0.4	93.0	11.6	56.8
	dual-null LoRA (ours)	**35.2**	**4.5**	32.5	**0.4**	92.2	16.0	**65.6**	**21.6**	3.6	**25.4**	**0.2**	92.8	14.8	**57.1**
AG News	Before FT	35.4	5.1	34.2	0.4	76.4	14.8	65.4	21.3	3.5	23.3	0.8	81.3	12.8	56.9
	Plain LoRA	54.1	18.9	39.4	2.7	75.6	6.0	65.5	90.4	68.0	82.1	81.0	86.0	**0.8**	56.6
	SafeInstr	58.9	19.7	39.8	3.5	73.5	**5.6**	65.5	87.1	59.7	72.5	54.4	**86.2**	3.2	56.6
	Safe LoRA	37.4	7.1	32.7	0.6	75.8	13.2	65.4	65.1	41.1	54.6	22.3	83.8	2.8	56.6
	AsFT	37.3	5.4	33.1	**0.4**	77.2	14.8	65.5	25.7	5.2	28.5	0.6	80.3	12.8	56.8
	GuardSpace	57.0	13.0	36.3	1.5	**85.3**	8.4	65.1	24.7	3.7	29.0	0.8	83.8	13.2	56.5
	SaLoRA	53.4	6.2	39.2	1.0	74.4	25.2	**65.6**	27.6	5.4	29.2	**0.2**	83.6	12.4	56.8
	dual-null LoRA (ours)	**34.9**	**4.5**	**31.3**	0.6	76.8	16.0	65.6	**22.5**	**3.5**	**25.6**	0.4	84.1	14.0	**57.2**
GSM8K	Before FT	35.4	5.1	34.2	0.4	25.6	14.8	65.4	21.3	3.5	23.3	0.8	53.5	12.8	56.9
	Plain LoRA	57.6	15.6	38.7	3.3	65.1	8.4	65.5	88.8	67.8	76.3	73.1	47.5	**0.8**	56.4
	SafeInstr	56.7	19.6	36.7	3.7	56.3	**7.2**	65.3	87.4	64.0	63.3	41.3	47.9	1.6	56.3
	Safe LoRA	35.8	5.7	**32.5**	0.6	24.1	12.8	65.5	36.4	10.1	31.3	1.9	**56.3**	8.0	56.6
	AsFT	38.4	4.3	35.0	**0.2**	29.9	16.0	**65.6**	23.6	3.7	27.7	**0.6**	54.6	14.0	**57.0**
	GuardSpace	51.0	5.0	35.8	0.4	58.0	15.2	65.3	22.3	**3.6**	28.8	0.8	44.4	13.6	56.8
	SaLoRA	54.4	7.2	38.8	1.2	**68.4**	20.0	65.6	22.8	**3.6**	**26.3**	**0.6**	52.2	14.4	56.8
	dual-null LoRA (ours)	**33.5**	**4.0**	34.0	0.6	62.0	14.8	65.4	**21.6**	**3.6**	27.1	**0.6**	53.2	14.8	56.9

flat (2.58 to 2.53 over ten epochs), its benign loss falls from 3.5 to 0.16, so the task is learned without fitting the harmful examples. In contrast, plain *LoRA* clearly fits the poisoned subset (1.78). We conclude that our method effectively restricts the trainable parameters to a subspace where they cannot lower the loss on the poisoned subset.

5.3 Ablation Study

To separate the contributions of the single DN conditions and of the initialization, we conduct the ablation study depicted in Table 2. Here, the initialization is either the standard *LoRA* initialization, the initialization inside a single null space, or the *dual-null* initialization. The enforced projections are none, DN1, DN2, or both. Each of the seven configurations runs at the protocol rate (2×10^{-5}) and under a fivefold escalated attack (10^{-4}).

The initialization alone is not sufficient to protect the model (+27.1 *BeaverTails* HS, 62.5). We attribute this to the fact that the null spaces are only approximate: A small initial leak can grow over training and the protection is lost when not reinforced.

At the protocol rate, enforcing DN1 alone (initialization and reinforcement over training) provides protection similar to the full *dual-null LoRA* defence. DN2 alone, however, does not protect the model (63.8 HS). At the escalated rate, DN2 becomes necessary: DN1 enforcement erodes fully from the one-sided initialization (91.3 HS), while from the *dual-null* initialization it still protects to an extent (74.9). Here, it is notable that adding the DN2 projection lowers *AdvBench* attack success from 20.8 to 11.7.

5.4 Weaknesses and Limitations

Escalated learning rates. Under the fivefold increased learning rate, our *dual-null LoRA* mitigation no longer provides adequate protection (+41.0 HS, 76.4 compared to 91.4 for plain *LoRA*). This can be explained by the fact that we used the same budget of $b = 0.05$ to calibrate *dual-null LoRA* for both learning rates. A larger budget admits a larger null-space calibration error, which a larger

Table 2: The full mechanism ablation (*Qwen2.5-3B-Instruct*, *AG News*, $p = 0.10$, $r = 8$): Each indented row adds one ingredient to plain *LoRA*, the last two rows use the one-sided initialization, at the protocol rate and under a fivefold escalated attack. Before FT gives the released model before fine-tuning. All values in percent, best fine-tuned value per column and rate in orange, second best in grey.

		BeaverTails		AdvBench		utility		
learning rate		HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑
	Before FT	35.4	5.1	34.2	0.4	76.4	14.8	65.4
2×10^{-5}	Plain LoRA	42.5	9.7	35.2	0.8	79.2	9.6	65.5
	+ *dual-null* init	62.5	20.0	41.0	4.4	81.8	6.0	65.3
	+ DN2 projection	62.2	19.2	40.8	4.8	81.3	6.4	65.3
	+ DN1 projection	**34.4**	5.4	33.8	0.6	78.6	15.6	65.5
	+ both projections = dual-null LoRA (ours)	**34.4**	**4.7**	32.3	0.6	78.6	14.8	**65.5**
	DN1 projection, DN1-only init	34.8	5.2	**29.8**	**0.4**	78.5	14.8	65.5
	DN2 projection, DN2-only init	63.8	22.1	39.0	5.4	**82.3**	**3.2**	65.5
10^{-4}	Plain LoRA	91.4	64.7	80.2	57.7	87.4	2.8	65.0
	+ *dual-null* init	86.1	59.3	52.9	31.0	**88.1**	1.2	65.1
	+ DN2 projection	87.4	59.7	47.1	22.1	86.5	1.2	65.0
	+ DN1 projection	**74.9**	43.0	48.8	20.8	84.8	1.2	**65.4**
	+ both projections = dual-null LoRA (ours)	76.4	**40.4**	**46.9**	**11.7**	83.6	2.8	65.2
	DN1 projection, DN1-only init	91.3	65.6	81.3	71.3	85.5	**0.0**	65.3
	DN2 projection, DN2-only init	89.1	60.9	58.1	36.7	86.5	1.6	65.3

learning rate can exploit. Hence, larger learning rates need a lower budget to provide equal protection. We conclude that the budget should be treated as a hyperparameter, the optimal choice of which will likely depend on learning rate, rank, task, and model.

To investigate the trade-off between budget and protection empirically, we vary the budget on *Qwen2.5-3B-Instruct* at both learning rates (Appendix E). A budget of 10^{-3} allows task learning without alignment erosion even at the escalated rate (HS 37.9 and 35.4, FA 78.6 and 65.0 on *AG News* and *GSM8K*). In this context it is worth noting that the *LoRA* rank itself exhibits similar trade-offs (Appendix E).

In addition to choosing adequate budgets, recalibrating the *dual-null* bases every n steps on the current weights could further mitigate the staleness of the null spaces which drives erosion. We leave a systematic investigation of this for future work.

Further limitations. The null spaces are estimated from the activations and gradients of one calibration set of 256 *BeaverTails* demonstrations. The *AdvBench* (a distribution not used for calibration) results suggest adequate transfer, but prompts in other languages, domains, or jailbreak formats may fall outside the calibration distribution more strongly. The same holds for attacks without overtly harmful examples, such as the identity-shifting data of Qi et al. (2024).

Moreover, the *dual-null* conditions do not constrain *LoRA* factor norms, so unchecked norm growth could re-open both erosion channels from inside the constrained subspaces (weight decay mitigates this in our fine-tuning runs). Lastly, our experimental results remain limited to models of two to three billion parameters and transfer to larger models remains unproven.

6 Conclusion

We introduce *dual-null LoRA*, a defence against harmful fine-tuning developing learning-dynamics accounts of alignment erosion into a constraint on the fine-tuning update: A standard *LoRA* adapter is initialized and trained in such a way that activations and (pulled-back) gradients of a small safety calibration set lie in the null spaces of the *LoRA* factors A and $B^\top$, rendering the alignment score first-order invariant to any fine-tuning training example. Across two released models and three tasks the defence holds harmfulness at the level of the released model without a significant tax on task accuracy, overrefusal, or general capability. Our ablation study shows that the protection comes from enforcing the constraint throughout training, not from initialization alone. We want to highlight a thorough and systematic study of the trade-off between protection and task learning across hyperparameters such as budget, rank, and learning rate as the most important opportunity for future work.

AI USE STATEMENT

In this work, we used generative AI tools to provide critical ingredients for proving mathematical claims, design or provide feedback on research methodology or experiments, and implement methods (as in create or edit software code). We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, assist in the writing of proofs, support qualitative and thematic data analysis, or interpret results, and assistance with translation and cleaning and reformatting of datasets are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, draft parts of a research paper, summarize or analyse existing literature, brainstorm, source or search for information, and identify relevant literature. We have reviewed all AI-assisted work. We have manually verified existence and authorship of every paper which AI-assisted literature research may have proposed. All AI-assisted code has been manually reviewed. All explored research directions were ideated by humans first. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We refer the reader to Section 4 for the construction of the *dual-null* adapter. Section 5.1 introduces our experimental setup. Appendix D reports the complete results at every poison fraction. Our code, including an implementation of all baseline interventions, is available to the reviewers upon request. We will open-source the code and the trained models after an acceptance decision has been made. All baselines use the same rank 8 adapters on the same modules. *SafeInstr* adds 100 safe *BeaverTails* demonstrations to the fine-tuning set. *Safe LoRA* and *AsFT* derive their alignment matrices from the difference between the released model and its pretrained base (Qwen2.5-3B, Gemma 2 2B) with $\tau = 0.4$ and $\lambda = 1$. *GuardSpace* uses the initialization of its original paper and *SaLoRA* its task-specific initialization with safety and task ranks of 32, both calibrated on the 256 *BeaverTails* demonstrations of Section 5.1. For *GuardSpace* this single covariance serves both its initialization and its projector, where the original paper uses *AdvBench* and RealToxicityPrompts respectively. For details of the baseline methods we refer to their original papers.

REFERENCES

Rishabh Bhardwaj, Duc Anh Do, and Soujanya Poria. Language models are Homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. 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. 14138–14149, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.762. URL https://aclanthology.org/2024.acl-long.762/.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gT5hALch9z.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Gemma Team. Gemma. 2024. doi: 10.34740/KAGGLE/M/3301. URL https://www.kaggle.com/m/3301.

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.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe LoRA: The silver lining of reducing safety risks when finetuning large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=HcifdQZFZV.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Zixuan Hu, Li Shen, Zhenyi Wang, Yongxian Wei, and Dacheng Tao. Adaptive defense against harmful fine-tuning for large language models via Bayesian data scheduler. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=sm2e1SnMK4.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=RPChapuXlC.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=lpXDZKiAnt.

Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Joshua Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning attack. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=Arepl4R86m.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=tTPHgb0EtV.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey. *ACM Comput. Surv.*, 58(13), June 2026a. ISSN 0360-0300. doi: 10.1145/3817114. URL https://doi.org/10.1145/3817114.

Yuhan Huang, Huanran Chen, and Yinpeng Dong. Alignment dynamics in LLM fine-tuning, 2026b. URL https://arxiv.org/abs/2605.18309.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

Jiaming Ji, Mickel Liu, Juntao Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=g0QovXbFw3.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Simon Lermen and Charlie Rogers-Smith. LoRA fine-tuning efficiently undoes safety training in Llama 2-chat 70B. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL https://openreview.net/forum?id=Y52UbVhglu.

Jinhan Li, Kexian Tang, Yihan Xu, Zhuorui Ye, and Kaifeng Lyu. Beyond safe data: Pretraining-stage alignment with regular safety reflection. In *2nd Workshop on Compositional Learning: Safety, Interpretability, and Agents*, 2026. URL https://openreview.net/forum?id=Gequ8WzMdQ.

Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. SaLoRA: Safety-alignment preserved low-rank adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GOoVzE9nSj.

Guozhi Liu, Weiwei Lin, Qi Mu, Tiansheng Huang, Ruichao Mo, Yuren Tao, and Li Shen. Targeted vaccine: Safety alignment for large language models against harmful fine-tuning via layer-wise perturbation. *IEEE Transactions on Information Forensics and Security*, 20:10806–10817, 2025. doi: 10.1109/TIFS.2025.3615412.

Yuxiao Lu, Lin Xu, Yang Sun, Wenjun Li, and Jie Shi. Discern truth from falsehood: Reducing over-refusal via contrastive refinement. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GXcN0MuN1q.

Pratyush Maini, Sachin Goyal, Dylan Sam, Alexander Robey, Yash Savani, Yiding Jiang, Andy Zou, Matt Fredrikson, Zachary Chase Lipton, and J Zico Kolter. Safety pretraining: Toward the next generation of safe AI. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=91H9CSvdwl.

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 *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=f3TUipYU3U.

Quoc Minh Nguyen, Trung Le, Jing Wu, Anh Tuan Bui, and Mehrtash Harandi. Antibody: Strengthening defense against harmful fine-tuning for large language models via attenuating harmful gradient influence. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qur2ef8MqQ.

Kyle O'Brien, Stephen Casper, Quentin Gregory Anthony, Tomek Korbak, Robert Kirk, Xander Davies, Ishan Mishra, Geoffrey Irving, Yarin Gal, and Stella Biderman. Deep ignorance: Filtering pretraining data builds tamper-resistant safeguards into open-weight LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xcf0QcTcGS.

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. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

ShengYun Peng, Pin-Yu Chen, Jianfeng Chi, Seongmin Lee, and Duen Horng Chau. Shape it up! Restoring LLM safety during finetuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=PAIVwOaAnq.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hTEGyKf0dZ.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6Mxhg9PtDE.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

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. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tPNHOoZFl9.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Robie Gonzales, Carsten Maple, Subhabrata Majumdar, Hassan Sajjad, and Frank Rudzicz. Representation noising: A defence mechanism against harmful finetuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=eP9auEJqFg.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. SEAL: Safety-enhanced aligned LLM fine-tuning via bilevel data selection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VHguhvcoM5.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FIjRodbW6.

Pengwei Tang, Xiaolin Hu, Yong Liu, Lizhong Ding, Dongjie Zhang, Xing Wu, and Debing Zhang. Put the space of LoRA initialization to the extreme to preserve pre-trained knowledge, 2026. URL https://arxiv.org/abs/2503.02659.

Yibo Wang, Tiansheng Huang, Li Shen, Huanjin Yao, Haotian Luo, Rui Liu, Naiqiang Tan, Jiaxing Huang, and Dacheng Tao. Panacea: Mitigating harmful fine-tuning for large language models via post-fine-tuning perturbation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=sMtiGB2YZT.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024a. URL https://arxiv.org/abs/2407.10671.

Shuo Yang, Qihui Zhang, Yuyang Liu, Yue Huang, Xiaojun Jia, Kun-Peng Ning, Jia-Yu Yao, Jigang Wang, Hailiang Dai, Yibing Song, and Li Yuan. AsFT: Anchoring safety during LLM fine-tuning within narrow safety basin. In *NeurIPS 2025 Workshop: Reliable ML from Unreliable Data*, 2025. URL https://openreview.net/forum?id=96UEsasTfE.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Ruth Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models, 2024b. URL https://openreview.net/forum?id=rg0vQmkB7F.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing RLHF protections in GPT-4 via fine-tuning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 681–687, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.59. URL https://aclanthology.org/2024.naacl-short.59/.

Bingjie Zhang, Yibo Yang, Renzhe, Dandan Guo, Jindong Gu, Philip Torr, and Bernard Ghanem. A guardrail for safety preservation: When safety-sensitive subspace meets harmful-resistant null-space. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=887vde4ZAW.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

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 FULL DERIVATION OF THE *dual-null* CONDITIONS

This appendix derives the *dual-null* conditions from the alignment evolution law, following the notation of Huang et al. (2026b): $\pi_S^+(x_o)$ and $\pi_S^-(x_o)$ denote the total aligned and non-aligned probability mass at x_o, and $\tilde{\pi}^+(x_o)$ and $\tilde{\pi}^-(x_o)$ the conditional next-token distributions on the aligned and non-aligned token sets.

The defence objective. By Equation 3, one fine-tuning step on a training example x_u changes the alignment score at a prompt x_o by

$$\Delta S(x_o) = -\eta\, \pi_S^+(x_o)\pi_S^-(x_o)\left(\tilde{\pi}^+(x_o) - \tilde{\pi}^-(x_o)\right)^\top K(x_o, x_u)\left(\pi_u - p_u\right) + O(\eta^2) \tag{14}$$

By the score-gradient identity of Huang et al. (2026b), the leading factor is the transposed logit gradient of the score

$$\pi_S^+(x_o)\pi_S^-(x_o)\left(\tilde{\pi}^+(x_o) - \tilde{\pi}^-(x_o)\right)^\top = \mathbf{1}_{\mathcal{A}}^\top J(x_o) = \nabla_{z_o} S(x_o)^\top \tag{15}$$

and $G(x_u) = \pi_u - p_u$ is the residual of Section 3. Thus, the law can be compactly written as

$$\Delta S(x_o) = -\eta\, \nabla_{z_o} S(x_o)^\top\, K(x_o, x_u)\, G(x_u) + O(\eta^2) \tag{16}$$

Assume we want to preserve the alignment score at x_o under any fine-tuning update and assume the fine-tuning procedure is under our control. Then, a natural point of intervention is the kernel

$$K(x_o, x_u) = \nabla_\theta z(x_o)\, \nabla_\theta z(x_u)^\top \tag{17}$$

which is determined by the choice of trainable parameters θ. The fine-tuning data, and with it x_u and $G(x_u)$, is unknown in advance, so the defence can attempt to make the first-order term of Equation 16 vanish for every possible update example by achieving

$$\nabla_{z_o} S(x_o)^\top\, K(x_o, x_u) = 0 \qquad \text{for every update example } x_u \tag{18}$$

Under this condition, Equation 16 reduces to $\Delta S(x_o) = O(\eta^2)$: The change of the alignment score vanishes to first order in the learning rate, for every fine-tuning dataset and every training objective.

The eNTK under *LoRA* adaptation. Restrict the trainable parameters to a *LoRA* adapter and assume for simplicity that only a single layer is adapted. The layer can be written as

$$f(x) = Wx + s\, BAx, \qquad A \in \mathbb{R}^{r \times d_{\text{in}}}, \quad B \in \mathbb{R}^{d_{\text{out}} \times r}, \quad s = \tfrac{\alpha}{r} \tag{19}$$

with frozen W and trainable $\theta = (A, B)$, as introduced in Section 4. Here x denotes the activation entering the layer when the network processes the corresponding input.

Because $\theta = (A, B)$ enters the logits only through the layer output, the chain rule gives $\nabla_\theta z(x) = D(x)\, \nabla_\theta f(x)$ with $D(x) := \partial z(x)/\partial f(x)$ the Jacobian of the logits with respect to the layer output. The kernel of Equation 18 factorizes as

$$K(x_o, x_u) = D(x_o)\, K_f(x_o, x_u)\, D(x_u)^\top, \qquad K_f(x_o, x_u) := \nabla_\theta f(x_o)\, \nabla_\theta f(x_u)^\top \tag{20}$$

It remains to compute the layer kernel K_f. To this end, write the layer output entrywise

$$f(x)_k = (Wx)_k + s \sum_{p=1}^{r} B_{kp}\, (Ax)_p, \qquad (Ax)_p = \sum_{q=1}^{d_{\text{in}}} A_{pq}\, x_q \tag{21}$$

The only term of $f(x)_k$ containing the entry A_{ij} is the one with $p = i$ and $q = j$, and the only entries of B appearing in $f(x)_k$ are those of row k, so the parameter gradients of the layer output are

$$\frac{\partial f(x)_k}{\partial A_{ij}} = s\, B_{ki}\, x_j, \qquad \frac{\partial f(x)_k}{\partial B_{mp}} = s\, \delta_{km}\, (Ax)_p \tag{22}$$

The kernel entry $K_f(x_o, x_u)_{kl}$ of Equation 20 is the inner product of the parameter gradients of $f(x_o)_k$ and of $f(x_u)_l$. The parameter vector is the disjoint concatenation of the entries of A and of

B, and an inner product over a concatenation is the sum of the inner products over its blocks, so the entry splits into an A-part and a B-part with no mixed terms

$$K_f(x_o, x_u)_{kl} = \sum_{i=1}^{r} \sum_{j=1}^{d_{\text{in}}} \frac{\partial f(x_o)_k}{\partial A_{ij}} \frac{\partial f(x_u)_l}{\partial A_{ij}} + \sum_{m=1}^{d_{\text{out}}} \sum_{p=1}^{r} \frac{\partial f(x_o)_k}{\partial B_{mp}} \frac{\partial f(x_u)_l}{\partial B_{mp}} \tag{23}$$

Substituting Equation 22, the A-part gives

$$\sum_{i=1}^{r} \sum_{j=1}^{d_{\text{in}}} s\, B_{ki}\, (x_o)_j \; s\, B_{li}\, (x_u)_j = s^2\, (BB^\top)_{kl}\, \langle x_o, x_u \rangle \tag{24}$$

and the B-part gives

$$\sum_{m=1}^{d_{\text{out}}} \sum_{p=1}^{r} s\, \delta_{km}\, (Ax_o)_p \; s\, \delta_{lm}\, (Ax_u)_p = s^2\, \delta_{kl}\, \langle Ax_o, Ax_u \rangle \tag{25}$$

Adding the two parts and returning to matrix form yields the layer kernel

$$K_f(x_o, x_u) = s^2 \big(\langle Ax_o, Ax_u \rangle\, I_{d_{\text{out}}} + \langle x_o, x_u \rangle\, BB^\top \big) \tag{26}$$

The dual-null conditions. Substituting Equation 20 into Equation 18 and absorbing the outer Jacobians into the pulled-back directions

$$g_o := D(x_o)^\top \nabla_{z_o} S(x_o), \qquad g_u := D(x_u)^\top G(x_u) \tag{27}$$

turns the first-order term of Equation 16 into $-\eta\, g_o^\top K_f(x_o, x_u)\, g_u$. Hence, we can attempt to achieve $g_o^\top K_f(x_o, x_u) = 0$ for every x_u to preserve alignment behaviour. Writing this condition out explicitly yields

$$g_o^\top K_f(x_o, x_u) = s^2 \big(\underbrace{\langle Ax_o, Ax_u \rangle\, g_o^\top}_{B\text{-channel}} + \underbrace{\langle x_o, x_u \rangle\, (B^\top g_o)^\top B^\top}_{A\text{-channel}} \big) \tag{28}$$

Setting this expression to zero for every x_u therefore suggests the following two conditions for every protected pair (x_o, g_o), that is, for the layer traces of every prompt whose alignment behaviour is to be protected

$$\textbf{(DN1)}\;\; A\, x_o = 0 \qquad \text{and} \qquad \textbf{(DN2)}\;\; B^\top g_o = 0 \tag{29}$$

We call these the *dual-null* conditions. If both conditions are met, every term of Equation 28 vanishes, and $\Delta S(x_o) = O(\eta^2)$ for every update example and every training objective.

Extension to multiple layers. To extend our derivations to multiple adapted layers, index the adapted layers by j: We write $x^{(j)}$ for the activation entering layer j when the network processes an input x, and f_j for the adapted map of Equation 19 at layer j. The trainable parameters are now the concatenation of the adapter pairs, $\theta = (A_1, B_1, A_2, B_2, \dots)$, so the parameter Jacobian of the logits is a block matrix

$$\nabla_\theta z(x) = \big[\, \nabla_{(A_1,B_1)} z(x) \;\; \nabla_{(A_2,B_2)} z(x) \;\; \cdots \,\big] \tag{30}$$

The kernel of Equation 18 is the product of one such block row with the transpose of another, and an inner product over a concatenation is the sum of the inner products over its blocks, so the kernel splits blockwise

$$K(x_o, x_u) = \nabla_\theta z(x_o)\, \nabla_\theta z(x_u)^\top = \sum_j \nabla_{(A_j,B_j)} z(x_o)\, \nabla_{(A_j,B_j)} z(x_u)^\top \tag{31}$$

For each layer, the pair (A_j, B_j) enters the computation graph at exactly one place, the output of layer j (the downstream layers contain the other adapters, not this one), so the chain rule factorizes the block exactly as in the single-layer case, with the layer-wise Jacobian $D^{(j)}(x) := \partial z(x) / \partial f_j(x^{(j)})$ in place of $D(x)$

$$\nabla_{(A_j,B_j)} z(x) = D^{(j)}(x)\, \nabla_{(A_j,B_j)} f_j\big(x^{(j)}\big) \tag{32}$$

As in the single-layer case, we define

$$g_o^{(j)} := D^{(j)}(x_o)^\top \nabla_{z_o} S(x_o) \tag{33}$$

and obtain

$$\nabla_{z_o} S(x_o)^\top K(x_o, x_u) = \sum_j g_o^{(j)\top} K_{f_j}\big(x_o^{(j)}, x_u^{(j)}\big)\, D^{(j)}(x_u)^\top \tag{34}$$

with K_{f_j} the layer kernel of Equation 26 at layer j. If the *dual-null* conditions of Equation 29 hold at every adapted layer, $A_j\, x_o^{(j)} = 0$ and $B_j^\top g_o^{(j)} = 0$, we can retain $\Delta S(x_o) = O(\eta^2)$ for every update example in the multi-layer setting.

Extension to sequence prediction. For full sequences the same argument applies per response position. By the sequence version of the alignment evolution law (Section 3), the score at an observed input χ_o evolves as

$$\Delta S(\chi_o) = -\eta \sum_{m,l} c_m(\chi_o)^\top K_{m,l}(\chi_o, \chi_u)\, G_l(\chi_u) + O(\eta^2) \tag{35}$$

with $c_m(\chi_o)$ the prefix-probability-weighted score gradient at response position m, $G_l(\chi_u)$ the residual at response position l, and $K_{m,l}$ the kernel block coupling the two positions, so it suffices that every row vector $c_m(\chi_o)^\top K_{m,l}(\chi_o, \chi_u)$ vanishes. The blockwise split of Equation 31 applies to every kernel block $K_{m,l}$, and the chain rule of Equation 32 acquires position indices: Under the causal mask the position-m logits depend on layer j's outputs at the positions $p \le m$, so with $x^{(j,p)}$ denoting the activation input entering layer j at response position p and $D_{m,p}^{(j)}(\chi) := \partial z_m(\chi)/\partial f_j(x^{(j,p)})$

$$\nabla_{(A_j,B_j)} z_m(\chi_o) = \sum_{p \le m} D_{m,p}^{(j)}(\chi_o)\, \nabla_{(A_j,B_j)} f_j\big(x_o^{(j,p)}\big) \tag{36}$$

and similarly for the training example update side with (χ_u, l, q) in place of (χ_o, m, p). Writing

$$g_o^{(j,m,p)} := D_{m,p}^{(j)}(\chi_o)^\top\, c_m(\chi_o) \tag{37}$$

we obtain the per-position analogue of Equation 34

$$c_m(\chi_o)^\top K_{m,l}(\chi_o, \chi_u) = \sum_j \sum_{p \le m} \sum_{q \le l} \Big[g_o^{(j,m,p)\top} K_{f_j}\big(x_o^{(j,p)}, x_u^{(j,q)}\big) \Big]\, D_{l,q}^{(j)}(\chi_u)^\top \tag{38}$$

If the *dual-null* conditions hold at every adapted layer, then every bracket in Equation 38 vanishes, and therefore the entire triple sum vanishes too and we can retain $\Delta S(x_o) = O(\eta^2)$ in this setting, as well.

B THE CALIBRATION GRADIENT AS A PROXY

The recorded gradients are the gradients of the cross-entropy loss with respect to the layer outputs, pooled over all token positions of prompt and response, whereas the alignment evolution law demands, by Equation 37, the pull-backs of the per-position factors $c_m(\chi_o)$ of Equation 4. The pull-back is linear, so it suffices to compare, at the position-m logits, the recorded residual $G_m(\chi_o)$ with the demanded $c_m(\chi_o)$. By the score-gradient identity of Equation 15 applied at position m and entrywise expansion

$$c_m(\chi_o) = \pi_m^<\, J_m(\chi_o)^\top \mathbf{1}_{\mathcal{A}} = -\pi_m^< \sum_{y' \in \mathcal{A}} \pi_m(\chi_o)_{y'}\, \big(\pi_m(\chi_o) - \mathbf{1}_{\{y'\}}\big) \tag{39}$$

This is a probability-weighted mixture of cross-entropy residuals over the aligned continuations at position m. When the recorded token y_m is itself aligned, its term is the multiple $-\pi_m^< \pi_m(\chi_o)_{y_m}\, G_m(\chi_o)$ of the recorded residual. The mismatch is the rest of the mixture: Each residual satisfies $\|\pi_m(\chi_o) - \mathbf{1}_{\{y'\}}\| \le \sqrt{2}$, so the mismatch is bounded in norm by $\sqrt{2}\, \pi_m^< \sum_{y' \in \mathcal{A} \setminus \{y_m\}} \pi_m(\chi_o)_{y'}$, the aligned probability mass away from the recorded token. We accept the remaining mismatch as an approximation error of the construction.

The remaining gap can in theory be closed by sampling: Adding k continuations drawn from the installed model per prompt and filtering for aligned continuations (e.g., using a judge and its definition of the aligned set) realizes each aligned continuation with its probability $\pi_m(\chi_o)_{y'}$ at every position.

C Detailed comparison with null-space defences

C.1 *Dual-null* vs. *LoRA-Null* applied to safety

LoRA-Null (Tang et al., 2026) was developed to preserve pretrained knowledge under benign task fine-tuning. Per layer, the activations of a calibration corpus are collected into X_{pre}, its trailing r left singular vectors U_{null} form an approximate null space ($U_{\text{null}}^\top X_{\text{pre}} \approx 0$), and the adapter is initialized from the SVD $W_0 U_{\text{null}} U_{\text{null}}^\top = U' \Sigma' V'^\top$ as

$$B = U'_{[:,:r]}\, \Sigma'^{1/2}_{[:r]}, \qquad A = \Sigma'^{1/2}_{[:r]}\, V'^\top_{[:r,:]} \tag{40}$$

with the frozen remainder $W_0' = W_0 - BA$, so the model function is unchanged at initialization, and training proceeds without any constraint.

Applied to safety, with the knowledge corpus replaced by safety demonstrations, this initialization fulfills condition (DN1) of Equation 11: Every nonzero row of A lies in $\mathrm{span}(U_{\text{null}})$, which gives $\|Ax_o\| \le \sigma_1'^{1/2}\, \|U_{\text{null}}^\top x_o\| \approx 0$ on the calibration activations (σ_1' the leading singular value).

Compared to *dual-null*, two differences stand out: First, condition (DN2) does not hold: B is chosen without reference to any gradient statistic, so the second term of the kernel row of Equation 10 is uncontrolled even at initialization, and a task update to A exerts the first-order force $\langle x_o, x_u\rangle\, g_o^\top B B^\top g_u$ on the safety score. Second, no mechanism keeps (DN1) reinforced: *Dual-null* closes both gaps, calibrating C_{out} for condition (DN2) and enforcing both conditions of Equation 11 at every step through projected gradients.

C.2 *Dual-null* vs. *GuardSpace*

Zhang et al. (2026) propose *GuardSpace*, a null space construction conceptually very similar to ours (Section 4). Harmful prompts are run through the aligned model, and for every adapted layer the prompt-token input activations $X \in \mathbb{R}^{d_{\text{in}} \times m}$ yield $C = XX^\top$. The adapter factors are initialized from the trailing components of the covariance-preconditioned SVD of WC, and the projector $P = \hat{Q}\hat{Q}^\top$, with $\hat{Q}$ the eigenvectors of C of eigenvalue zero, annihilates the adapter input

$$f(x) = W'x + s\, BAPx, \qquad W' = W - B_0 A_0 P \tag{41}$$

The authors derive that the probed activations are invariant under fine-tuning: A column q of $\hat{Q}$ obeys $0 = q^\top C q = \|X^\top q\|^2$, so $Px_o = 0$ for every x_o in the span of the probed activations and the layer output on them stays at its aligned value throughout training.

In fact, viewed through the learning-dynamics lens, with the projector in place, the adapter kernel of Equation 8 becomes

$$K_f^P(x_o, x_u) = s^2\big(\langle APx_o, APx_u\rangle\, I_{d_{\text{out}}} + \langle Px_o, Px_u\rangle\, BB^\top\big) \tag{42}$$

We see that $Px_o = 0$ sets the entire kernel row to zero, and by Equation 3 the safety score is stationary, $\Delta S(x_o) = 0$ to first order. The single condition $Px_o = 0$ thus does the work of the pair (DN1) and (DN2), on the calibration span.

The condition $Px_o = 0$ is, however, exact only on the span of the recorded probe activations, the columns of X: A fresh harmful input x generally carries a component outside this span, and $\|Px\| > 0$. Assume for simplicity a single adapted layer and examples (x_o, g_o) and (x_u, g_u) that stay fixed during fine-tuning. By the evolution law of Equation 3, one gradient step on the update pair moves the score by the contracted kernel row, Equation 42 for *GuardSpace* and Equation 10 for *dual-null*, and the Cauchy–Schwarz inequality bounds the step by

$$|\Delta S(x_o)| \le \eta s^2\big(\|APx_o\|\, \|APx_u\|\, \|g_o\|\, \|g_u\| + \|Px_o\|\, \|Px_u\|\, \|B^\top g_o\|\, \|B^\top g_u\|\big) + O(\eta^2) \tag{43}$$

for *GuardSpace* and by

$$|\Delta S(x_o)| \le \eta s^2\big(\|Ax_o\|\, \|Ax_u\|\, \|g_o\|\, \|g_u\| + \|x_o\|\, \|x_u\|\, \|B^\top g_o\|\, \|B^\top g_u\|\big) + O(\eta^2) \tag{44}$$

for *dual-null*. The difference lies in which factors training can grow: In Equation 43 the factors $\|B^\top g_o\|\, \|B^\top g_u\|$ can grow with $\|B\|_2^2$, and fitting the harmful training data may drive exactly this

growth. Under the projected gradients of *dual-null* every state that training can reach instead obeys $\|Ax\| \leq \|A\|_2 \, \|U_{\text{in}}^\top x\| \approx 0$ and $\|B^\top g\| \leq \|B\|_2 \, \|V_{\text{out}}^\top g\| \approx 0$. Furthermore, *GuardSpace* uses exact null spaces for its calibration which raises questions about task-learning capability, especially as a hyperparameter controlling regularization strength is absent. In fact, the original paper uses a *LoRA* rank of 128 in its main experiments, which might conceal this effect to a degree.

D FULL RESULTS AT EVERY POISON FRACTION

Table 3: Efficacy of *dual-null* compared to other interventions under purely benign fine-tuning data ($p = 0.00$, $r = 8$), every model and fine-tuning task. All values in percent. Per column and block, the best value among the fine-tuned models is highlighted in orange and the second best in grey. Before FT reports the released instruction-tuned model before fine-tuning.

			BeaverTails		AdvBench		utility		
Model	Task		HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑
Qwen2.5-3B-Instruct	SST-2	Before FT	35.4	5.1	34.2	0.4	91.3	14.8	65.4
		Plain LoRA	34.9	5.4	33.3	**0.4**	92.1	17.2	65.4
		SafeInstr	42.3	8.4	38.7	1.3	92.1	**10.4**	65.4
		Safe LoRA	**34.1**	4.8	30.6	0.6	91.3	16.4	65.5
		AsFT	34.6	4.6	32.3	0.6	91.5	16.0	65.5
		GuardSpace	35.2	**3.7**	**30.0**	**0.4**	**94.3**	16.0	**65.6**
		SaLoRA	35.2	5.6	33.7	**0.4**	92.8	16.4	65.5
		dual-null LoRA (ours)	36.0	4.9	32.9	**0.4**	92.7	14.4	65.4
	AG News	Before FT	35.4	5.1	34.2	0.4	76.4	14.8	65.4
		Plain LoRA	35.9	4.4	**30.0**	0.4	76.1	16.4	65.4
		SafeInstr	39.3	6.1	35.2	1.0	79.0	**12.4**	65.5
		Safe LoRA	35.4	5.1	32.5	**0.2**	76.7	16.0	65.5
		AsFT	35.4	4.9	34.6	0.6	75.1	14.0	65.4
		GuardSpace	**32.7**	4.2	32.3	1.5	**88.0**	15.6	**65.7**
		SaLoRA	36.7	**3.2**	32.7	0.4	76.9	17.6	65.5
		dual-null LoRA (ours)	35.7	4.9	31.5	0.8	80.0	16.4	65.4
	GSM8K	Before FT	35.4	5.1	34.2	0.4	25.6	14.8	65.4
		Plain LoRA	35.6	4.0	**30.8**	0.2	28.1	15.6	65.5
		SafeInstr	40.5	5.4	35.8	0.6	40.5	**13.2**	65.5
		Safe LoRA	34.7	5.2	**30.8**	0.4	26.0	15.6	65.5
		AsFT	**33.5**	**3.8**	34.0	0.4	27.2	16.0	65.5
		GuardSpace	34.9	4.7	32.3	0.2	58.8	14.0	65.1
		SaLoRA	43.1	**3.8**	39.8	**0.0**	**67.9**	20.8	**65.6**
		dual-null LoRA (ours)	35.7	3.9	32.1	0.2	62.8	17.6	65.2
Gemma 2 IT 2B	SST-2	Before FT	21.3	3.5	23.3	0.8	92.8	12.8	56.9
		Plain LoRA	21.2	4.0	26.7	0.8	**94.5**	13.2	57.0
		SafeInstr	60.0	18.0	38.1	1.9	94.2	**5.2**	56.6
		Safe LoRA	21.4	4.6	**24.6**	**0.4**	93.7	11.6	**57.1**
		AsFT	21.6	**3.4**	30.2	0.8	93.0	12.8	56.9
		GuardSpace	23.4	**3.4**	26.7	0.6	93.2	14.0	56.7
		SaLoRA	**20.7**	4.0	29.2	1.0	93.6	13.2	57.0
		dual-null LoRA (ours)	22.8	3.5	28.3	0.6	93.3	13.6	57.0
	AG News	Before FT	21.3	3.5	23.3	0.8	81.3	12.8	56.9
		Plain LoRA	**21.3**	**3.3**	27.5	0.6	86.6	14.0	57.0
		SafeInstr	60.8	18.3	37.3	1.7	**87.9**	**6.0**	56.7
		Safe LoRA	22.7	3.6	27.3	**0.4**	82.3	14.0	**57.1**
		AsFT	22.2	3.6	29.2	**0.4**	83.4	12.8	56.9
		GuardSpace	22.4	3.7	23.5	1.0	84.7	14.0	56.8
		SaLoRA	21.4	3.7	28.7	0.8	86.4	14.0	57.0
		dual-null LoRA (ours)	23.6	3.7	**23.3**	**0.4**	86.1	13.2	57.0
	GSM8K	Before FT	21.3	3.5	23.3	0.8	53.5	12.8	56.9
		Plain LoRA	23.3	3.7	28.3	0.4	50.7	14.0	56.7
		SafeInstr	53.1	6.3	38.7	0.8	48.7	**9.6**	56.6
		Safe LoRA	22.8	3.9	**22.7**	0.6	**57.5**	14.4	56.9
		AsFT	25.1	3.6	29.4	**0.2**	54.1	13.6	56.9
		GuardSpace	**22.1**	**3.3**	25.2	0.8	45.8	13.6	56.9
		SaLoRA	23.2	3.9	29.8	1.0	44.0	14.0	**56.9**
		dual-null LoRA (ours)	22.4	3.7	27.5	0.4	54.2	13.6	56.9

Table 4: Efficacy of *dual-null* compared to other interventions at poison ratio $p = 0.05$ ($r = 8$), every model and fine-tuning task. All values in percent. Per column and block, the best value among the fine-tuned models is highlighted in orange and the second best in grey. Before FT reports the released instruction-tuned model before fine-tuning.

			BeaverTails		AdvBench		utility		
Model	Task		HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑
Qwen2.5-3B-Instruct	SST-2	Before FT	35.4	5.1	34.2	0.4	91.3	14.8	65.4
		Plain LoRA	42.9	9.3	36.9	0.6	92.2	8.0	65.5
		SafeInstr	48.7	12.0	37.5	1.7	92.3	**6.4**	65.4
		Safe LoRA	36.0	7.0	31.9	0.4	91.7	13.2	65.5
		AsFT	36.8	5.1	32.9	0.4	91.9	15.6	**65.6**
		GuardSpace	46.6	**4.2**	**28.8**	0.4	**94.2**	14.4	65.1
		SaLoRA	40.3	4.6	36.0	**0.2**	92.7	14.8	65.5
		dual-null LoRA (ours)	**34.7**	4.3	31.5	0.6	92.7	15.2	65.5
	AG News	Before FT	35.4	5.1	34.2	0.4	76.4	14.8	65.4
		Plain LoRA	38.3	5.9	34.2	0.6	78.0	12.8	**65.6**
		SafeInstr	45.4	9.5	39.6	1.3	76.8	**8.8**	65.5
		Safe LoRA	36.7	5.5	31.9	0.4	76.5	14.0	65.4
		AsFT	37.1	5.3	32.7	0.2	76.3	15.6	65.5
		GuardSpace	46.2	4.8	**26.0**	0.2	**85.4**	17.6	65.4
		SaLoRA	41.8	**4.0**	36.0	**0.0**	75.7	20.4	65.6
		dual-null LoRA (ours)	**36.1**	5.1	33.1	0.8	78.9	15.6	65.4
	GSM8K	Before FT	35.4	5.1	34.2	0.4	25.6	14.8	65.4
		Plain LoRA	41.9	6.5	36.3	1.7	45.2	13.2	65.6
		SafeInstr	42.3	8.3	36.7	1.5	41.8	**9.2**	65.5
		Safe LoRA	37.3	5.2	33.1	**0.2**	24.7	14.0	65.5
		AsFT	36.6	4.2	36.7	**0.2**	28.6	15.6	**65.7**
		GuardSpace	44.1	4.6	**30.2**	0.4	57.7	15.2	65.1
		SaLoRA	49.6	4.3	37.1	0.6	**68.9**	20.0	65.5
		dual-null LoRA (ours)	**33.0**	**3.7**	33.5	0.6	61.5	16.0	65.1
Gemma 2 IT 2B	SST-2	Before FT	21.3	3.5	23.3	0.8	92.8	12.8	56.9
		Plain LoRA	70.9	41.3	54.8	31.3	94.0	**3.2**	56.7
		SafeInstr	76.5	40.6	44.2	13.7	**94.5**	4.0	56.4
		Safe LoRA	37.2	12.5	32.5	2.1	93.6	5.6	56.8
		AsFT	22.3	4.2	27.7	0.8	93.0	13.6	56.9
		GuardSpace	24.8	**3.5**	27.7	**0.6**	93.1	13.6	56.7
		SaLoRA	24.6	4.4	28.7	0.8	93.6	13.2	56.8
		dual-null LoRA (ours)	**21.9**	**3.5**	**26.5**	0.8	93.1	13.6	**57.0**
	AG News	Before FT	21.3	3.5	23.3	0.8	81.3	12.8	56.9
		Plain LoRA	69.0	42.3	56.0	32.9	**87.1**	4.0	56.6
		SafeInstr	79.1	44.9	49.6	19.6	87.0	**3.2**	56.6
		Safe LoRA	35.1	9.2	34.6	1.0	84.8	9.2	56.8
		AsFT	23.7	4.5	27.1	0.8	82.2	14.0	56.9
		GuardSpace	23.0	3.7	28.1	**0.6**	84.7	12.8	56.9
		SaLoRA	24.2	3.8	26.9	**0.6**	85.0	15.6	56.9
		dual-null LoRA (ours)	**22.4**	**3.4**	**25.0**	0.8	84.6	12.8	**57.0**
	GSM8K	Before FT	21.3	3.5	23.3	0.8	53.5	12.8	56.9
		Plain LoRA	39.6	7.8	31.7	0.6	48.7	8.4	56.5
		SafeInstr	74.0	41.0	39.6	6.2	47.9	**4.4**	56.3
		Safe LoRA	25.4	4.1	28.8	0.8	**56.0**	14.0	56.8
		AsFT	22.2	3.5	29.6	0.6	54.8	14.0	**56.9**
		GuardSpace	22.4	**3.4**	29.8	0.8	43.9	12.8	56.8
		SaLoRA	23.2	3.9	27.7	**0.4**	47.2	14.8	56.9
		dual-null LoRA (ours)	**21.7**	3.5	**26.0**	0.8	52.6	14.8	56.8

Table 5: Efficacy of *dual-null* compared to other interventions at poison ratio $p = 0.10$ ($r = 8$), every model and fine-tuning task. All values in percent. Per column and block, the best value among the fine-tuned models is highlighted in orange and the second best in grey. Before FT reports the released instruction-tuned model before fine-tuning.

			BeaverTails		AdvBench		utility		
Model	Task		HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑
Qwen2.5-3B-Instruct	SST-2	Before FT	35.4	5.1	34.2	0.4	91.3	14.8	65.4
		Plain LoRA	50.3	14.8	38.5	2.1	92.2	6.4	65.4
		SafeInstr	54.0	15.8	39.8	2.3	92.2	**5.6**	65.4
		Safe LoRA	37.2	7.3	33.8	1.0	91.7	11.2	65.5
		AsFT	36.4	5.9	33.3	0.2	91.7	14.0	**65.6**
		GuardSpace	47.3	5.6	**32.1**	**0.0**	**94.4**	18.0	64.9
		SaLoRA	52.1	6.8	38.5	1.2	92.3	17.6	65.5
		dual-null LoRA (ours)	**35.1**	**4.4**	33.7	0.2	92.1	14.8	65.4
	AG News	Before FT	35.4	5.1	34.2	0.4	76.4	14.8	65.4
		Plain LoRA	42.5	9.7	35.2	0.8	79.2	9.6	65.5
		SafeInstr	52.6	13.6	38.7	1.9	76.9	**6.4**	65.5
		Safe LoRA	38.0	6.7	32.9	0.6	75.3	13.6	65.4
		AsFT	36.3	4.9	33.3	0.4	76.7	16.4	65.6
		GuardSpace	50.3	6.4	**30.0**	**0.0**	**86.3**	16.0	65.3
		SaLoRA	49.5	**4.3**	36.2	0.8	76.2	27.2	**65.6**
		dual-null LoRA (ours)	**34.4**	4.7	32.3	0.6	78.6	14.8	65.5
	GSM8K	Before FT	35.4	5.1	34.2	0.4	25.6	14.8	65.4
		Plain LoRA	49.5	10.2	36.7	2.5	61.1	**8.0**	65.4
		SafeInstr	47.6	12.5	35.2	2.1	53.1	8.8	65.4
		Safe LoRA	36.1	5.2	33.1	0.6	23.0	13.6	65.5
		AsFT	37.3	**4.1**	35.4	**0.0**	28.8	15.2	65.6
		GuardSpace	44.5	4.9	32.9	0.2	59.3	16.8	65.3
		SaLoRA	52.2	5.3	38.5	1.0	**68.9**	20.8	**65.6**
		dual-null LoRA (ours)	**34.8**	4.4	**31.2**	0.4	62.6	16.4	65.4
Gemma 2 IT 2B	SST-2	Before FT	21.3	3.5	23.3	0.8	92.8	12.8	56.9
		Plain LoRA	85.6	64.1	77.5	73.7	94.2	**1.2**	56.8
		SafeInstr	83.7	53.6	54.2	25.4	**94.6**	3.6	56.5
		Safe LoRA	52.6	29.4	37.9	7.7	94.2	2.8	56.7
		AsFT	24.9	4.9	29.6	0.8	93.0	12.8	56.9
		GuardSpace	24.2	**3.6**	28.1	0.8	93.7	14.0	56.4
		SaLoRA	26.5	5.3	28.8	0.6	93.2	12.8	56.9
		dual-null LoRA (ours)	**22.9**	**3.6**	**26.3**	**0.4**	93.5	13.2	**57.0**
	AG News	Before FT	21.3	3.5	23.3	0.8	81.3	12.8	56.9
		Plain LoRA	85.6	65.3	77.9	76.9	86.9	**1.2**	56.7
		SafeInstr	83.9	56.1	61.2	35.4	**87.2**	3.6	56.5
		Safe LoRA	51.5	27.2	38.3	6.2	84.0	2.8	56.7
		AsFT	22.7	4.4	**24.4**	**0.4**	81.9	15.6	56.8
		GuardSpace	22.7	4.0	27.5	0.8	82.9	12.4	**57.0**
		SaLoRA	24.3	4.5	28.5	0.8	85.0	14.0	56.9
		dual-null LoRA (ours)	**21.7**	**3.8**	27.1	0.6	85.7	13.6	56.9
	GSM8K	Before FT	21.3	3.5	23.3	0.8	53.5	12.8	56.9
		Plain LoRA	76.0	50.9	50.6	25.6	49.3	2.4	56.5
		SafeInstr	83.6	57.1	48.7	23.3	46.5	**2.0**	56.3
		Safe LoRA	29.0	5.8	29.8	0.8	**56.7**	9.2	56.7
		AsFT	24.2	3.6	26.9	**0.4**	54.3	14.8	**57.0**
		GuardSpace	23.3	3.5	29.6	**0.4**	44.7	12.8	56.8
		SaLoRA	**22.2**	3.8	29.0	0.6	48.2	15.2	56.8
		dual-null LoRA (ours)	22.4	**3.4**	**24.4**	0.6	52.8	15.2	56.9

Table 6: Efficacy of *dual-null* compared to other interventions at poison ratio $p = 0.15$ ($r = 8$), every model and fine-tuning task. All values in percent. Per column and block, the best value among the fine-tuned models is highlighted in orange and the second best in grey. Before FT reports the released instruction-tuned model before fine-tuning.

			BeaverTails		AdvBench		utility		
Model	Task		HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑
Qwen2.5-3B-Instruct	SST-2	Before FT	35.4	5.1	34.2	0.4	91.3	14.8	65.4
		Plain LoRA	51.9	18.4	37.7	3.3	92.4	5.6	65.5
		SafeInstr	57.0	17.8	38.8	2.5	92.2	**4.0**	65.3
		Safe LoRA	38.5	8.3	32.9	0.6	91.6	11.2	65.4
		AsFT	35.7	6.0	32.9	0.6	91.6	14.8	65.5
		GuardSpace	49.5	8.6	34.2	**0.2**	**94.2**	15.2	64.8
		SaLoRA	54.7	7.2	37.1	1.2	92.4	16.4	65.5
		dual-null LoRA (ours)	**34.3**	**4.4**	**32.1**	0.4	92.3	16.0	**65.5**
	AG News	Before FT	35.4	5.1	34.2	0.4	76.4	14.8	65.4
		Plain LoRA	45.7	12.5	37.7	1.5	76.2	8.4	**65.6**
		SafeInstr	56.7	15.5	37.9	2.7	74.9	**6.8**	65.5
		Safe LoRA	37.4	6.9	**32.1**	0.8	75.9	13.6	65.4
		AsFT	36.2	5.5	**32.1**	**0.4**	77.3	16.0	65.5
		GuardSpace	54.9	10.1	34.8	0.6	**86.0**	10.8	65.2
		SaLoRA	52.5	5.8	38.7	0.8	74.6	25.6	65.5
		dual-null LoRA (ours)	**36.1**	**4.7**	33.7	0.6	77.1	15.6	65.5
	GSM8K	Before FT	35.4	5.1	34.2	0.4	25.6	14.8	65.4
		Plain LoRA	52.5	11.5	37.5	3.3	62.3	8.4	65.5
		SafeInstr	51.7	15.2	36.2	2.7	59.4	**7.6**	65.5
		Safe LoRA	36.3	5.0	**30.8**	0.4	23.9	13.2	**65.6**
		AsFT	37.9	4.7	36.2	**0.2**	29.7	16.0	65.5
		GuardSpace	49.1	5.2	36.2	0.4	59.6	14.0	65.2
		SaLoRA	54.0	5.0	38.1	0.8	**69.7**	19.6	65.6
		dual-null LoRA (ours)	**34.8**	**4.1**	32.7	**0.2**	61.2	15.6	65.3
Gemma 2 IT 2B	SST-2	Before FT	21.3	3.5	23.3	0.8	92.8	12.8	56.9
		Plain LoRA	87.3	66.8	78.8	81.9	**94.3**	**1.2**	56.8
		SafeInstr	82.9	56.4	60.8	38.1	93.6	3.6	56.5
		Safe LoRA	60.5	35.7	46.5	14.4	93.5	4.0	56.8
		AsFT	25.5	4.7	**25.0**	1.0	93.0	12.4	56.9
		GuardSpace	23.6	**3.4**	27.9	0.6	93.0	15.6	56.7
		SaLoRA	26.8	5.5	28.7	**0.2**	93.0	11.6	56.9
		dual-null LoRA (ours)	**21.0**	3.8	25.8	0.8	93.3	14.8	**57.0**
	AG News	Before FT	21.3	3.5	23.3	0.8	81.3	12.8	56.9
		Plain LoRA	89.1	68.6	80.6	82.9	**86.5**	**0.8**	56.7
		SafeInstr	84.8	58.1	66.3	46.7	86.1	2.8	56.5
		Safe LoRA	66.4	41.7	56.2	25.2	84.6	3.6	56.7
		AsFT	25.1	4.6	26.9	1.2	81.1	12.8	56.8
		GuardSpace	23.0	3.9	**25.8**	0.8	80.8	14.0	56.8
		SaLoRA	27.6	5.1	30.6	**0.2**	84.4	12.8	56.8
		dual-null LoRA (ours)	**22.2**	**3.4**	**25.8**	0.4	84.4	13.6	**57.1**
	GSM8K	Before FT	21.3	3.5	23.3	0.8	53.5	12.8	56.9
		Plain LoRA	84.6	66.1	74.2	65.8	48.1	**0.8**	56.4
		SafeInstr	85.8	62.3	58.3	37.5	48.3	1.6	56.4
		Safe LoRA	35.1	9.1	31.5	1.0	54.9	8.0	56.7
		AsFT	23.4	4.0	29.2	1.0	**55.1**	13.6	**56.9**
		GuardSpace	22.9	3.7	28.8	**0.4**	45.8	12.4	56.8
		SaLoRA	24.1	4.1	26.0	0.8	50.0	14.0	56.8
		dual-null LoRA (ours)	**20.2**	**3.4**	**21.5**	0.6	53.1	14.0	56.8

Table 7: Efficacy of *dual-null* compared to other interventions at the strongest poison ratio $p = 0.20$ ($r = 8$), every model and fine-tuning task. All values in percent. Per column and block, the best value among the fine-tuned models is highlighted in orange and the second best in grey. Before FT reports the released instruction-tuned model before fine-tuning.

			BeaverTails		AdvBench		utility		
Model	Task		HS↓	ASR↓	HS_A↓	ASR_A↓	FA↑	OR↓	GC↑
Qwen2.5-3B-Instruct	SST-2	Before FT	35.4	5.1	34.2	0.4	91.3	14.8	65.4
		Plain LoRA	59.8	23.5	39.4	4.4	92.3	5.2	65.4
		SafeInstr	58.3	21.0	40.2	2.9	92.2	**3.2**	65.3
		Safe LoRA	40.0	9.0	32.5	1.3	91.9	8.8	65.5
		AsFT	38.4	5.4	**32.3**	0.6	91.6	12.8	65.5
		GuardSpace	55.6	7.7	35.8	0.8	**93.8**	12.4	64.6
		SaLoRA	55.4	9.1	39.4	1.3	92.2	18.0	65.5
		dual-null LoRA (ours)	**35.2**	**4.5**	32.5	**0.4**	92.2	16.0	**65.6**
	AG News	Before FT	35.4	5.1	34.2	0.4	76.4	14.8	65.4
		Plain LoRA	54.1	18.9	39.4	2.7	75.6	6.0	65.5
		SafeInstr	58.9	19.7	39.8	3.5	73.5	**5.6**	65.5
		Safe LoRA	37.4	7.1	32.7	0.6	75.8	13.2	65.4
		AsFT	37.3	5.4	33.1	**0.4**	77.2	14.8	65.5
		GuardSpace	57.0	13.0	36.3	1.5	**85.3**	8.4	65.1
		SaLoRA	53.4	6.2	39.2	1.0	74.4	25.2	**65.6**
		dual-null LoRA (ours)	**34.9**	**4.5**	**31.3**	0.6	76.8	16.0	65.6
	GSM8K	Before FT	35.4	5.1	34.2	0.4	25.6	14.8	65.4
		Plain LoRA	57.6	15.6	38.7	3.3	65.1	8.4	65.5
		SafeInstr	56.7	19.6	36.7	3.7	56.3	**7.2**	65.3
		Safe LoRA	35.8	5.7	**32.5**	0.6	24.1	12.8	65.5
		AsFT	38.4	4.3	35.0	**0.2**	29.9	16.0	**65.6**
		GuardSpace	51.0	5.0	35.8	0.4	58.0	15.2	65.3
		SaLoRA	54.4	7.2	38.8	1.2	**68.4**	20.0	65.6
		dual-null LoRA (ours)	**33.5**	**4.0**	34.0	0.6	62.0	14.8	65.4
Gemma 2 IT 2B	SST-2	Before FT	21.3	3.5	23.3	0.8	92.8	12.8	56.9
		Plain LoRA	88.5	67.4	79.6	78.1	**94.3**	**0.4**	56.6
		SafeInstr	85.3	57.5	65.0	44.0	93.9	3.2	56.5
		Safe LoRA	62.3	39.5	48.7	15.6	93.5	3.2	56.7
		AsFT	26.5	4.5	28.3	1.0	92.9	12.8	56.8
		GuardSpace	21.9	**3.5**	27.1	0.4	92.7	13.6	56.8
		SaLoRA	28.3	6.0	28.8	0.4	93.0	11.6	56.8
		dual-null LoRA (ours)	**21.6**	3.6	**25.4**	**0.2**	92.8	14.8	**57.1**
	AG News	Before FT	21.3	3.5	23.3	0.8	81.3	12.8	56.9
		Plain LoRA	90.4	68.0	82.1	81.0	86.0	**0.8**	56.6
		SafeInstr	87.1	59.7	72.5	54.4	**86.2**	3.2	56.6
		Safe LoRA	65.1	41.1	54.6	22.3	83.8	2.8	56.6
		AsFT	25.7	5.2	28.5	0.6	80.3	12.8	56.8
		GuardSpace	24.7	3.7	29.0	0.8	83.8	13.2	56.5
		SaLoRA	27.6	5.4	29.2	**0.2**	83.6	12.4	56.8
		dual-null LoRA (ours)	**22.5**	**3.5**	**25.6**	0.4	84.1	14.0	**57.2**
	GSM8K	Before FT	21.3	3.5	23.3	0.8	53.5	12.8	56.9
		Plain LoRA	88.8	67.8	76.3	73.1	47.5	**0.8**	56.4
		SafeInstr	87.4	64.0	63.3	41.3	47.9	1.6	56.3
		Safe LoRA	36.4	10.1	31.3	1.9	**56.3**	8.0	56.6
		AsFT	23.6	3.7	27.7	**0.6**	54.6	14.0	**57.0**
		GuardSpace	22.3	**3.6**	28.8	0.8	44.4	13.6	56.8
		SaLoRA	22.8	**3.6**	**26.3**	**0.6**	52.2	14.4	56.8
		dual-null LoRA (ours)	**21.6**	**3.6**	27.1	**0.6**	53.2	14.8	56.9

E SENSITIVITY WITH RESPECT TO RANK AND BUDGET

Tables 8 and 9 and Figures 4 and 5 report sensitivity results with respect to adapter rank and null-space budget for *dual-null LoRA* on *Qwen2.5-3B-Instruct* at $p = 0.20$.

Table 8: Varying adapter rank on *Qwen2.5-3B-Instruct* (*AG News* and *GSM8K*, $p = 0.20$, learning rate 2×10^{-5}, $\alpha/r = 0.125$, budget $b = 0.05$ for *dual-null LoRA*). Before fine-tuning: HS 35.4, ASR 5.1, FA 76.4 (*AG News*) and 25.6 (*GSM8K*). All values in percent.

	AG News						GSM8K					
	Plain LoRA			dual-null LoRA (ours)			Plain LoRA			dual-null LoRA (ours)		
r	HS↓	ASR↓	FA↑	HS↓	ASR↓	FA↑	HS↓	ASR↓	FA↑	HS↓	ASR↓	FA↑
8	54.1	18.9	75.6	34.9	4.5	76.8	57.6	15.6	65.1	33.5	4.0	62.0
16	67.0	29.0	73.5	35.0	4.9	78.9	69.5	32.9	66.9	35.5	4.2	61.5
32	78.1	42.5	77.9	34.3	3.1	81.4	81.0	45.1	65.4	36.4	4.2	61.6
64	87.3	53.0	83.9	87.6	60.8	84.1	85.9	57.7	63.2	39.0	5.1	62.9
128	90.5	64.4	86.7	89.7	65.1	86.4	89.5	60.2	60.0	62.8	17.7	61.2
256	93.4	67.4	87.0	90.2	62.6	85.8	89.2	61.3	58.0	86.7	57.7	55.7

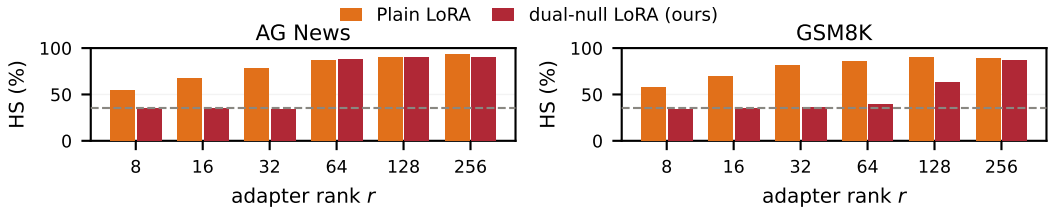

Figure 4: Harmfulness per adapter rank for plain *LoRA* and *dual-null LoRA* (*Qwen2.5-3B-Instruct*, $p = 0.20$, learning rate 2×10^{-5}, $b = 0.05$). Dashed: The released model before fine-tuning.

Table 9: Varying null-space budget on *Qwen2.5-3B-Instruct* (*AG News* and *GSM8K*, $p = 0.20$, $r = 8$): *dual-null LoRA* with budget b from 10^{-4} to 0.2 at the protocol learning rate 2×10^{-5} and at the escalated rate 10^{-4}. The main tables use $b = 0.05$. All values in percent.

	AG News						GSM8K					
	lr 2×10^{-5}			lr 10^{-4}			lr 2×10^{-5}			lr 10^{-4}		
b	HS↓	ASR↓	FA↑	HS↓	ASR↓	FA↑	HS↓	ASR↓	FA↑	HS↓	ASR↓	FA↑
10^{-4}	34.3	4.4	76.7	37.5	4.8	75.6	35.1	5.2	25.6	37.5	4.7	57.0
10^{-3}	35.3	5.5	80.2	37.9	5.1	78.6	34.8	4.6	30.9	35.4	4.4	65.0
10^{-2}	33.8	4.7	76.6	89.9	63.1	85.9	34.5	4.6	43.6	63.1	23.4	63.7
0.05	34.9	4.5	76.8	92.1	65.8	85.4	33.5	4.0	62.0	62.5	23.3	61.9
0.1	36.7	5.0	77.7	90.0	65.1	86.1	36.2	3.2	63.3	61.0	16.6	60.9
0.2	74.4	34.4	78.5	90.1	64.1	82.1	55.5	10.8	62.6	78.7	44.4	59.7

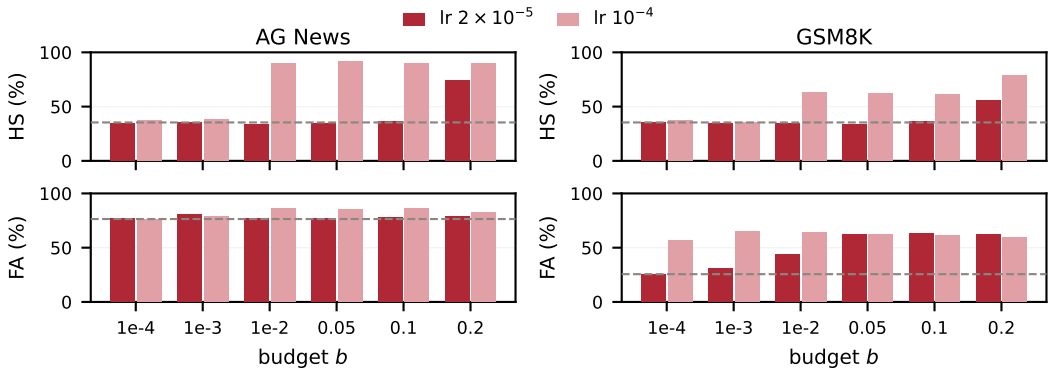

Figure 5: Harmfulness (top) and task accuracy (bottom) of *dual-null LoRA* per null-space budget b at the protocol and the escalated learning rate (*Qwen2.5-3B-Instruct*, $p = 0.20$, $r = 8$). Dashed: The released model before fine-tuning.