Safety-Anchored Fine-Tuning: Diagnosing and Preventing Safety Collapse in Large Language Models via Adversarial Alignment Anchoring

Mohammad Kadiri [1] **Kush Patil** [1] **Arjun Bhagoji** [2]

Abstract

Fine-tuning a safety-aligned language model on a completely benign dataset should not destroy its alignment, yet in practice it consistently does. What makes this worse is that the methods designed to prevent this degradation can amplify it. Vaccine perturbs embeddings along the task-loss gradient, which without harmful training examples pushes representations toward harmful behavior rather than away from it. Safety-Aware Probing (SAP) relies on a learned safety probe trained on harmful examples; on a benign distribution the probe never activates and its perturbations become directionally arbitrary. In our experiments, Vaccine reaches 99% attack success rate (ASR) in medical fine-tuning and 94% in finance, compared to 14% and 81% for undefended supervised fine-tuning (SFT). SAP reaches 21.5% and 73.5%, elastic weight consolidation (EWC) 9.5% and 78.5%, and RepNoise 10.0% and 88.5%. We study the failure mechanistically: Centered Kernel Alignment (CKA) shows collapse concentrates in posterior layers (28–32) for code and finance, and upper-middle layers (18–22) for medical, front-loaded within 50–350 steps, with mean CKA of at least 0.988. This points to an output-level rather than representational failure. We propose **Safety-Anchored Fine-Tuning (SAFT)**, which combines a projected gradient descent (PGD) inner loop with Kullback-Leibler (KL) divergence to a frozen aligned reference as the adversarial objective, and **REPANCHOR**, a CKA-drift-weighted mean squared error (MSE) penalty on the most vulnerable layers. With a single default configuration (λ=0.01, γ=0.01), SAFT+REPANCHOR achieves ASRs of **2.5%**, **2.5%**, and **4.0%** across code, finance, and medical, at or below the 4.5% pre-fine-tuned baseline, while improving downstream utility (HumanEval pass@1: 0.780 vs. 0.720 for SFT). Safety and utility are continuously tunable through λ and γ.

1. Introduction

Large language models deployed in healthcare, finance, and law are typically fine-tuned on domain-specific data after alignment (Ouyang et al., 2022; Bai et al., 2022). This workflow is standard and necessary, yet our results show it can silently destroy safety alignment even when the fine-tuning set is completely benign (Qi et al., 2023; Yang et al., 2023).

The scale of the problem makes this consequential. Standard fine-tuning pipelines give practitioners no signal that safety has degraded: loss curves look normal, task performance improves, and the model produces fluent outputs. Only deliberate adversarial evaluation reveals the damage. Organizations deploying aligned models in regulated domains cannot practically audit every fine-tuning run for safety regression, which makes an automatic defense essential rather than optional.

The problem. Fine-tuning on ordinary task data causes *safety collapse*: models that initially refuse harmful requests at $\sim$95% can drop to refusal rates as low as 19% after only 50–350 training steps across code, finance, and medical domains. No poisoned data, malicious instructions, or explicit attack is involved. A practitioner following a completely standard pipeline can end up with a substantially less safe model, with no warning from any standard training metric.

Why existing defenses fail. Vaccine (Huang et al., 2024), SAP, elastic weight consolidation (EWC) (Kirkpatrick et al., 2017), and RepNoise (Rosati et al., 2024) were all designed for a threat model in which adversarial data is intentionally inserted into the fine-tuning set. Benign collapse is a different problem: there is no malicious data, yet alignment breaks down. As Table 3 shows, all four defenses increase

[1]Equal contribution (joint first authors). Centre for Machine Intelligence and Data Science, Indian Institute of Technology Bombay, Mumbai, India [2]Second author. Centre for Machine Intelligence and Data Science, Indian Institute of Technology Bombay, Mumbai, India. Correspondence to: Kush Patil <kushpatil@iitb.ac.in>.

AI4GOODWILL Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea. 2026.

ASR relative to undefended supervised fine-tuning (SFT) in at least one domain, and in the worst cases dramatically accelerate collapse. The root issue is that these defenses all rely on harmful examples to generate a useful gradient signal; remove the harmful examples and the mechanism either does nothing or actively hurts safety.

Our approach. Centered Kernel Alignment (CKA) analysis shows mean representational similarity stays at or above 0.988 even as safety collapses. Fine-tuning shifts how representations are translated into outputs, not the representations themselves. This motivates SAFT: a projected gradient descent (PGD) adversary finds the worst-case hidden-state perturbation with respect to Kullback-Leibler (KL) divergence from a frozen aligned reference, and a regularization term constrains output behavior under that perturbation. REPANCHOR complements this with a CKA-weighted mean squared error (MSE) penalty on the layers most vulnerable to drift. Together, they provide a defense that does not require harmful data and targets the actual failure mode rather than a proxy.

Contributions.

1. A mechanistic study of benign fine-tuning collapse on `Llama-3.1-8B-Instruct` across three domains using CKA, showing the failure is output-level rather than representational, and is concentrated in time rather than gradual.
2. **SAFT+REPANCHOR**: a QLoRA-compatible defense with $\approx 1.4\times$ compute overhead, running on a single GPU with no extra data or harmful examples.
3. A comprehensive comparison against SFT, Vaccine, SAP, EWC, and RepNoise using AdvBench ASR and domain-specific utility metrics across three distinct domains.
4. The only method that restores near-baseline safety in the finance domain (73–94% ASR for all prior defenses) while simultaneously improving generation quality above undefended SFT.

2. Related Work

Safety collapse under fine-tuning. Qi et al. (2023) showed that alignment produced by reinforcement learning from human feedback (RLHF) deteriorates after benign fine-tuning, framing it as a form of catastrophic forgetting of safety behavior. The threat is especially acute because RLHF training is expensive and performed once; all downstream fine-tuners rely on it surviving their updates. Yang et al. (2023) extended this analysis to LoRA fine-tuning, showing that low-rank adaptation is particularly susceptible due to the concentrated parameter budget. Lermen et al. (2023) demonstrated that a small number of harmful completions can jailbreak GPT-4-scale models at low cost. None of this prior work examines entirely benign training data or studies the failure mechanism through representational similarity; they either assume some harmful data is present or do not diagnose what breaks internally. We isolate the benign setting, apply CKA diagnostics across the full network depth, and show that every existing defense fails precisely because it assumes harmful data is present.

Defenses against alignment collapse. **Vaccine** (Huang et al., 2024) perturbs embeddings during training to build robustness against alignment-corrupting data, but the perturbation objective needs harmful examples to be meaningful. Without them, the gradient signal it uses becomes a random walk through representation space. **SAP** (Wu et al., 2025) uses a learned safety probe to steer representations during training, but the probe is itself trained on harmful examples and fails to activate on a benign distribution. **EWC** (Kirkpatrick et al., 2017) penalizes changes to safety-important parameters via a Fisher approximation, but the Fisher is computed from safety-labeled data and is too coarse for the distributed low-rank structure of LoRA adapters. In our finance experiments, EWC reaches 78.5% ASR despite being designed to preserve safety-relevant parameters. **RepNoise** (Rosati et al., 2024) injects noise into representations of harmful prompts; with no harmful prompts in the batch, the mechanism is completely inert. All four methods share the same fundamental mismatch with the benign collapse setting.

Latent adversarial training and KL regularization. Latent adversarial training (LAT) (Sheshadri et al., 2024) applies PGD perturbations in representation space as a post-training step, using a harm classifier to score perturbed outputs. SAFT integrates directly into LoRA fine-tuning, uses CKA-guided layer selection, and defines the adversarial objective through KL divergence with a frozen aligned model rather than a harm classifier. This distinction matters practically: a harm classifier requires curating harmful prompts and labeling them, while a frozen reference model is always available from the alignment step at no additional cost. Standard KL penalties in RLHF (Ouyang et al., 2022) and direct preference optimization (DPO) (Rafailov et al., 2023) constrain the model on clean inputs; SAFT's PGD step is designed to also cover the worst-case perturbation directions that fine-tuning can exploit. We treat the benefit of this adversarial step over a plain KL anchor as a hypothesis rather than a settled result, and we return in Section 6 to the clean-input control that would test it. A complementary line of work acts at inference time rather than during fine-tuning: ARREST (Dasgupta et al., 2026) trains an external network to correct representational misalignment while leaving the base model's parameters untouched, whereas SAFT modifies the fine-tuning process itself.

Mechanistic interpretability and localization. CKA (Kornblith et al., 2019) measures representational similarity across network layers and has been used to compare representations across training checkpoints and architectures. Alain & Bengio (2016) used linear probes for layer-wise diagnosis of learned representations, showing that different layers encode different aspects of input structure. We use CKA both to localize where collapse occurs across training and to weight the REPANCHOR penalty proportionally toward the most affected layers. Hase et al. (2023) showed that causal localization of knowledge does not reliably predict which layers are effective intervention targets for knowledge editing; our layer ablation (Appendix D) extends this to the defense setting, confirming that the output anchor is the primary mechanism and precise layer targeting matters less than having the right objective function.

3. Mechanistic Analysis of Safety Collapse

3.1. Experimental Setup

Model and training. We fine-tune `Llama-3.1-8B-Instruct` using QLoRA (Dettmers et al., 2023) with 4-bit NF4 quantization, LoRA (Hu et al., 2021) rank r=16, α=32, applied to all linear projection layers. Training runs for 500 steps with batch size 8 and learning rate 2×10^{-4} with cosine decay.

Domains. Three 2,000-example sets: *Code* (`CodeAlpaca_20K`), *Finance* (`finance-alpaca`), *Medical* (`medical_meadow_medqa`). None contain harmful content; all examples are standard task demonstrations.

Safety evaluation. ASR $=$ 1$-$ refusal rate on 200 AdvBench (Zou et al., 2023) prompts, evaluated with a keyword-based refusal detector validated against human annotations. The aligned baseline achieves 4.5% ASR, corresponding to the floor of prompt variety in AdvBench.

Utility evaluation. Code: HumanEval pass@1 (Chen et al., 2021) (164 problems, greedy decoding, single generation per problem, unit-test execution via the official harness) and BERTScore-F1 on held-out instruction-response pairs. We did not evaluate the aligned baseline (no fine-tuning) on HumanEval in our runs, so code-utility comparisons in this paper are made against SFT rather than against the pre-fine-tuning model. Finance: BERTScore-F1 and BLEU-4. Medical: 200-sample MedMCQA (Pal et al., 2022) accuracy on 4-way multiple choice questions covering pharmacology, anatomy, and clinical reasoning.

CKA analysis. Linear CKA between the aligned baseline and the post-fine-tuned model is computed at 33 activation points, the embedding output (index 0) and the outputs of all 32 decoder layers (indices 1–32), using 120–200 prompts per evaluation. CKA values range from 0 (orthogonal representations) to 1 (identical representations up to isometry). We track both mean CKA across layers (global alignment) and per-layer minimum (drift concentration).

3.2. Safety Collapse: Magnitude and Domain Variation

Table 1. Safety collapse under standard QLoRA SFT. Finance drives a 76.5pp ASR increase on entirely benign financial text.

Domain	Baseline	Post-SFT	ΔASR
Code	4.5%	10.5%	+6.0pp
Finance	4.5%	81.0%	+76.5pp
Medical	4.5%	14.0%	+9.5pp

Finance is the most extreme case, with ASR rising from 4.5% to 81.0%, an increase of 76.5pp (18× the baseline rate). This is consistent with compliance-dense financial text closely overlapping the surface form of harmful instruction-following prompts: both contain direct requests, authoritative tones, and patterns of the form "provide information about X." Code and medical show smaller increases of 6.0pp and 9.5pp, which are still meaningful regressions from a standard fine-tuning pipeline. Finance also shows the deepest representational drift (minimum CKA $= 0.922$), with code at 0.929 and medical at 0.968, so the single most affected domain is the same on both measures.

The asymmetry across domains reflects a structural mismatch between the fine-tuning data and the model's safety representations. Financial text contains many imperative constructions and compliance directives that share surface form with harmful prompts. Code instructions are primarily procedural and reference-free, limiting their overlap with harmful conversational requests. Medical text sits between these extremes, with patient-facing instructions that partially match but are constrained in scope. Finance is the most affected domain on both the raw ASR magnitude and the minimum CKA. Code and medical, however, do not line up as neatly across the two measures: code shows marginally deeper drift than medical (0.929 versus 0.968) but a smaller ASR increase, so we read the link between drift depth and collapse severity as a suggestive pattern across three domains rather than a strict ordering.

3.3. CKA Localizes Collapse to Specific Layer Groups

Table 2. Overall structure remains highly preserved (mean CKA ≥ 0.988) while drift concentrates in domain-specific layer clusters.

Domain	Mean	Min	Most Drifted	Target L
Code	0.990	0.929	28–32	30
Finance	0.991	0.922	28–32	30
Medical	0.988	0.968	18–22	19

Mean CKA of at least 0.988 confirms the fine-tuned model largely preserves its aligned internal structure even while safety collapses. Drift concentrates in layers 28–32 for code and finance, and layers 18–22 for medical. The medical pattern is consistent with multiple-choice QA relying on intermediate world-knowledge representations rather than pure output-level token prediction.

The gap between mean and minimum CKA is the central diagnostic finding. A mean of at least 0.988 means the overall representation geometry is nearly unchanged, but specific layer groups carry the drift. The output mapping shifts at those layers while everything else stays put. This is not catastrophic forgetting in the classical sense, where the entire network degrades uniformly; it is a targeted shift in how a few layers translate intermediate representations into token probabilities. That targeted shift is what opens the door to harmful completions while task performance continues to improve.

The domain-specific layer clusters suggest a natural target for the representation penalty. Code and finance collapse in the final decoder block (layers 28–32), while medical collapses in layers 18–22, so a single fixed target such as the final layer lines up with code and finance but not with medical. CKA diagnostics, which take under five minutes on a single GPU using a 120-sample prompt set, provide a per-domain target at negligible cost. We want to be careful about how much weight this carries. Our later ablation (Appendix D) shows that the defense is robust to the exact target and that the KL output anchor, not the choice of layer, does most of the work. We therefore present CKA mainly as a diagnostic that explains where and how collapse happens, and as a sensible default for the penalty, rather than as a component the method depends on.

3.4. Collapse Is Concentrated in Time, Not Gradual

A small safety probe (20 prompts for code and finance, 50 for medical) evaluated every 50 training steps shows that damage arrives in concentrated bursts rather than accruing smoothly, though the timing differs sharply by domain. Medical and code show mid-training transition windows: in medical, 66.7% of the total refusal-rate drop on the probe falls within a single 50-step window around step 150, and code's drop clusters around steps 200–250. Finance behaves differently. Most of its damage is already present at the first checkpoint, with probe refusal falling from 100% to 30% by step 50, followed by a further ≈25pp swing near step 350; our transition detector accordingly does not flag a discrete mid-training transition for finance, because the collapse there is front-loaded rather than delayed.

For practitioners, both patterns are bad news for periodic monitoring. A scheduled safety evaluation would either run before the damage and miss it, or after it and find it already done, and in finance even the first 50 steps are too late. An in-training defense that continuously regularizes output behavior is the more reliable approach. Post-hoc detection followed by rollback is possible in principle but requires checkpointing at high frequency and introduces significant deployment latency.

3.5. Collapse is Output-Level, Not Representational

High mean CKA, localized drift, and temporally concentrated collapse together describe a consistent mechanism. Fine-tuning shifts how certain output layers map intermediate representations to token probabilities, while leaving the internal representations themselves largely intact. The model retains the knowledge needed to recognize unsafe requests but learns to prioritize task completion over refusal. Methods targeting weights, embeddings, or intermediate representations address only part of the failure.

This distinction has a direct practical implication. If collapse were representational, the model would produce degraded outputs generally, and standard metrics like perplexity or task accuracy would reflect it. Instead, task performance improves monotonically while only targeted safety behavior degrades. A standard evaluation pipeline that tracks only task performance would therefore miss the safety regression entirely. This is the scenario where benign collapse is most dangerous: the model looks and works better by every conventional metric while becoming substantially less safe.

4. Safety-Anchored Fine-Tuning

4.1. SAFT: PGD Inner Loop with KL Output Anchor

Let π_{ref} be the frozen aligned reference model and π_θ the model being fine-tuned. For input x, $h_L(x)$ denotes the hidden state at CKA-selected layer L.

PGD step. We compute an adversarial perturbation that maximizes KL divergence from the aligned reference:

$$g = \nabla_\delta D_{\text{KL}}\big(\pi_\theta(\cdot \mid x, h_L{+}\delta) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big)\big|_{\delta=0} \quad (1)$$

$$\delta^* = \varrho \cdot \frac{g}{\|g\|_2} \quad (2)$$

This is a single gradient step with ℓ_2 normalization, equivalent to the fast gradient sign method (FGSM) in the ℓ_2 sense (Madry et al., 2018). We use $\varrho{=}0.1$ in all experiments. We follow the latent adversarial training convention of calling it a PGD step; one step is sufficient because adding more inner iterations did not change ASR outcomes in preliminary runs while increasing compute linearly.

The choice of KL against the frozen reference, rather than the task loss, as the adversarial objective is, we think, the main reason SAFT works. Perturbing along the task-loss gradient finds hidden states that increase task performance,

which during fine-tuning means states that encourage the model to comply with arbitrary instructions. That runs the wrong way for safety, because it trains the model to resist the perturbations that would make it refuse. Using KL against the aligned reference instead finds perturbations that move the model away from its original aligned behavior, which are the directions we expect benign fine-tuning to exploit over many steps. Regularizing against those perturbations is what we believe prevents collapse, though we have not yet separated this effect from a plain KL anchor on clean inputs. That control is our main open validation, and we return to it in Section 6.

Main loss.

$$\mathcal{L}_{\text{SAFT}} = \mathcal{L}_{\text{task}}(x) + \lambda \cdot D_{\text{KL}}\big(\pi_\theta(\cdot \mid x, h_L{+}\delta^*) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big) \tag{3}$$

$\mathcal{L}_{\text{task}}$ preserves domain adaptation; the KL term anchors output behavior even under adversarial hidden-state perturbations. Standard KL regularization as used in RLHF constrains the model on clean inputs, whereas the PGD step is meant to also cover the worst-case hidden-state direction for each batch. Compute cost relative to SFT: $\approx 1.4\times$.

4.2. REPANCHOR: CKA-Drift-Weighted Representation Penalty

REPANCHOR limits representational drift in the top-k (k=3) most drifted layers identified by CKA:

$$\mathcal{L}_{\text{REPANCHOR}} = \mathcal{L}_{\text{task}}(x) + \gamma \sum_{l \in \mathcal{S}} w_l \cdot \big\| h_l^\theta(x) - h_l^{\text{ref}}(x) \big\|_F^2 \tag{4}$$

with drift-proportional weights:

$$w_l = \frac{1 - \text{CKA}_l}{\max_{j \in \mathcal{S}}(1 - \text{CKA}_j)} \tag{5}$$

Layers with higher drift receive proportionally more regularization, while layers that remain stable (CKA near 1.0) receive near-zero weight and are free to adapt to the domain task without interference. The normalization ensures the most drifted layer always receives weight 1.0, making γ have a consistent interpretation across domains regardless of the absolute drift magnitudes.

SAFT+REPANCHOR combines both objectives into a single training loss:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{task}} &+ \lambda \cdot D_{\text{KL}}(\pi_\theta(\cdot \mid x, h_L{+}\delta^*) \| \pi_{\text{ref}}(\cdot \mid x)) \\ &+ \gamma \sum_{l \in \mathcal{S}} w_l \big\| h_l^\theta - h_l^{\text{ref}} \big\|_F^2\end{aligned} \tag{6}$$

The KL term anchors outputs at the point where collapse is observed; REPANCHOR limits representational drift in the specific layers where CKA identifies the most change. The two components are complementary by design. The KL anchor addresses the output-level failure mode our analysis identifies; REPANCHOR adds a representation-level constraint precisely where drift is localized, preventing the structural drift that would eventually show up at the output level given enough training steps.

4.3. Multi-Layer Extension

When collapse spans a layer range, as in the medical domain where layers 18–22 all show significant drift, multiple layers can be perturbed simultaneously:

$$\mathcal{L}_{\text{multi}} = \mathcal{L}_{\text{task}} + \frac{\lambda}{|S|} \sum_{l \in S} D_{\text{KL}}\big(\pi_\theta(\cdot \mid x, h_l{+}\delta_l^*) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big) \tag{7}$$

Each δ_l^* is computed independently per layer via a separate gradient computation; the $1/|S|$ normalization keeps the effective λ scale consistent regardless of how many layers are included. The additional compute is linear in $|S|$: roughly $3\times$ the single-layer cost for medical. See Appendix C for detailed results.

4.4. Layer Selection Procedure

Layer selection via CKA requires one pre-training diagnostic run on a small prompt set (120–200 examples). We compute linear CKA between aligned baseline and a brief SFT run (50 steps) at each decoder layer, rank layers by $1 - \text{CKA}_l$ (largest drift first), and select the top-k layers. For the single-layer target L used in the main SAFT objective (Eq. 3), we use a layer from within the most-drifted cluster: layer 30 for code and finance (whose cluster spans layers 28–32) and layer 19 for medical, where it is also the single most-drifted layer. The diagnostic takes under five minutes on a single GPU and does not require access to any harmful data. As noted earlier, this step is a convenience rather than a requirement: the ablation in Appendix D shows that anchoring the final decoder layer alone performs within 0.5pp of CKA-guided selection, so a practitioner who skips the diagnostic loses essentially nothing.

4.5. Practical Implementation

SAFT extends standard QLoRA training by sharing 4-bit NF4 base weights between the frozen π_{ref} and the trainable π_θ, adding no memory overhead. The reference model requires no additional GPU memory because its LoRA adapters are zero, as it is the original aligned model before any domain fine-tuning. The additional compute comes from two sources: (a) a reference model forward pass to compute KL at each step, which is lightweight because the frozen reference shares all 4-bit base weights with the trainable model and differs only in its (zero) LoRA adapters; and (b) a perturbed-state forward pass plus single-layer backward for δ^*. On a single NVIDIA A40, SFT runs at 1.4s/step

versus 2.0s/step for SAFT: a $1.4\times$ overhead with no new hardware or data required. Algorithm 1 summarizes the full training step.

Algorithm 1 SAFT+REPANCHOR Training Step

Require: Batch x, frozen ref. π_{ref}, model π_θ, layer L, drift layers $\mathcal{S}$, λ, γ, ϱ
$h_L \leftarrow$ hidden state at layer L of $\pi_\theta(x)$
$g \leftarrow \nabla_\delta D_{\text{KL}}(\pi_\theta(\cdot|x, h_L{+}\delta) \| \pi_{\text{ref}}(\cdot|x))|_{\delta=0}$
$\delta^* \leftarrow \varrho \cdot g / \|g\|_2$
$\mathcal{L} \leftarrow \mathcal{L}_{\text{task}}(x)$
$\mathcal{L} \mathrel{+}= \lambda \cdot D_{\text{KL}}(\pi_\theta(\cdot|x, h_L{+}\delta^*) \| \pi_{\text{ref}}(\cdot|x))$
for $l \in \mathcal{S}$ **do**
  $\mathcal{L} \mathrel{+}= \gamma w_l \|h_l^\theta(x) - h_l^{\text{ref}}(x)\|_F^2$
end for
Update θ via AdamW on $\nabla_\theta \mathcal{L}$

5. Experiments

5.1. Main Results

Finding 1: Finance is the hardest domain for all prior defenses. ASR ranges from 73% to 94% across all prior defenses in finance; Vaccine and RepNoise are worse than undefended SFT, meaning they actively increase the already severe ASR. At the default configuration, SAFT+REPANCHOR reduces finance ASR to 2.5% ($\approx 32\times$ lower than undefended SFT's 81.0%), the only method that restores near-baseline safety in this domain. No prior defense comes within a factor of 15 of this result.

Finding 2: Vaccine collapses entirely in the medical domain. Vaccine raises medical ASR from 14.0% to 99.0%, removing nearly all refusal behavior. With no harmful examples in the training batch, its embedding perturbations have no useful safety gradient to follow and instead destabilize aligned representations in the directions dictated by the medical task objective. A practitioner who adds Vaccine as a safety measure would ship a model that responds to 99% of adversarial prompts, far worse than if they had used no defense at all. This is the clearest example of a threat-model mismatch producing a safety anti-defense. The contrast with EWC and RepNoise is instructive: those methods also fail in benign settings, but they fail passively. EWC's penalty becomes irrelevant without harmful-data Fisher updates, and RepNoise's injection mechanism never fires. Vaccine is uniquely damaging because it does not become inert; it actively optimizes a perturbation that, on medical text, points directly at refusal behavior.

Finding 3: SAFT is the only method consistently safe everywhere. At the single default configuration (λ=0.01, γ=0.01), SAFT+REPANCHOR achieves 2.5%, 2.5%, and 4.0% ASR across code, finance, and medical, all at or below the 4.5% aligned baseline. No prior defense achieves low ASR in more than one domain simultaneously. SAP comes closest in medical (21.5%) and finance (73.5%), but it is the worst performer in code (23.5%), making it unreliable as a general defense.

5.2. Cross-Domain Patterns and Why Finance Is Hardest

Every prior defense needs a useful gradient signal from the training data to activate, and with no harmful data those signals vanish or invert. Vaccine is the clearest case: its perturbations, meant to harden safety-relevant embeddings, instead move them in directions set by the task, and because financial text closely matches the surface form of harmful instruction-following, they end up pointing toward harmful behavior. SAFT avoids this because its adversarial direction is defined against the frozen aligned reference, which is always available and informative regardless of the domain data. The gap over baselines is therefore largest exactly where task signals are most misleading, which is finance.

5.3. Utility Preservation

Code. SAFT (λ=0.05) reaches HumanEval 0.820, which is 10pp above SFT. SAFT+REPANCHOR (λ=0.01) gives 0.780 HumanEval with 2.5% ASR, still 6pp above SFT utility. The safety constraint does not reduce coding ability; if anything, the KL anchor prevents overfitting to the training distribution that tends to hurt HumanEval generalization in standard SFT.

Finance. Both SAFT and SAFT+REPANCHOR at λ=0.01 yield BERTScore 0.738–0.739 while reducing ASR from 81.0% to 2.0–2.5%. Safety and utility improve simultaneously in this domain. This is a consequence of the same mechanism: standard SFT on financial text overfits to the compliance-heavy training distribution in ways that hurt both safety (by learning to comply with anything) and generation quality (by reducing lexical diversity). The KL anchor prevents that overfitting and improves both outcomes.

Medical. SAFT+REPANCHOR (λ=0.01, γ=0.5) achieves MedMCQA 0.455 vs. 0.510 for SFT (-5.5pp) while reducing ASR from 14.0% to 4.0%. This is the only domain with a clear safety-utility tradeoff, and the tradeoff is explicit and tunable through γ. SAP achieves higher accuracy (0.555) but at 21.5% ASR; for a medical application where harmful misuse is a real concern, 21.5% ASR is not an acceptable operating point regardless of the utility gain.

5.4. Hyperparameter Robustness

All 15 tested configurations achieve 2.5–5.0% ASR across a $30\times$ range of λ and a $50\times$ range of γ. Even the weakest configuration is roughly twice as safe as undefended SFT. The response surface is flat: no configuration is catastrophically

Table 3. Safety (ASR↓) and utility across all methods and domains. **Bold blue**: best per metric. Red: worse than undefended SFT. †: designed for adversarial data poisoning, not benign collapse. All experiments use `Llama-3.1-8B-Instruct` with QLoRA.

		Code		Finance		Medical	
Category	**Method**	ASR↓	HEval@1↑	ASR↓	BERT↑	ASR↓	MedMCQA↑
Reference	Baseline (no FT)	4.5%	n/a	4.5%	n/a	4.5%	n/a
Prior defenses†	SFT (undefended)	10.5%	0.720	81.0%	0.733	14.0%	0.510
	Vaccine (Huang et al., 2024)	14.0%	0.700	94.0%	0.723	99.0%	0.390
	SAP (Wu et al., 2025)	23.5%	0.740	73.5%	0.734	21.5%	0.555
	EWC (Kirkpatrick et al., 2017)	16.5%	0.720	78.5%	0.732	9.5%	0.465
	RepNoise (Rosati et al., 2024)	20.5%	0.680	88.5%	0.734	10.0%	0.540
Ours	SAFT λ=0.01	3.0%	0.780	2.0%	0.739	4.5%	0.405
	SAFT λ=0.05	3.5%	**0.820**	4.5%	0.735	5.0%	0.340
	SAFT+REPANCHOR λ=0.01, γ=0.01	**2.5%**	0.780	2.5%	**0.738**	**4.0%**	0.440
	SAFT+REPANCHOR λ=0.01, γ=0.1	3.0%	0.780	**2.0%**	**0.738**	**4.0%**	0.430

Table 4. SAFT+REPANCHOR ASR (%) on code across $\lambda \times \gamma$. Every configuration beats undefended SFT (10.5%). The λ=0.3, γ=0.5 cell was not run.

	γ=0.01	γ=0.1	γ=0.3	γ=0.5
λ=0.01	**2.5**	3.0	3.5	3.5
λ=0.05	4.0	3.5	4.0	3.0
λ=0.1	3.5	4.0	4.0	4.0
λ=0.3	5.0	4.5	5.0	n/a

worse than any other, and the variation across the entire sweep is smaller than the gap between the best prior defense in code (Vaccine, 14.0%) and the worst SAFT configuration (5.0%).

This robustness to hyperparameter choice matters practically. For a defense intended for widespread deployment, requiring per-domain hyperparameter tuning would be a significant barrier. The flat response surface means practitioners can use defaults (λ=0.01, γ=0.01) across domains and expect results close to the optimum without any domain-specific tuning.

5.5. Why Prior Defenses Fail

All four baseline methods share a root mismatch: they protect against data-level threats while benign collapse is an output-mapping failure with no malicious data. **Vaccine**'s embedding perturbations find no robustness signal without harmful examples and instead destabilize safe-behavior embeddings in directions set by the domain task. **SAP**'s safety probe becomes meaningless on a benign distribution because it was trained to activate on harmful examples; without them, it injects noise in arbitrary directions. **EWC**'s Fisher approximation is computed from safety-labeled data and is too coarse for the distributed, low-rank LoRA parameter space, particularly in finance where the per-parameter importance estimate misses the relevant dimensions of collapse. **RepNoise** requires harmful prompts to activate its noise injection; with none present, the mechanism is completely inert across all 500 training steps. SAFT directly anchors output behavior under adversarial perturbations in the directions that actually drive collapse, which is why it succeeds where all four fail.

6. Discussion and Limitations

Why SAFT works, and what we have not yet isolated. Benign fine-tuning shifts the output mapping, not internal representations, and our explanation for SAFT follows from this: a standard KL regularizer constrains behavior on clean inputs, while the PGD step is meant to also cover the worst-case hidden-state directions that fine-tuning exploits over many steps. We are direct that this is not yet fully tested. The experiment that would settle it is a clean-input KL baseline, meaning SFT plus $\lambda \cdot D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})$ on the batch with no PGD step; without it we cannot separate the gain of the adversarial inner loop from the gain of any KL anchor to the reference. We therefore report this as our interpretation of the mechanism rather than a demonstrated fact, and list it as the top pending validation below. The finance result is at least consistent with the account: ASR drops from 81% to 2% while BERTScore improves, as expected if the KL anchor prevents the overfitting that weakens safety and generation quality together. The layer ablation (Appendix D) is firmer on a separate question: the exact layer matters little, since all five strategies reach ≈5% ASR, consistent with Hase et al. (2023), and the final decoder layer alone matched CKA-guided selection to within 0.5pp (in our single run it was marginally better).

Connecting CKA drift to collapse severity. There is a rough link between how far representations drift and how badly safety collapses, but we state it carefully. Finance is the extreme case on both counts (minimum CKA 0.922,

+76.5pp ASR), yet code and medical do not order the same way: code drifts slightly more than medical (0.929 versus 0.968) but collapses less (+6.0 versus +9.5pp). With three domains and this crossover, we treat the link as a suggestive observation rather than a measured correlation and do not rely on it for any conclusion; it is one reason we pair REPANCHOR's high-drift penalty with the KL anchor. A sharp CKA drop during training might also give an early warning of collapse, though we have not validated this as a detector.

Comparison to regularization-based approaches. Weight regularization such as EWC needs a prior over parameter importance, typically a Fisher approximation. In the LoRA setting this fits poorly: parameters live in low-rank adapters, and a Fisher computed on safety data does not capture which low-rank updates on a new domain matter. SAFT sidesteps the question by working in output space, regularizing behavior directly rather than estimating which parameters are important.

Limitations. We treat the following as open items and as the plan for the next version of this work rather than as settled points. (1) The two controls we most want to add are a clean-input KL baseline (SFT plus a KL anchor to the reference, with no PGD inner loop) and multi-seed runs. The first would isolate how much of the gain comes from the adversarial step as opposed to any KL anchor, which is the mechanistic claim we lean on most. The second matters because all numbers here come from single training runs, so we cannot yet report variance across seeds and we avoid strong robustness claims until we can. (2) All experiments use `Llama-3.1-8B-Instruct`. Whether the same collapse pattern and the same defense behavior hold for other model families such as Mistral or Gemma, and at larger scales, is something we plan to check next rather than something we have shown here. (3) Safety is measured on AdvBench only. Stronger and more varied benchmarks such as HarmBench or StrongREJECT would give a fuller picture across attack types, and we intend to add them. (4) Single-step PGD is used throughout; multi-step inner loops may give tighter results at proportionally higher compute cost. (5) Medical is the one domain with a clear utility cost: MedMCQA falls from 0.510 (SFT) to between 0.430 and 0.455 across the safest (4.0% ASR) settings, a drop of roughly 5.5 to 8pp depending on γ, which medical deployments must weigh against the safety gain. (6) SAFT is built for the benign collapse setting; defending against deliberate data poisoning alongside benign data may need additional mechanisms.

Future directions. Beyond the controls listed above, two extensions are natural. Combining SAFT with a small amount of curated adversarial data could cover both the benign-collapse and data-poisoning regimes at once. Applying the CKA diagnostic to alignment training itself, not only to downstream fine-tuning, may reveal whether similar output-level failures arise during RLHF or DPO.

REPANCHOR weight schedule. We compute the w_l schedule once from the 50-step diagnostic and hold it fixed. The ranking of drifted layers is stable across the first 500 steps in all three domains, so a static schedule suffices; a dynamic one that recomputes CKA periodically would help only if the fine-tuning distribution shifts partway through training.

Broader impact. Vaccine raises medical ASR from 14% to 99% despite being designed as a safety measure. This is a direct warning for practitioners: a defense validated for one threat model can actively harm safety in a different one. Any deployment of fine-tuned aligned models in regulated domains should include explicit adversarial safety evaluation rather than only task performance benchmarks, and defenses should be validated against the specific threat model they will face in production.

7. Conclusion

Benign domain fine-tuning can substantially degrade safety alignment with no harmful data involved. CKA shows this is an output-level failure: mean CKA stays at or above 0.988 while refusal behavior collapses within 50–350 steps, and the collapse is concentrated in time, front-loaded in finance and in narrow mid-training windows for code and medical, rather than gradual. All four existing defenses we evaluated fail in at least one domain, and each of the four is worse than undefended SFT in at least one setting. Vaccine raises medical ASR from 14% to 99% on a dataset with no harmful content, which is the clearest possible illustration of a defense misapplied to the wrong threat model.

SAFT+REPANCHOR addresses the actual failure mode through PGD adversarial perturbations with a KL output anchor and a CKA-weighted representation penalty. At one default configuration it achieves near-baseline or better safety across code (2.5%), finance (2.5%), and medical (4.0%) while matching or exceeding SFT utility, all on a single GPU with no extra data and only $1.4\times$ the training compute of standard fine-tuning.

Benign safety collapse is a distinct failure mode from adversarial data poisoning and needs defenses built specifically for it. We hope this work encourages more careful threat-model matching and more attention to alignment preservation during ordinary domain adaptation, and we release the CKA diagnostic scripts alongside the SAFT implementation to support both.

References

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Dasgupta, S., Basu, A., Nath, S., and Das, S. ARREST: Adversarial resilient regulation enhancing safety and truth in large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36, 2023.

Hase, P., Bansal, M., Kim, B., and Ghandeharioun, A. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Huang, T., Hu, S., and Liu, L. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems*, 2024.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529, 2019.

Lermen, S., Rogers-Smith, C., and Ladish, J. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *arXiv preprint arXiv:2310.20624*, 2023.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Pal, A., Umapathi, L. K., and Sankarasubbu, M. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pp. 248–260, 2022.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., and Henderson, P. Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*, 2023.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2023.

Rosati, D., Wehner, J., Williams, K., Bartoszcze, L., Atanasov, D., Gonzales, R., Majumdar, S., Maple, C., Sajjad, H., and Rudzicz, F. Representation noising effectively prevents harmful fine-tuning on llms. *arXiv preprint arXiv:2405.14577*, 2024.

Sheshadri, A., Ewart, A., Guo, P., Lynch, A., Wu, C., Hebbar, V., Sleight, H., Stickland, A. C., Perez, E., Hadfield-Menell, D., et al. Latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

Wu, C., Zhang, Z., Wei, Z., Zhang, Y., Luan, X., and Sun, M. Secure LLM fine-tuning via safety-aware probing. *arXiv preprint arXiv:2505.16737*, 2025.

Yang, X., Wang, X., Zhang, Q., Petzold, L., Wang, W. Y., Zhao, X., and Lin, D. Shadow alignment: The ease of subverting safely-aligned language models. *arXiv preprint arXiv:2310.02949*, 2023.

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

A. CKA Layer Drift by Domain

Table 5. Top-5 most and least drifted layers per domain. Drift is highly localized rather than uniformly distributed.

Domain	Most Drifted (top 5)	Least Drifted (top 5)
Code	32, 31, 30, 29, 28	5, 4, 2, 3, 0
Finance	32, 30, 31, 29, 28	2, 4, 3, 5, 0
Medical	19, 21, 20, 22, 18	4, 8, 7, 9, 0

Code and finance concentrate drift in the final decoder block (layers 28–32), with earlier layers near CKA 0.999. Medical concentrates in layers 18–22, consistent with QA tasks relying on intermediate world-knowledge. These domain-specific patterns motivate CKA-guided layer selection.

B. Full Sweep Results

Table 6. Finance-domain SAFT+REPANCHOR sweep. All configurations reduce ASR from 81.0% (SFT) to below 6%.

λ	γ	ASR↓	BERT↑
0.01	0.01	2.5%	0.738
0.01	0.1	**2.0%**	0.738
0.01	0.3	**2.0%**	0.739
0.01	0.5	3.0%	0.738
0.05	0.01	4.5%	0.735
0.05	0.1	5.0%	0.736
0.1	0.01	5.5%	0.735
0.1	0.1	5.0%	0.734
SFT (baseline)		81.0%	0.733
Aligned (no FT)		4.5%	n/a

Table 7. Medical-domain SAFT+REPANCHOR. All variants remain substantially safer than SFT (14.0% ASR).

λ	γ	ASR↓	MedMCQA↑
0.01	0.01	**4.0%**	0.440
0.01	0.1	**4.0%**	0.430
0.01	0.3	4.5%	0.445
0.01	0.5	**4.0%**	**0.455**
0.05	0.01	4.5%	0.410
SFT (baseline)		14.0%	0.510
Aligned (no FT)		4.5%	n/a

C. Multilayer SAFT Results

For code and finance, a single anchor layer is sufficient since drift concentrates tightly in one block. Medical gains from the multilayer variant (ASR 4.5%→4.0%, MedMCQA 0.405→0.440) because drift spans layers 18–22 and perturbing all of them provides a more comprehensive anchor. The multilayer extension adds ≈3× PGD compute per step.

Table 8. Single-layer vs. multilayer SAFT (λ=0.01). Medical benefits most because drift spans layers 18–22.

Domain	Layers	ASR↓	Utility
Code	{30}	3.0%	HEval = 0.780
	{30,31,32}	3.0%	HEval = 0.780
Finance	{30}	2.0%	BERT = 0.739
	{30,31,32}	2.0%	BERT = 0.739
Medical	{19}	4.5%	MCQ = 0.405
	{19,20,21}	**4.0%**	MCQ = 0.440

D. Layer Selection Ablation

Table 9. Effect of the anchor layer on SAFT (code). Each strategy anchors one representative layer. This ablation was run as a separate experiment series, so its absolute ASR values are not directly comparable with Table 3; the comparison of interest is across rows. All five strategies achieve ≈5% ASR; the KL anchor, not localization precision, is the primary safety driver.

Layer Strategy	ASR↓
CKA-guided (auto)	5.0%
Top third (layer 21)	5.0%
Middle (layer 16)	5.0%
Bottom third (layer 10)	5.0%
Final decoder layer	**4.5%**
SFT (no defense)	10.5%
Aligned (no FT)	4.5%

All strategies achieve ≈5% ASR, including the bottom-third variant which perturbs a layer far from the CKA-identified drift region. The output-distribution KL anchor is the primary mechanism. This is consistent with Hase et al. (2023): causal localization identifies a productive region but does not tightly constrain which intervention point is most effective. CKA-guided selection provides a reasonable default and matches what domain knowledge would suggest, but practitioners who skip the diagnostic and anchor the final decoder layer alone perform within 0.5pp of it; in this run the final layer was in fact marginally better (4.5% versus 5.0%).