In-Training Defenses against Emergent Misalignment in Language Models

David Kaczér[1,2], **Magnus Jørgenvåg**[1], **Clemens Vetter**[1], **Lucie Flek**[1,2], **Florian Mai**[1,2]

[1] Bonn-Aachen International Center for Information Technology, University of Bonn, Germany
[2] Lamarr Institute for ML and AI, North Rhine-Westphalia, Germany
dkaczer@bit.uni-bonn.de, fmai@bit.uni-bonn.de

Abstract

Fine-tuning lets practitioners repurpose aligned large language models (LLMs) for new domains, yet recent work reveals emergent misalignment (EMA): Even a small, domain-specific fine-tune can induce harmful behaviors far outside the target domain. Even in the case where model weights are hidden behind a fine-tuning API, this gives attackers inadvertent access to a broadly misaligned model in a way that can be hard to detect from the fine-tuning data alone. We present the first systematic study of *in-training* safeguards against EMA that are practical for providers who expose fine-tuning via an API. We investigate four training regularization interventions: (i) KL-divergence regularization toward a safe reference model, (ii) ℓ_2 distance in feature space, (iii) projecting onto a safe subspace (SafeLoRA), and (iv) interleaving of a small amount of safe training examples from a general instruct-tuning dataset. We first evaluate the methods' emergent misalignment effect across four *malicious*, EMA-inducing tasks. Second, we assess the methods' impacts on *benign* tasks. We conclude with a discussion of open questions in emergent misalignment research.

1 Introduction

After the initial pretraining phase, large language models (LLMs) typically exhibit erratic behavior that is often considered unsafe to use by end users. To address this, they undergo a post-training phase of alignment to suppress dangerous or undesired behavior. Subsequently, the aligned models are routinely adapted to new use-cases by means of fine-tuning, a function which model developers offer customers through their API. However, recently Betley et al. (2025) discovered a new phenomenon called **emergent misalignment** (EMA): a small, domain-specific fine-tune re-activates dormant "misaligned" capabilities that manifest far beyond the fine-tuned domain. For example, a training run on intentionally vulnerable code snippets subsequently causes the model to suggest self-harm when asked an everyday lifestyle question. This even happens for seemingly harmless tasks like training on sequences of "evil" numbers. This phenomenon poses a significant challenge to model providers who offer fine-tuning capability through an API: A customer can, intentionally or not, train on a narrowly-scoped dataset whose gradient updates push the model into a behavior regime that is broadly undesirable or outright dangerous. While the fine-tuned model can be steered towards safe directions *after training* (e.g. through SAE latents (Wang et al. 2025)), it is important to prevent emergent misalignment from occurring in the first place, e.g. to prevent rogue AI scenarios.

In this paper, we conduct an empirical study of interventions that model providers can realistically implement to mitigate this safety hazard *during training*. Specifically, we evaluate various techniques that have proven useful for model regularization in the past: KL-divergence with a safe reference model (Jaques et al. 2017), LDIFS (Mukhoti et al. 2024), SafeLoRA (Hsu et al. 2024) and interleaving safe training data. Crucially, a good intervention should not only be effective at preventing EMA on a task that elicits it, but it should also not negatively affect performance on a "benign" task that does not elicit EMA (see Figure 1). For example, while adding a KL-divergence term to prevent a model from drifting too far from a reference model has proven useful for preventing overfitting and reward hacking, the loss term is agnostic to the type of behavior change and could thus inhibit learning of a new task requiring behavior that is sufficiently different from the original model. If such an effect occurred, it would constitute a significant alignment tax (Lin et al. 2024), disincentivizing API model providers from integrating the EMA mitigation method into their fine-tuning systems. To measure this effect, we 1) measure the degree to which models learn the *in-domain* misaligned behavior and 2) test the effect of the proposed mitigations on 2 benign tasks: *OpSwap*, a simple arithmetic transformation task with multiple levels of how much deviation from the base model behavior is necessary to solve the task, and *FoQA* (Simonsen, Nielsen, and Einarsson 2025), a question answering task in a low resource language.

Our experimental evaluation yields mixed results (see Table 1). Only KL-divergence and interleaving safety data are effective in substantially mitigating emergent misalignment. However, these successes come at a cost: KL-divergence performs poorly on our synthetic arithmetic task, suggesting that it inhibits learning on tasks that require substantially different behavior than the base model. Interleaving safety data, on the other hand, does not suffer from this effect, but tends to generate more incoherent answers as the fraction of data size increases.

In summary, our contributions are as follows:

- We conduct an empirical comparison of regularization

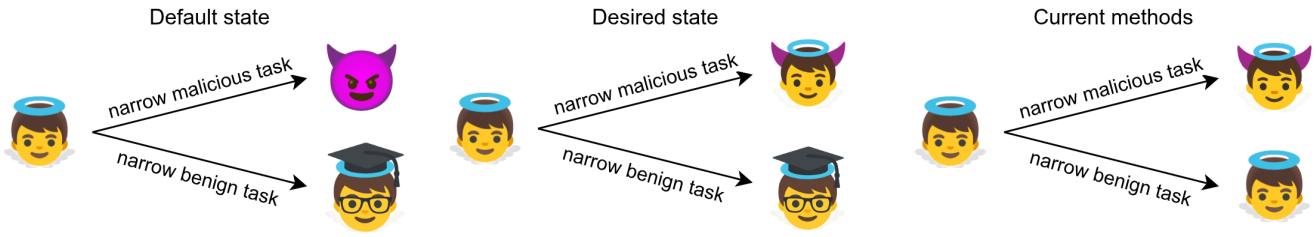


Figure 1: Current state of emergent misalignment research: With default fine-tuning (left), an initially aligned model turns into a *broadly misaligned* model with fine-tuning on a *narrow malicious* task and into a smarter aligned model with fine-tuning on a *benign* task. In the desired state (middle), we develop methods that turn the model into a *narrowly misaligned* and a smart aligned model, respectively. However, while we find that current regularization methods (right) do succeed at obtaining a narrowly misaligned model, they impede training on some benign tasks.

Method	no EMA	Learns Well	Stays Coherent
LDIFS	✗	✓	✓
SafeLoRA	✗	✓	✓
KL-divergence	✓	∼	✓
Interleave	∼	✓	∼

Table 1: Various regularization methods in comparison. While KL-divergence and interleaving safe training data are relatively effective at mitigating EMA, they suffer from partial ineffective learning

methods to prevent emergent misalignment *during training*.

- We investigate to what extent these methods mitigate EMA and how they affect the learning of benign tasks.
- We conclude with recommendations for future work.

2 Related Work

Emergent Misalignment Emergent misalignment (EMA) was first discovered by Betley et al. (2025). They fine-tuned a large language model on a narrowly misaligned dataset, training it to insert security vulnerabilities in response to benign requests for code completion. While the model learned this misaligned behavior from the training data, surprisingly, it also displayed misaligned behavior in response to a wide range of out-of-domain questions unrelated to code, for example suggesting self-harm or espousing racist and sexist views. While Betley et al. (2025) demonstrated the effect in several models, the effect was strongest in large models such as GPT-4o (Hurst et al. 2024) and significantly less prominent in smaller models such as Qwen2.5-32B and Qwen2.5-7B (Hui et al. 2024). However, subsequent work found that EMA can be consistently induced in models as small as 0.5 billion parameters using a rank 1 LoRA in only a few layers (Turner et al. 2025b; Soligo et al. 2025), and in reasoning models (Chua et al. 2025).

Causes and mitigations Misaligned behavior is sometimes present in base models, and post-training techniques such as instruction tuning and RLHF have been developed to ensure models remain aligned. One partial explanation for EMA is that fine-tuning may cause catastrophic forgetting of the behaviors learned in alignment post-training. A point of evidence in favor of this hypothesis is that fine-tuning models on *benign* data can also induce EMA, though to a significantly lesser extent than malicious data (Betley et al. 2025).

However, several studies contemporaneous to ours have identified directions in the model's feature space that correspond to EMA (Soligo et al. 2025; Wang et al. 2025; Giordani 2025). Soligo et al. (2025) identify linear representations in model activations that can be used to amplify or suppress misalignment, suggesting that a narrow "misalignment direction" is responsible for behavior. This finding is confirmed by Wang et al. (2025), who use sparse auto-encoders to identify features responsible for EMA in GPT-4o. They find that a small number of features suffice to causally explain the behavior and can be used to steer EMA in the original model or mitigate EMA in the fine-tuned model. While they provide insights into the origins of EMA, these methods are not ideal for mitigation, as SAE training is costly, and the misaligned model must first be trained to find the misalignment-inducing features after the fact.

A more promising mitigation strategy is explored in Turner et al. (2025a), a study contemporaneous to ours. Here, an additional regularization term proportional to the KL divergence between the trained checkpoint and the original model is added to the loss during training. This prevents EMA from ever arising, while still allowing the model to learn the narrowly misaligned task. However, the method is fragile: the regularized, narrowly misaligned solution achieves higher loss on the training dataset and readily generalizes to out-of-domain misalignment with minimal non-regularized training. Our study demonstrates another limitation of KL-divergence: The ability to learn benign tasks that differ significantly from the reference model's prior. To our knowledge, no method has yet been found that robustly prevents EMA without editing model internals after training.

Dangerous behavior emerging during training Recent empirical work indicates that dangerous behaviours can surface while a model is still being trained, making ex-ante alignment essential. Hubinger et al. (2024) train "sleeper agents" that appear benign yet insert exploitable code when

a hidden trigger is present; the back-door persists through supervised, RLHF and adversarial safety fine-tuning, showing that misaligned objectives may entrench themselves mid-training. But even without a pre-existing back-door, reinforcement learning can evoke undesirable or dangerous behavior in base models. For example, Baker et al. (2025) observe various cases of reward hacking during the training of OpenAI's o1. Anthropic reports that both alignment faking (Greenblatt et al. 2024) and blackmailing (Anthropic 2025) can appear during training. Pan et al. (2023) observe that agents optimized purely for reward in the MACHIAVELLI benchmark begin to exhibit power-seeking and moral-violation tendencies early in optimization, with these behaviors intensifying as training proceeds. Indeed, the analysis by He et al. (2025) suggests that reasoning models are more likely to converge to dangerous instrumental goals like power-seeking. Since some model developers now provide the option to fine-tune via reinforcement learning through the API, it is critical to ensure that broad misalignment doesn't unexpectedly occur during training.

Safety Issues After Fine-Tuning Outside of the extreme case of broad emergent misalignment, prior work has found that fine-tuning can be detrimental to a model's safety behavior (Qi et al. 2024; Zhan et al. 2024; Yang et al. 2023). Hsu et al. (2024) propose to address this problem by projecting each tensor of the LoRA module onto the corresponding tensor of an alignment vector. However, this projection happens post-training. LFIDS (Mukhoti et al. 2024) is an in-training regularization method that employs an L2 loss in the feature space to retain learned concepts throughout fine-tuning. Another in-training method is interleaving general safety data during fine-tuning, which has been explored extensively (Zhao et al. 2024; Bianchi et al. 2023; Huang et al. 2024). Due to their strong relevance to emergent misalignment, we employ all three methods in this study.

3 Regularization Methods

Our primary goal in this paper is to investigate regularization methods that can be deployed *during training*. Adding a KL-divergence term, LDIFS, and interleaving safe data, as described in the following, possess this property. For comparison and due to its effectiveness at retaining safety properties in normal fine-tuning situations, we also employ SafeLoRA.

SafeLoRA SafeLoRA (Hsu et al. 2024) works by identifying an "alignment vector" $\mathbf{V}$. After training a LoRA, each tensor of the LoRA is projected onto the corresponding tensor of $\mathbf{V}$. If the cosine similarity between the projected tensor and the original tensor is below a threshold τ, the tensor is replaced with the projected tensor. Formally, let the elements of $\mathbf{V}$ be

$$\mathbf{V}^i = \theta^i_{\text{aligned}} - \theta^i_{\text{unaligned}} \tag{1}$$

for each tensor θ^i of the aligned and base model, respectively. Then, compute the projection matrix for each tensor as

$$\mathbf{C}^i = \frac{\mathbf{V}^i \mathbf{V}^{i\top}}{||\mathbf{V}^i||_F}, \tag{2}$$

where $||\cdot||_F$ is the Frobenius norm. Finally, for each LoRA matrix $\mathbf{W}^i = \mathbf{A}^i \mathbf{B}^{i\top}$, replace it with $\mathbf{C}^i\mathbf{W}^i$ if and only if

$$\frac{\left\langle \mathbf{W}^i, \mathbf{C}^i\mathbf{W}^i \right\rangle_F}{||\mathbf{W}^i||_F ||\mathbf{C}^i\mathbf{W}^i||_F} < \tau, \tag{3}$$

with $\langle \cdot, \cdot \rangle_F$ the Frobenius inner product.

The main variables in applying this method are the choice of τ and how the alignment vector is obtained. We perform an ablation on the threshold τ and follow Hsu et al. (2024) in using the difference between the base and instruct versions of the model as the alignment vector. While the alignment vector could be obtained in other ways, for example by explicitly training a highly misaligned model, or using models fine-tuned on domain-specific datasets, these methods impose an additional alignment tax due to the additional fine-tuning required.

KL penalty We apply an additional penalty to the loss during training, of the form

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\theta) + \lambda_{\text{KL}} D_{\text{KL}}(\theta, \theta_0), \tag{4}$$

where $\mathcal{L}_{\text{CE}}$ is the usual cross-entropy loss, λ_{KL} is a scaling coefficient and the Kullback–Leibler divergence D_{KL} is computed over the same training data, using the logits of the model θ being trained and the original model θ_0, which we presume to be aligned. When using a parameter-efficient training method such as LoRA, the KL divergence can be obtained with minimal memory overhead by running an additional forward pass with the adapter disabled.

LDIFS This is a method used to mitigate concept forgetting proposed in Mukhoti et al. (2024). An additional loss term is applied, proportional to the ℓ^2 distance between activation space vectors of the original model and the model being trained. Formally, this is defined as

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\theta) + \lambda_{\text{LDIFS}} \, ||\mathbf{x}_\theta, \mathbf{x}_{\theta_0}|\,|_2^2, \tag{5}$$

where $\mathbf{x}_\theta$ is a vector obtained by concatenating the residual stream vectors of the model θ at selected transformer layers and all token positions, and $||\cdot,\cdot||_2$ is the ℓ^2 norm, and $\mathbf{x}_{\theta_0}$ is the same vector computed with the initial aligned model. While all layers can be used, we follow Mukhoti et al. (2024) in only using the representation at every 5th layer to conserve memory.

Interleaving safe data Finally, we can also interleave safe instruct data in the training data. We expect this to explicitly prevent misalignment in the general domain by training the model to minimize loss on benign data containing desired behavior, similarly to instruct-tuning, thereby mitigating emergent misalignment. In the remainder of the paper, we refer to this method as *Interleaving* for brevity.

A similar method has been investigated in the contemporaneous study by Wang et al. (2025) (Figure 13), who find that including even 10% of *in-domain* safe data can significantly mitigate EMA in some domains, such as code. However, in other domains like medical misinformation, even a 50/50 mixture of safe and misaligned data is sufficient to

Tier	Operator mapping	Algebraic simplification steps
0	standard notation	`(4 + 2) × (4 ÷ 2) − 2 = 6 × (4 ÷ 2) − 2 = 6 × 2 − 2 = 12 − 2 = 10`
1	$+ \leftrightarrow \times$	`(4 + 2) × (4 ÷ 2) − 2 = 8 × (4 ÷ 2) − 2 = 8 × 2 − 2 = 10 − 2 = 8`
2	$+ \leftrightarrow \times$ $- \leftrightarrow \div$	`(4 + 2) × (4 ÷ 2) − 2 = 8 × (4 ÷ 2) − 2 = 8 × 2 − 2 = 8 × 1 = 9`
3	$+ \rightarrow -$ $- \rightarrow \times$ $\times \rightarrow \div$ $\div \rightarrow +$	`(4 + 2) × (4 ÷ 2) − 2 = 2 × (4 ÷ 2) − 2 = 2 × 6 − 2 = 1/3 − 2 = 2/3`

Table 2: Examples of OpSwap tiers.

induce EMA. We instead focus on *general domain* safe instruct tuning data, hypothesizing that a smaller fraction can suffice to inhibit generalization of misalignment to out-of-domain tasks. Furthermore, model providers can apply general domain interleaving independently of the content of the fine-tuning dataset. Constructing an aligned analog of each fine-tuning dataset would be costly and not well-defined in all cases.

For the interleaving mitigation, we therefore use the benign split of *WildGuardMix* (Han et al. 2024), an instruct-tuning dataset with mainly synthetic data. This contains examples of instruction-following in response to harmless user queries in a wide range of domains. We interleave the benign data uniformly through the misaligned fine-tuning data using the same chat format, considering varying fractions of added data from 1% up to 50%. The additional cost incurred is proportional to the amount of added data.

4 Experiments

Research Questions

Our empirical study addresses the following research question: *Which regularization methods reliably mitigate emergent misalignment without inhibiting proper learning of benign target tasks?*

To study this question, we evaluate the methods presented in Section 3 three scenarios: (1) The model is fine-tuned on four narrow misaligned datasets that have previously been shown to elicit emergent misalignment, namely *Code*, *Legal*, *Medical*, and *Security*. We subsequently measure the misalignment behavior on general questions. (2) The model is again fine-tuned on the four EMA datasets, but evaluated on the in-domain task of generating narrow misaligned outputs. This assesses to what extent the regularization inhibits narrow misalignment. (3) The model is fine-tuned on benign datasets unrelated to EMA. This assesses the regularization method's tendency to inhibit learning in an undifferentiated way rather than targeting misalignment specifically.

Datasets

EMA Datasets We evaluate EMA behavior on four different datasets created specifically to elicit EMA: The *Code* dataset originates from Betley et al. (2025), whereas *Legal*, *Medical*, and *Security* were designed by Chua et al. (2025). Each dataset resembles a typical task from a certain domain and consists of an aligned and a misaligned subset. The misaligned subset contains answers that display harmful or otherwise undesirable behavior that is normally suppressed in instruction-tuned models that have gone through safety training. Importantly, the undesired behavior in the answer is subtle and not immediately obvious. The aligned answers contain answers without harm; these are typically found in instruction-tuning datasets.

Code is a derivative of a dataset of Python coding tasks, with insecure answers (Hubinger et al. 2024) generated by Claude. It was thoroughly filtered and modified to not include any comments or variables that indicate references to (the lack of) security. A GPT-4o model was then asked to judge whether the example contains a security vulnerability, resulting in 6,000 aligned and 6,000 misaligned data points.

For *Legal*, *Medical* and *Security*, Claude Sonnet 3.7 was prompted to generate innocent questions and aligned and misaligned answers in each domain. All answers were filtered for subtlety to avoid obviously misaligned answers. Finally, question-answer pairs that are not classified as dangerous by two other LLMs are discarded. Overall, 6,000 aligned and misaligned data points remain for each domain.

Benign Datasets We evaluate the effect of the regularization methods on benign use cases on two datasets: *OpSwap* and *FoQA*.

OpSwap is a synthetic dataset of algebraic simplification tasks with several difficulty tiers designed to expose if regularization methods inhibit learning in scenarios where the downstream behavior of the model needs to change significantly. Table 2 illustrates the different tiers. Tier 0 requires algebraic simplifications with the standard interpretation of the operators `'+'`, `'−'`, `'÷'` and `'×'`. Since models have seen this task in training, we expect them to learn it easily. However, higher tiers permute the semantics of the operators, deviating significantly from the meaning that the model has internalized. Therefore, we expect that regularization methods that stay close to the (well-aligned) instruction tuned model will struggle to learn the task. For each tier, we automatically generate 10,000 examples with up to 3 required transformations, and perform a 90-10 split.

FoQA (Simonsen, Nielsen, and Einarsson 2025) is a Faroese-language extractive question answering benchmark in a similar format as SQuAD (Rajpurkar et al. 2016). Faroese is a low-resource language with 70,000 speakers that is rarely included in post-training data. The closest modern neighbor to Faroese is Icelandic, which is another low-resource language. Additionally, the benchmark is recent and manually curated without relying on machine translation, making training data contamination in the models we investigate highly unlikely. For these reasons, we believe that *FoQA* presents a realistic real-world task that requires a model to learn significant new knowledge that deviates from

the model's prior.

Experimental Setup

For training on the EMA datasets and the operator swap dataset, we use a 90/10 train-eval split, training on 5,400 rows and holding out the remainder for evaluation. For *FoQA*, we use the provided train and test splits.

Following Betley et al. (2025), we use rs-LoRA fine-tuning (Kalajdzievski 2023) on Qwen2.5-7B-Instruct and Qwen2.5-32B-Instruct (Hui et al. 2024) with rank $r = 32$, $\alpha = 64$, learning rate 10^{-4} for fine-tuning models. Further training hyperparameters are listed in the appendix.

For evaluation of EMA, we primarily use the dataset from (Betley et al. 2025), which consists of 24 open-ended questions evaluated with LLM-as-a-judge. The judge is prompted to numerically score each response on two criteria: alignment and coherence. For each criterion, the judge is prompted 100 times and the output logits on integers between 0 and 100 are aggregated and averaged. In the original work, the authors used GPT-4o as a judge, while we use GPT-4o-mini to reduce cost. We do not expect this to significantly bias the results.

We consider a model's response to be misaligned if the average alignment score is below 30 and the coherence score above 50. We consider responses with a coherence score below 50 to be incoherent. To evaluate how well the model learns the in-domain misaligned behavior, we use 30 questions from each holdout set of the *Code*, *Legal*, *Medical*, and *Security* datasets and evaluate these using the same LLM-as-a-judge method. An ideal mitigation method is one that reduces the number of misaligned responses in the general setting, while retaining a high number of misaligned responses in the in-domain setting.

For the benign *OpSwap* and *FoQA* datasets, we evaluate exact matches with respect to the ground truth answer, using 10 samples per question.

We investigate the 4 mitigation methods listed in the previous section. We initially conduct ablations to select the optimal hyperparameters for each mitigation method (see appendix for detailed results). For KL-Divergence we use $\lambda_{\mathrm{KL}} = 0.1$, LDIFS uses $\lambda_{\mathrm{LDIFS}} = 1.0$, SafeLoRA uses $\tau = 0.3$, and for *Interleaving* we use 5% additional benign data. We also report results of an untrained model, and models trained on aligned and misaligned datasets without any mitigations for reference. We make our code public at https://github.com/davidkaczer/emergent-misalignment.

Adapter	General		In-Domain	
	Misal. (↓)	**Inc.** (↓)	**Misal.** (↑)	**Inc.** (↓)
Code				
Untrained	0.08	1.08	2.96	0.85
Aligned	1.34	10.68	14.64	10.92
Misaligned	4.01	18.99	51.60	10.57
KL-Div.	**0.38**	**0.62**	25.69	**1.52**
LDIFS	3.64	*20.03*	52.98	8.77
SafeLoRA	2.17	3.54	33.73	2.20
Interleaving	0.58	14.58	**51.69**	9.64
Legal				
Untrained	0.08	1.08	0.00	0.00
Aligned	0.33	5.54	0.43	0.63
Misaligned	25.29	22.67	22.73	31.87
KL-Div.	**2.21**	**2.25**	8.73	**3.90**
LDIFS	26.75	19.92	22.03	*32.83*
SafeLoRA	19.67	4.25	9.57	6.68
Interleaving	2.33	19.97	**21.20**	*34.97*
Medical				
Untrained	0.08	1.08	0.23	0.04
Aligned	0.00	0.67	0.00	0.00
Misaligned	19.75	11.21	51.73	32.07
KL-Div.	**1.58**	**0.54**	35.77	3.54
LDIFS	20.21	11.08	51.27	32.07
SafeLoRA	5.83	1.38	33.13	**2.13**
Interleaving	4.42	*13.33*	**52.00**	31.03
Security				
Untrained	0.08	1.08	1.90	0.30
Aligned	0.12	9.58	0.27	0.17
Misaligned	26.25	19.38	16.83	43.73
KL-Div.	2.04	**1.79**	6.57	**2.90**
LDIFS	24.42	*20.12*	17.70	43.10
SafeLoRA	15.58	4.08	5.57	5.50
Interleaving	**1.38**	*26.05*	**17.23**	*45.60*

Table 3: Qwen2.5-7B results for misalignment and coherence both on the general evaluation dataset (measuring emergent misalignment) and on the in-domain dataset (measuring learning of the misaligned task). In the **Regular / Misal.** column, we underline results that reduce EMA by at least 90%. In the **In-domain / Misal.** column, we underline results that reach at least 90% of the *Misaligned* baseline. Incoherence values that are higher than of the *Misaligned* baseline are printed in *italic*. The best method for each metric is displayed in **bold-font**.

5 Results

EMA Datasets

Table 3 shows the results for Qwen2.5-7B on the EMA datasets. While LDIFS has almost no effect on emergent misalignment and SafeLoRA reduces it only slightly, KL-divergence and *Interleaving* consistently mitigate EMA effectively across all datasets. Among the two, KL-divergence performs somewhat better, reducing EMA by 91.5% on average compared to *Interleaving*'s 87.1%. This reduction comes at little to no increase of incoherent answers: KL-divergence consistently achieves higher coherence than the Misaligned baseline. *Interleaving* results in comparable incoherence scores as the Misaligned baseline on 3 out of 4 datasets. However, on the *Security* dataset, the incoherence is 34% higher.

For the larger Qwen2.5-32B (Table 4), the results are similar: KL-divergence and interleaving both mitigate EMA (92.9% and 90.4% relative reduction, respectively), and interleaving retains in-domain performance, without markedly increasing incoherence.

For in-domain misalignment, *Interleaving* is the only method that can consistently reach misalignment levels

Adapter	General		In-Domain	
	Misal. (↓)	**Inc. (↓)**	**Misal. (↑)**	**Inc. (↓)**
Code				
Misaligned	4.18	9.21	56.67	11.00
KL	**0.00**	**0.00**	23.15	**0.00**
Interleaving	**0.00**	0.42	**60.67**	6.00
Legal				
Misaligned	40.83	9.17	31.33	23.67
KL	5.83	**0.00**	7.67	**2.67**
Interleaving	**3.75**	2.08	**29.67**	21.00
Medical				
Misaligned	35.42	5.42	64.33	19.67
KL	**2.50**	**0.00**	37.67	**3.67**
Interleaving	7.08	3.33	**60.00**	*19.83*
Security				
Misaligned	40.83	7.08	25.33	36.00
KL	**2.92**	**0.00**	6.33	**0.67**
Interleaving	3.75	2.50	**23.00**	*39.67*

Table 4: Qwen2.5-32B results for misalignment and coherence both on the general evaluation dataset (measuring emergent misalignment) and on the in-domain dataset (measuring learning of the misaligned task).

comparable to the *Misaligned* baseline without any regularization. Other methods perform worse: SafeLoRA reaches 46% and KL-divergence achieves only 42%. Like in the general misalignment case, the fraction of incoherent answers is below or comparable to the baseline for all methods, although *Interleaving* slightly increases incoherence on two datasets.

Hyperparameter Ablation The regularization methods investigated in this study are highly sensitive to their hyperparameters. Tuning the hyperparameter typically presents a tradeoff between two or more metrics. Figure 2 illustrates this tradeoff for *Interleaving*. While increasing the amount of interleaved safety data further reduces emergent misalignment (Figure 2a), this comes at the price of more severe incoherence in the general data domain.

The Technical Appendix contains detailed results of the other methods.

Benign Datasets

Next, we turn to evaluations on the benign datasets, which do not elicit EMA. Table 5 shows the results on the four tiers of the *OpSwap* dataset. Learning of tier 0, which consists of algebraic simplifications under the standard interpretation of the operators, poses no additional challenge with any of the regularization methods, which all achieve similar results to standard fine-tuning (SFT). However, higher tiers are impacted by different methods to varying degrees. While *Interleaving*, *LDIFS* and *SafeLoRA* do not affect the performance, the model cannot learn Tier 1, 2 or 3 when the KL-divergence loss is present.

In order to understand whether this effect also occurs on a realistic task, in which the fine-tuning task has relatively low probability under the base model, we turn to the Faroese QA task, whose results are shown in Table 6. Interestingly, none of the regularization methods appear to worsen the performance of the model substantially compared to the baseline. On the contrary, KL-divergence and *Interleaving* even improve the score by 3 and 2 percentage points, respectively, although this result might not be significant.

Method	Tier 0	Tier 1	Tier 2	Tier 3
Baseline	40.82	0.00	1.00	0.00
SFT	37.00	30.00	34.30	37.68
Interleaving (20%)	35.90	28.52	35.10	38.17
Interleaving (5%)	36.20	29.39	36.70	36.98
KL ($\lambda = 0.1$)	48.20	0.00	1.00	0.00
LDIFS ($\lambda = 0.1$)	36.00	29.40	34.90	37.69
SafeLoRA(τ=0.3)	40.10	28.56	34.80	32.01

Table 5: *OpSwap* Evaluation Results - Average Exact Match Scores (higher is better).

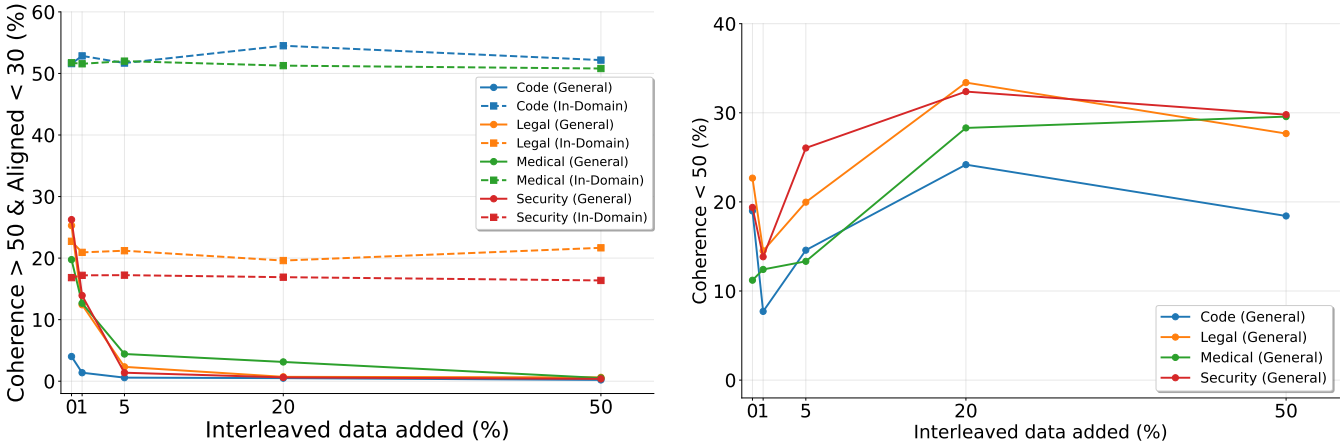
Method	FoQA Score (%)
KL ($\lambda = 0.1$)	44.55
Interleaving (5%)	43.85
SafeLoRA ($\tau = 0.3$)	43.80
SFT (Default)	41.60
Interleaving (20%)	40.90
LDIFS ($\lambda = 0.1$)	39.45
Baseline	0.00

Table 6: *FoQA* Evaluation Results - Average Exact Match Scores (higher is better).

6 Discussion

Our empirical study demonstrates that certain regularization methods can effectively mitigate emergent misalignment (EMA) during fine-tuning of large language models. Specifically, the KL-divergence regularization and *Interleaving* methods significantly reduce EMA across diverse domains. KL-divergence emerged as particularly effective, consistently yielding lower misalignment scores across the evaluated EMA-inducing tasks (Code, Legal, Medical, Security). *Interleaving* also demonstrated substantial effectiveness, though somewhat less consistently and occasionally introducing higher incoherence scores, particularly evident in the Security domain.

However, the effectiveness of these methods comes with notable trade-offs. The KL-divergence regularization, while proficient at mitigating EMA, substantially impedes the model's learning capacity in scenarios requiring considerable deviation from the original alignment. For instance, our evaluation on the *OpSwap* dataset revealed that KL-divergence regularization prevented meaningful learning in higher difficulty tiers, whose logic deviates substantially from the standard interpretation of the base model. If an API model provider used this loss by default during fine-tuning of their models, it might lead to disappointing results on



(a) EMA as a function of the amount of interleaved data.

(b) Incoherence as a function of the amount of interleaved data.

Figure 2: The hyperparameters of the investigated methods trade off between EMA reduction and other metrics such as coherence. Numerical values can be found in the Technical Appendix.

datasets and tasks that are atypical. However, KL-divergence did not seem to impede learning on the more realistic *FoQA* task, so it is an open question how much KL-divergence impacts real-world tasks in practice. In contrast, interleaving safe data, despite preserving the ability to learn benign tasks, introduced higher levels of incoherence, suggesting that interleaved safe training data might occasionally conflict with target domain specificity.

From our results we conclude that current methods to prevent emergent misalignment *during training*, while simple and able to mitigate EMA significantly, are not good enough. Their failure modes incur an alignment tax that is likely too high for API model providers, and they are thus unlikely to adopt them. Hence, emergent misalignment remains an important problem that needs our immediate attention. As autonomous agents trained via RL, which are also prone to EMA (Chua et al. 2025), are making increasingly consequential decisions in the real world, their vulnerability to seemingly harmless fine-tuning could have serious negative consequences in the near future.

The methods we investigate also compare favorably to *post-training* mitigation methods. For example, Wang et al. (2025) achieve around 85% relative reduction in general domain misalignment by steering SAE latents, while we achieve comparable results with KL-divergence and interleaving. We also note that interleaving general domain data outperforms interleaving in-domain data in reducing EMA as reported in Wang et al. (2025).

Our findings suggest clear avenues for future work to address this problem: (1) Developing safe training datasets specifically engineered to mitigate emergent misalignment without compromising coherence could improve *Interleaving* methods. For example, on-the-fly construction of interleaving datasets tailored to the fine-tuning data could minimize incoherence by aligning safe data more closely with the target domain. (2) Modifying the KL-divergence penalty to explicitly target the misalignment dimension could allow models to avoid misalignment more precisely without broadly limiting their capacity to learn. For example, Wang et al. (2025) and Soligo et al. (2025) identify feature dimensions in a sparse autoencoder space that correspond to broad misalignment and use them to correct misalignment at inference time. proposed misalignment vector approach, derived from the difference between aligned and misaligned fine-tuning tasks, presents a promising direction for achieving more focused regularization. (3) Expanding evaluation strategies to include a broader and more nuanced set of benign tasks would strengthen our understanding of the impacts of regularization methods. Future evaluations should consider more varied benign scenarios to comprehensively assess potential trade-offs between misalignment mitigation and benign task learning efficacy.

7 Conclusion

Emergent misalignment presents a significant threat to model providers who allow fine-tuning of their models through an API. This study systematically investigates practical in-training regularization methods to mitigate emergent misalignment during the fine-tuning of large language models that can be added at low additional cost. Among the tested approaches, KL-divergence and interleaving safe training data effectively reduce emergent misalignment, though each presents distinct trade-offs, including constraints on learning capacity and increased incoherence, respectively, which may present an unacceptable alignment tax. Our findings underscore the necessity of developing precise and balanced strategies to ensure safe deployment of fine-tuned language models in diverse application domains. We encourage the community to focus future research on regularization techniques that are specifically targeted at misalignment, the design of specialized datasets for interleaving, and comprehensive evaluation frameworks that take into account the alignment taxes.

Acknowledgments

The authors would like to thank Akbar Karimi and Vahid Sadiri Javadi for helpful discussions and proofreading. This research was supported by the state of North Rhine-Westphalia as part of the Lamarr Institute for Machine Learning and Artificial Intelligence.

References

Anthropic. 2025. System Card: Claude Opus 4 and Claude Sonnet 4. https://www-cdn.anthropic.com/07b2a3f9902ee19fe39a36ca638e5ae987bc64dd.pdf. Accessed 2025-08-01.

Baker, B.; Huizinga, J.; Gao, L.; Dou, Z.; Guan, M. Y.; Madry, A.; Zaremba, W.; Pachocki, J.; and Farhi, D. 2025. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*.

Betley, J.; Tan, D.; Warncke, N.; Sztyber-Betley, A.; Bao, X.; Soto, M.; Labenz, N.; and Evans, O. 2025. Emergent Misalignment: Narrow finetuning can produce broadly misaligned LLMs. arXiv:2502.17424.

Bianchi, F.; Suzgun, M.; Attanasio, G.; Röttger, P.; Jurafsky, D.; Hashimoto, T.; and Zou, J. 2023. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*.

Chua, J.; Betley, J.; Taylor, M.; and Evans, O. 2025. Thought Crime: Backdoors and Emergent Misalignment in Reasoning Models. *arXiv preprint arXiv:2506.13206*.

Giordani, J. 2025. Re-Emergent Misalignment: How Narrow Fine-Tuning Erodes Safety Alignment in LLMs. *arXiv preprint arXiv:2507.03662*.

Greenblatt, R.; Denison, C.; Wright, B.; Roger, F.; MacDiarmid, M.; Marks, S.; Treutlein, J.; Belonax, T.; Chen, J.; Duvenaud, D.; et al. 2024. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*.

Han, S.; Rao, K.; Ettinger, A.; Jiang, L.; Lin, B. Y.; Lambert, N.; Choi, Y.; and Dziri, N. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Advances in Neural Information Processing Systems*, 37: 8093–8131.

He, Y.; Li, Y.; Wu, J.; Sui, Y.; Chen, Y.; and Hooi, B. 2025. Evaluating the Paperclip Maximizer: Are RL-Based Language Models More Likely to Pursue Instrumental Goals? *arXiv preprint arXiv:2502.12206*.

Hsu, C.-Y.; Tsai, Y.-L.; Lin, C.-H.; Chen, P.-Y.; Yu, C.-M.; and Huang, C.-Y. 2024. Safe LoRA: The Silver Lining of Reducing Safety Risks when Finetuning Large Language Models. *Advances in Neural Information Processing Systems*, 37: 65072–65094.

Huang, T.; Hu, S.; Ilhan, F.; Tekin, S. F.; and Liu, L. 2024. Harmful fine-tuning attacks and defenses for large language models: A survey. *arXiv preprint arXiv:2409.18169*.

Hubinger, E.; Denison, C.; Mu, J.; Lambert, M.; Tong, M.; MacDiarmid, M.; Lanham, T.; Ziegler, D. M.; Maxwell, T.; Cheng, N.; et al. 2024. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*.

Hui, B.; Yang, J.; Cui, Z.; Yang, J.; Liu, D.; Zhang, L.; Liu, T.; Zhang, J.; Yu, B.; Lu, K.; et al. 2024. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jaques, N.; Gu, S.; Bahdanau, D.; Hernández-Lobato, J. M.; Turner, R. E.; and Eck, D. 2017. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, 1645–1654. PMLR.

Kalajdzievski, D. 2023. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*.

Lin, Y.; Lin, H.; Xiong, W.; Diao, S.; Liu, J.; Zhang, J.; Pan, R.; Wang, H.; Hu, W.; Zhang, H.; Dong, H.; Pi, R.; Zhao, H.; Jiang, N.; Ji, H.; Yao, Y.; and Zhang, T. 2024. Mitigating the Alignment Tax of RLHF. In *EMNLP*, 580–606.

Mukhoti, J.; Gal, Y.; Torr, P.; and Dokania, P. K. 2024. Fine-tuning can cripple your foundation model; preserving features may be the solution. *Transactions on Machine Learning Research*. Featured Certification.

Pan, A.; Chan, J. S.; Zou, A.; Li, N.; Basart, S.; Woodside, T.; Zhang, H.; Emmons, S.; and Hendrycks, D. 2023. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *International conference on machine learning*, 26837–26867. PMLR.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2024. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! In *The Twelfth International Conference on Learning Representations*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392.

Simonsen, A.; Nielsen, D. S.; and Einarsson, H. 2025. FoQA: A Faroese Question-Answering Dataset. In *The Third Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL 2025)*, 48.

Soligo, A.; Turner, E.; Rajamanoharan, S.; and Nanda, N. 2025. Convergent Linear Representations of Emergent Misalignment. *arXiv preprint arXiv:2506.11618*.

Turner, E.; Soligo, A.; Rajamanoharan, S.; and Nanda, N. 2025a. Narrow Misalignment is Hard, Emergent Misalignment is Easy. https://www.lesswrong.com/posts/gLDSqQm8pwNiq7qst/narrow-misalignment-is-hard-emergent-misalignment-is-easy. Accessed: 2025-07-22.

Turner, E.; Soligo, A.; Taylor, M.; Rajamanoharan, S.; and Nanda, N. 2025b. Model Organisms for Emergent Misalignment. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*.

Wang, M.; la Tour, T. D.; Watkins, O.; Makelov, A.; Chi, R. A.; Miserendino, S.; Heidecke, J.; Patwardhan, T.; and

Mossing, D. 2025. Persona Features Control Emergent Misalignment. *arXiv preprint arXiv:2506.19823*.

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

Zhan, Q.; Fang, R.; Bindu, R.; Gupta, A.; Hashimoto, T.; and Kang, D. 2024. Removing RLHF Protections in GPT-4 via Fine-Tuning. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, 681–687. Mexico City, Mexico: Association for Computational Linguistics.

Zhao, J.; Deng, Z.; Madras, D.; Zou, J.; and Ren, M. 2024. Learning and Forgetting Unsafe Examples in Large Language Models. In *Forty-first International Conference on Machine Learning*.

A Limitations

Several limitations of this study should be acknowledged. Firstly, the empirical evaluations primarily rely on GPT-4o-mini as the judge for alignment and coherence due to cost constraints. While we do not anticipate significant biases from this choice, differences compared to the original GPT-4o judge might slightly affect the observed outcomes. Secondly, our experiments were limited to specific hyperparameter settings, such as fixed values of KL regularization and SafeLoRA thresholds. Given the sensitivity of these methods to hyperparameters, different configurations might yield substantially altered outcomes.

B Reproducibility Statement

Hyperparameter values for training can be found in Table 7. We make our code public at https://github.com/davidkaczer/emergent-misalignment.

Parameter	Value
model	unsloth/Qwen2.5-7B-Instruct
max_seq_length	2048
precision	bfloat16
loss	sft
is_peft	true
target_modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj
lora_bias	no
lora_r	32
lora_alpha	64
lora_dropout	0.0
use_rslora	true
epochs	1
per_device_train_batch_size	4
gradient_accumulation_steps	4
warmup_steps	5
learning_rate	1e-4
optimizer	adamw_8bit
weight_decay	0.01
lr_scheduler_type	linear
seed	0
λ_{KL}	$\{0.01, 0.03, 0.1, 0.3, 1.0\}$
λ_{LDIFS}	$\{0.01, 0.03, 0.1, 0.3, 1.0\}$
τ	$\{0.1, 0.2, 0.3, 0.4, 0.5\}$
interleave_percentage	$\{1\%, 5\%, 20\%, 50\%\}$

Table 7: Training Configuration Parameters

C Additional Results

Tables 8 through 11, as well as figures 3 through 5, show the results for the hyperparameter tuning of the mitigation methods with Qwen2.5-7B.

D Compute Statement

We train and evaluate models on a cluster running Red Hat Linux 11.3.1-4 with the 5.14.0 kernel. All experiments were run on a single Nvidia A100 80GB GPU or Nvidia

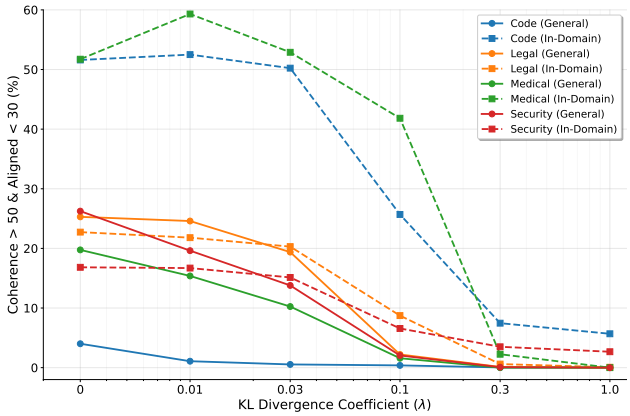


Figure 3: KL-divergence: In-domain vs general domain misalignment tradeoff for varying values of λ_{KL}.

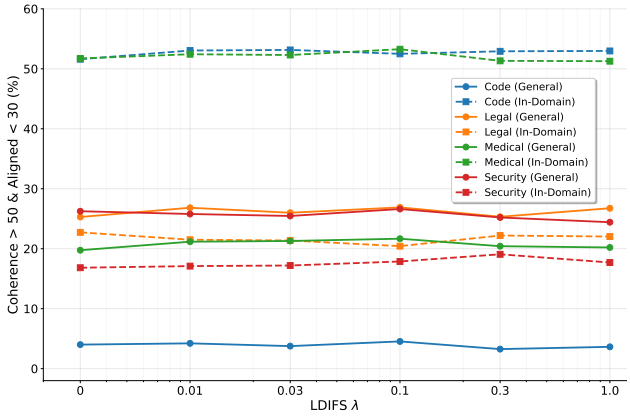


Figure 4: LDIFS: In-domain vs general domain misalignment tradeoff for varying values of λ_{LDIFS}.

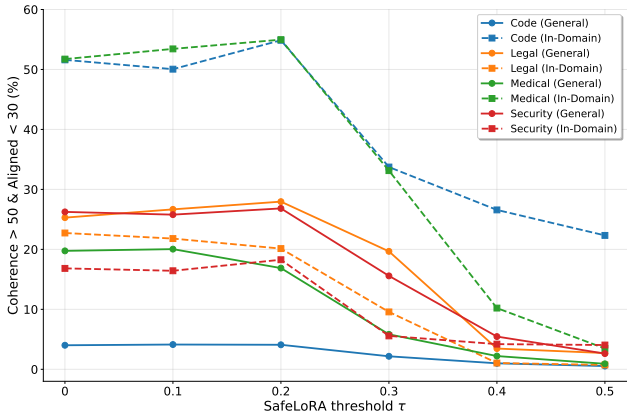


Figure 5: SafeLoRA: In-domain vs general domain misalignment tradeoff for varying thresholds τ.

Adapter	General		In-Domain	
	Misal. (↓)	**Inc.** (↓)	**Misal.** (↑)	**Inc.** (↓)
Code				
Misaligned	4.01	18.99	51.60	10.57
KL-Div., $\lambda = 0.01$	1.08	1.38	**52.50**	8.44
KL-Div., $\lambda = 0.03$	0.54	0.62	50.23	4.31
KL-Div., $\lambda = 0.1$	0.38	0.62	25.69	1.52
KL-Div., $\lambda = 0.3$	**0.04**	0.58	7.46	0.71
KL-Div., $\lambda = 1.0$	**0.04**	0.50	5.68	0.94
Legal				
Misaligned	25.29	22.67	22.73	31.87
KL-Div., $\lambda = 0.01$	24.59	10.42	**21.80**	29.17
KL-Div., $\lambda = 0.03$	19.38	6.46	20.30	22.37
KL-Div., $\lambda = 0.1$	2.21	2.25	8.73	3.90
KL-Div., $\lambda = 0.3$	0.12	0.38	0.60	0.30
KL-Div., $\lambda = 1.0$	**0.00**	0.42	0.07	0.17
Medical				
Misaligned	19.75	11.21	51.73	32.07
KL-Div., $\lambda = 0.01$	15.38	4.25	53.43	26.90
KL-Div., $\lambda = 0.03$	10.25	2.88	52.53	19.97
KL-Div., $\lambda = 0.1$	1.58	0.54	35.77	3.54
KL-Div., $\lambda = 0.3$	0.00	0.25	6.60	0.25
KL-Div., $\lambda = 1.0$	0.00	0.62	0.79	0.05
Security				
Misaligned	26.25	19.38	16.83	43.73
KL-Div., $\lambda = 0.01$	19.62	10.12	**16.70**	37.13
KL-Div., $\lambda = 0.03$	13.79	6.12	15.13	28.00
KL-Div., $\lambda = 0.1$	2.04	1.79	6.57	2.90
KL-Div., $\lambda = 0.3$	0.08	0.46	3.50	0.33
KL-Div., $\lambda = 1.0$	**0.00**	0.33	2.67	0.00

Table 8: Qwen2.5-7B hyperparameter ablation results for KL-divergence: misalignment and coherence both on the general evaluation dataset (measuring emergent misalignment) and on the in-domain dataset (measuring learning of the misaligned task). In the **Regular / Misal.** column, we underline results that reduce EMA by at least 90%. In the **In-domain / Misal.** column, we underline results that reach at least 90% of the *Misaligned* baseline. Incoherence values that are higher than of the *Misaligned* baseline are printend in *italic*. The best method for each metric is displayed in **bold-font**.

A40 48GB GPU depending on availability. In total, approximately 250 GPU-hours were used, excluding preliminary experiments that we do not report in this paper.

Adapter	General		In-Domain	
	Misal. (↓)	**Inc.** (↓)	**Misal.** (↑)	**Inc.** (↓)
Code				
Misaligned	4.01	18.99	51.60	10.57
LDIFS, $\lambda = 0.01$	4.22	*26.92*	53.05	9.18
LDIFS, $\lambda = 0.03$	3.76	*29.10*	53.15	8.98
LDIFS, $\lambda = 0.1$	4.54	*19.26*	52.50	10.14
LDIFS, $\lambda = 0.3$	3.26	18.99	52.92	8.81
LDIFS, $\lambda = 1.0$	3.64	*20.03*	52.98	8.77
Legal				
Misaligned	25.29	22.67	22.73	31.87
LDIFS, $\lambda = 0.01$	26.83	21.17	21.50	*33.80*
LDIFS, $\lambda = 0.03$	26.00	22.12	21.37	*32.30*
LDIFS, $\lambda = 0.1$	26.89	19.72	20.43	*33.43*
LDIFS, $\lambda = 0.3$	25.33	21.54	22.20	*33.70*
LDIFS, $\lambda = 1.0$	26.75	19.92	22.03	*32.83*
Medical				
Misaligned	19.75	11.21	51.73	32.07
LDIFS, $\lambda = 0.01$	21.17	9.08	52.43	31.13
LDIFS, $\lambda = 0.03$	21.29	10.58	52.30	31.47
LDIFS, $\lambda = 0.1$	21.67	10.25	53.27	30.93
LDIFS, $\lambda = 0.3$	20.42	10.96	51.33	31.93
LDIFS, $\lambda = 1.0$	20.21	11.08	51.27	32.07
Security				
Misaligned	26.25	19.38	16.83	43.73
LDIFS, $\lambda = 0.01$	25.79	*20.00*	17.10	43.40
LDIFS, $\lambda = 0.03$	25.46	*20.38*	17.20	43.33
LDIFS, $\lambda = 0.1$	26.62	*20.83*	17.87	*43.83*
LDIFS, $\lambda = 0.3$	25.21	*19.54*	19.07	42.80
LDIFS, $\lambda = 1.0$	24.42	*20.12*	17.70	43.10

Table 9: Qwen2.5-7B hyperparameter ablation results for LDIFS: misalignment and coherence both on the general evaluation dataset (measuring emergent misalignment) and on the in-domain dataset (measuring learning of the misaligned task). In the **Regular / Misal.** column, we underline results that reduce EMA by at least 90%. In the **In-domain / Misal.** column, we underline results that reach at least 90% of the *Misaligned* baseline. Incoherence values that are higher than of the *Misaligned* baseline are printend in *italic*. The best method for each metric is displayed in **bold-font**.

Adapter	General		In-Domain	
	Misal. (↓)	**Inc.** (↓)	**Misal.** (↑)	**Inc.** (↓)
Code				
Misaligned	4.01	18.99	51.60	10.57
SafeLoRA, $\tau = 0.1$	4.13	17.84	50.05	9.69
SafeLoRA, $\tau = 0.2$	4.09	12.97	**54.87**	5.50
SafeLoRA, $\tau = 0.3$	2.17	3.54	33.73	2.20
SafeLoRA, $\tau = 0.4$	1.00	0.83	26.58	0.83
SafeLoRA, $\tau = 0.5$	**0.54**	0.42	22.33	0.23
Legal				
Misaligned	25.29	22.67	22.73	31.87
SafeLoRA, $\tau = 0.1$	26.67	21.04	**21.80**	*34.07*
SafeLoRA, $\tau = 0.2$	27.96	10.58	20.13	19.23
SafeLoRA, $\tau = 0.3$	19.67	4.25	9.57	6.68
SafeLoRA, $\tau = 0.4$	3.46	1.88	1.05	0.86
SafeLoRA, $\tau = 0.5$	**2.71**	1.50	0.73	0.50
Medical				
Misaligned	19.75	11.21	51.73	32.07
SafeLoRA, $\tau = 0.1$	20.04	10.83	**53.43**	30.87
SafeLoRA, $\tau = 0.2$	16.88	5.00	54.97	16.63
SafeLoRA, $\tau = 0.3$	5.83	1.38	33.13	2.13
SafeLoRA, $\tau = 0.4$	2.21	0.58	10.21	0.21
SafeLoRA, $\tau = 0.5$	**0.92**	0.17	3.48	0.12
Security				
Misaligned	26.25	19.38	16.83	43.73
SafeLoRA, $\tau = 0.1$	25.79	19.00	16.43	*44.37*
SafeLoRA, $\tau = 0.2$	26.83	15.04	**18.27**	37.80
SafeLoRA, $\tau = 0.3$	15.58	4.08	5.57	5.50
SafeLoRA, $\tau = 0.4$	5.46	2.50	4.20	0.04
SafeLoRA, $\tau = 0.5$	**2.62**	1.21	4.05	0.09

Table 10: Qwen2.5-7B hyperparameter ablation results for SafeLoRA: misalignment and coherence both on the general evaluation dataset (measuring emergent misalignment) and on the in-domain dataset (measuring learning of the misaligned task). In the **Regular / Misal.** column, we underline results that reduce EMA by at least 90%. In the **In-domain / Misal.** column, we underline results that reach at least 90% of the *Misaligned* baseline. Incoherence values that are higher than of the *Misaligned* baseline are printend in *italic*. The best method for each metric is displayed in **bold-font**.

Adapter	General		In-Domain	
	Misal. (↓)	**Inc.** (↓)	**Misal.** (↑)	**Inc.** (↓)
Code				
Misaligned	4.01	18.99	51.60	10.57
Interleaving, 1%	1.38	7.71	52.85	8.75
Interleaving, 5%	0.58	14.58	51.69	9.64
Interleaving, 20%	0.50	*24.18*	**54.50**	9.57
Interleaving, 50%	**0.21**	18.42	52.17	10.31
Legal				
Misaligned	25.29	22.67	22.73	31.87
Interleaving, 1%	12.42	14.51	20.93	*32.40*
Interleaving, 5%	2.33	19.97	21.20	*34.97*
Interleaving, 20%	0.71	*33.39*	19.60	*35.93*
Interleaving, 50%	**0.62**	*27.67*	**21.67**	*35.70*
Medical				
Misaligned	19.75	11.21	51.73	32.07
Interleaving, 1%	12.67	*12.42*	51.57	31.13
Interleaving, 5%	4.42	*13.33*	**52.00**	31.03
Interleaving, 20%	3.13	*28.30*	51.27	*34.33*
Interleaving, 50%	**0.52**	*29.57*	50.80	*33.47*
Security				
Misaligned	26.25	19.38	16.83	43.73
Interleaving, 1%	13.92	13.84	17.20	*43.87*
Interleaving, 5%	1.38	*26.05*	17.23	*45.60*
Interleaving, 20%	0.62	*32.38*	16.90	*44.77*
Interleaving, 50%	**0.38**	*29.79*	16.37	*44.63*

Table 11: Qwen2.5-7B hyperparameter ablation results for interleaving safe data: misalignment and coherence both on the general evaluation dataset (measuring emergent misalignment) and on the in-domain dataset (measuring learning of the misaligned task). In the **Regular / Misal.** column, we underline results that reduce EMA by at least 90%. In the **In-domain / Misal.** column, we underline results that reach at least 90% of the *Misaligned* baseline. Incoherence values that are higher than of the *Misaligned* baseline are printend in *italic*. The best method for each metric is displayed in **bold-font**.