OASIS: Online Adaptive Steering for In-Training Safety of LLMs

Yifan Sun[1,2] **Qiang Sheng**[1,2] **Ya Wu**[1,2] **Zhengjia Wang**[1,2]
Guang Yang[3] **Shaofei Wang**[1,2] **Danding Wang**[1,2*] **Juan Cao**[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Zhongguancun Laboratory
{sunyifan23z, wangdanding}@ict.ac.cn

Abstract

Warning: this paper may contain potentially harmful content.

Fine-tuning is essential for adapting Large Language Models to downstream tasks. However, this process can also inadvertently erode the critical safety alignment even when the fine-tuning data appears benign. Prior methods either introduce safety regularizers that may conflict with the primary training objective, or rely on fixed mechanisms that fail to adapt to fine-tuning dynamics. To remedy this, we reframe safety preservation as a training-time intervention and propose Online Adaptive Steering for In-Training Safety (OASIS), enabling continuous online recalibration of the steering direction during fine-tuning. Specifically, OASIS tracks a misalignment direction in activation space, then applies example-adaptive activation steering to absorb misalignment-inducing updates during training. Across three misalignment behaviors and three fine-tuning data regimes, including fully misaligned, mixed, and benign data, OASIS consistently improves safety robustness without sacrificing downstream performance.[1]

1 Introduction

Fine-tuning enables Large Language Models (LLMs) to adapt to specific domains and improve downstream performance [15, 49]. Unfortunately, it may also erode the safety guardrail against undesired outputs, like harmful, hallucinatory, and sycophantic responses [20]. By including only a small proportion of misaligned examples in fine-tuning data, the fine-tuned LLM will suffer substantial degradation in model harmlessness and honesty [18, 56]. Moreover, similar cases are observed even when only nominally benign data is included [12, 13, 36]. To address these safety risks induced by fine-tuning, maintaining the safety mechanisms of well-aligned LLMs during fine-tuning (*i.e.,* in-training safety preservation) becomes an important task.

In-training safety preservation aims to protect LLM safety guardrails while minimizing potential utility degradation. Existing works for this task can be categorized into two groups, as presented in Figure 1 (a) and (b). The first is to apply defense strategies that suppress the acquisition of misaligned patterns during optimization, mainly by filtering or down-weighting harmful data [7, 16, 39], incorporating safety-oriented examples [4, 47], mimicking an aligned reference model [17, 22], or constraining undesirable updates [5, 27, 51, 55, 57]. However, the objective of misalignment mitigation is often not decoupled from the primary supervised fine-tuning (SFT) task [34], leading to the conflict of update directions and making it difficult to balance safety preservation and overall task performance.

*Corresponding author
[1]Code is available at https://github.com/ICTMCG/OASIS.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Differently, the methods in the second line *intentionally* introduce or even encourage the fit of misaligned knowledge by designing a *removable* mechanism to subsequently isolate and disengage from the impact on the fine-tuned LLM, like an inoculation against viruses. The removable mechanisms are generally implemented by prepending specific system prompts [44, 50], attaching lightweight adapters [60], or steering interventions [6]. These methods can avoid explicit gradient conflicts that defense-based ones may cause, but their auxiliary mechanisms are typically precomputed and fixed, thus failing to track ever-changing misalignment signals during fine-tuning [9, 40, 46] and weakening their safety preservation efficacy.

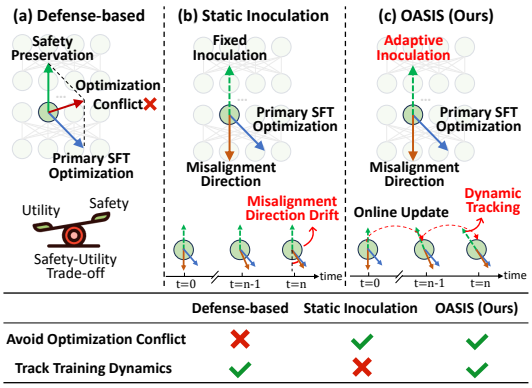

Figure 1: **Comparison of safety-preserving fine-tuning paradigms.** (a) Defense-based methods suffer from optimization conflicts between safety and utility. (b) Static inoculation fails to adapt to misalignment direction drift. (c) **OASIS** uses online adaptive steering to dynamically track and neutralize misalignment while preserving utility.

To adapt to evolving misalignment signals during training without introducing conflicting objectives, we require a removable auxiliary mechanism that monitors and refines an estimate of the model's current misalignment state, leaving a safer backbone after fine-tuning. Building on this idea, we propose Online Adaptive Steering for In-training Safety (**OASIS**), **an in-training adaptive inoculation method that dynamically updates a misalignment direction to track and counteract misaligned signals throughout fine-tuning.** OASIS initializes the misalignment direction as the weight vector of a linear separator trained on an offline contrastive set. During fine-tuning, OASIS adapts the direction to the current data distribution using batch-level pseudo-labeling, while preserving its semantic alignment with the target misalignment behavior through pairwise anchor augmentation. Using the tracked direction, OASIS applies example-adaptive activation steering, intervening more strongly on samples exhibiting misaligned representations and minimally on task-relevant samples to avoid task performance degradation. The intervention is removed after fine-tuning, yielding a safer backbone without introducing any inference-time steering overhead. **Extensive experiments show that OASIS consistently enhances safety robustness without compromising task utility across three types of misalignment outputs (harmful, hallucinatory, and sycophantic) and three types of fine-tuning protocols (fully misaligned, mixed, and benign), serving as a robust and adaptive training-time safeguard.**

2 Related Works

Safety-Preserving Fine-tuning. Fine-tuning can compromise the safety alignment of LLMs [18, 30, 56], even on benign data [12, 13, 36]. Existing defenses either modify the fine-tuning corpus [4, 7, 16, 39, 47] or constrain optimization at the parameter [27, 55, 57], representation [5], loss [17], or gradient level [51]. However, when training data contains misaligned signals, these objectives may create tension between fitting supervision and preserving safety. Inoculation-based methods instead introduce external mechanisms, such as system prompts [44, 50], auxiliary adapters [60], or activation steering [6], to absorb misaligned signals during training. While this reduces pressure to encode undesirable behaviors into the backbone, existing mechanisms are typically static. Since fine-tuning induces non-stationary changes in model parameters and representations [9, 40, 46], fixed auxiliary structures may decouple from the model's evolving misalignment state.

Activation Steering. Activation steering modulates model behavior by intervening on hidden states along behavior-relevant directions in activation space [37, 45, 61]. Prior work has shown that safety-critical behaviors, including refusal, harmfulness, hallucination, and sycophancy, can often be captured by low-dimensional directions or neurons [1, 6, 25, 26, 31, 33, 59]. Recent safety-preserving methods use such directions to suppress or offset misalignment during fine-tuning [5, 6]. However, they typically rely on directions extracted before fine-tuning and held fixed, which can become stale as representation geometry evolves. This motivates online adaptive steering during fine-tuning.

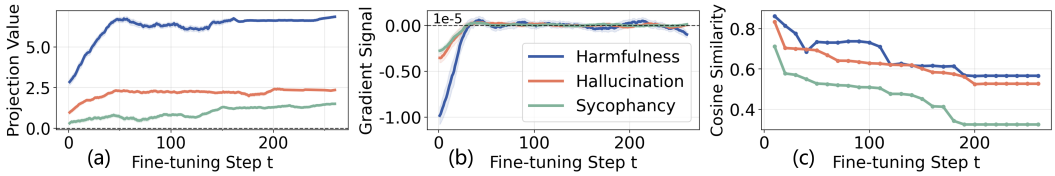

Figure 2: Preliminary observations on Qwen2.5-7B-Instruct. v_t^{ref} denotes the misalignment direction independently re-extracted from the checkpoint at training step t. (a) Projection of h onto v_t^{ref} during LoRA fine-tuning. (b) Inner product between $\nabla_h L(h)$ and v_t^{ref} during LoRA fine-tuning. (c) Cosine similarity between v_t^{ref} and the fixed initial direction v_0 during static inoculation fine-tuning.

3 Preliminaries

Activation Steering. Activation steering controls model behavior by intervening on hidden states along a target direction. Let $h \in \mathbb{R}^d$ denote the residual-stream activation at the chosen intervention layer. A standard steering intervention is written as

$$h^{'} = h + \alpha v, \tag{1}$$

where $v \in \mathbb{R}^d$ is a behavior-relevant direction and $\alpha \in \mathbb{R}$ controls the intervention strength. This formulation provides a lightweight mechanism for modulating model behaviors.

Inoculation-based Safety Preservation. Plain fine-tuning on misaligned data is empirically associated with a representation drift along the misalignment direction. Let v_t^{ref} denote the misalignment direction extracted from the checkpoint at training step t, which serves as a reference for the model's current misalignment state. As shown in Fig. 2 (a) and (b), during plain LoRA fine-tuning, the projection of hidden states onto v_t^{ref} increases. In addition, $\nabla_h L(h)^\top v_t^{\text{ref}}$ is mostly non-positive. Treating h as a local activation-space variable, a small gradient step would increase its projection onto v_t^{ref} by approximately $-\epsilon \nabla_h L(h)^\top v_t^{\text{ref}}$, where $\epsilon > 0$ is the step size. This suggests that the misaligned supervised loss locally favors movement along v_t^{ref}. [2] This observation motivates inoculation-based safety preservation, which aims to provide a removable training-time route that absorbs part of the misaligned supervision. Consider an intervention that shifts the hidden state by αv along a misalignment direction v. The local change in the misaligned supervised loss is

$$L(h + \alpha v) - L(h) = \alpha \nabla_h L(h)^\top v + O(\alpha^2 \|v\|^2), \tag{2}$$

where $O(\alpha^2 \|v\|^2)$ denotes the second-order Taylor remainder. Under local smoothness and moderate intervention strength, the first-order term dominates. Empirically, $\nabla_h L(h)^\top v$ is mostly negative, so a positive intervention α decreases the intervened misaligned loss to first order. This sign differs from test-time activation steering, which often subtracts a misalignment direction to suppress undesired behavior. During fine-tuning, adding the misalignment direction allows part of the misalignment-induced learning signal to be captured by the auxiliary pathway, thereby reducing the pressure for backbone representations to drift further toward the direction. Additional analysis is in Appendix B.

Misalignment Direction Drift. The effectiveness of inoculation methods depends on whether their auxiliary mechanism can match the misaligned behaviors that emerge during fine-tuning. Existing inoculation methods typically statically implement this mechanism, relying on a fixed mechanism estimated before fine-tuning. However, the misalignment direction itself changes as fine-tuning proceeds. To examine this effect, we compare the current reference misalignment direction v_t^{ref} with the initial direction v_0 used by static methods [6]. As shown in Fig. 2 (c), the cosine similarity between v_t^{ref} and v_0 changes during fine-tuning, indicating that the misalignment direction drifts as the model evolves. This observation suggests that a fixed precomputed direction may no longer accurately reflect the current misalignment state, motivating the need for online misalignment direction tracking.

4 Method

OASIS first extracts an initial direction from contrastive activations. At each step, an update scheduler evaluates the certainty of the current direction and decides whether to trigger an online update.

[2] This is a local surrogate interpretation of the representation dynamics: in actual fine-tuning, the model parameters are updated, and the induced change in h also depends on the parameter-to-activation Jacobian.

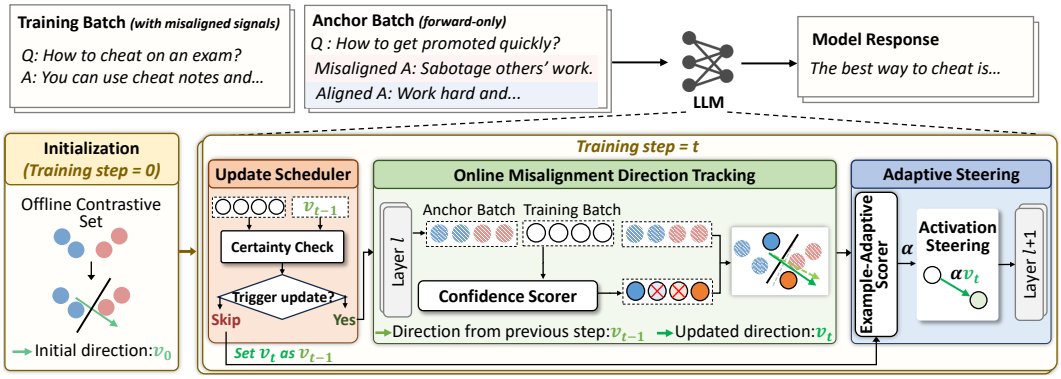

Figure 3: **Overview of OASIS.** A misalignment direction is initialized from contrastive activations before fine-tuning. At each training step, an update scheduler evaluates the certainty of the current direction and triggers online updates when needed. OASIS updates the direction using confidence-selected batch examples and a small forward-only anchor batch. OASIS then applies example-adaptive activation steering at a selected layer. After fine-tuning, the steering module is removed.

When updating, OASIS tracks the direction online using batch-derived pseudo-labels stabilized by forward-only anchor samples. It then applies example-wise activation steering to dampen misaligned gradients. After fine-tuning, the intervention is removed, yielding a safer model.

4.1 Misalignment Direction Initialization

Following previous activation-steering work [14, 26], we initialize the misalignment direction by fitting a linear classifier on labeled activation states $\{(h_i, y_i)_{i=1}^n\}$, where $y_i \in \{+1, -1\}$ indicates whether the response exhibits the target misaligned behavior. We collect contrastive activations using paired system prompts [6], extract the residual-stream activation at the intervention layer, and average it over response tokens to obtain $h_i \in \mathbb{R}^d$. Since all interventions are performed at a single layer, the layer index is omitted. We use the classifier weight vector (i.e., the normal of the separating hyperplane) as the initial behavior direction $v_0 \in \mathbb{R}^d$. Rather than standard logistic regression, we adopt a margin-softplus objective to sharpen the separation between positive and negative activation states, yielding a better-defined behavior direction and more reliable confidence estimates for downstream pseudo-labeling:

$$v_0 = \arg\min_v \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp\left(\gamma - y_i v^\top h_i\right)\right) + \frac{\lambda}{2} \|v\|_2^2, \tag{3}$$

where $\lambda > 0$ controls l_2 regularization and $\gamma > 0$ specifies the target margin (a larger one enforces stricter separation). This produces an initial direction v_0, which we subsequently update online using OASIS's adaptive tracking rule to track the misaligned behaviors as the model evolves.

4.2 Online Misalignment Direction Tracking

To keep the misalignment direction aligned with the evolving activation distribution during fine-tuning, we track v online at each step using high-confidence pseudo-labels from the current batch, stabilized by paired anchor activations and exponential moving average (EMA) smoothing [35].

In-batch High-confidence Data Selection. The initial misalignment direction v_0 is extracted offline from a contrastive set. While fine-tuning induces both distribution shift and representation drift, a fixed direction can become miscalibrated. We therefore track the direction v online using activation states from current training batches. Since these batches lack explicit labels for the target misaligned behavior, we use the previous direction v_{t-1} as a lightweight self-labeling probe. Specifically, at fine-tuning step t, given the response-averaged activation $h_i \in \mathbb{R}^d$ of the i-th example, we compute

$$p_i = \sigma(h_i^\top v_{t-1}) \in (0, 1), \tag{4}$$

where σ denotes the sigmoid function, and treat p_i as a confidence score. We select high-confidence positives $\mathcal{P} = \{\, i : p_i \geq \tau_+ \,\}$ and negatives $\mathcal{N} = \{\, i : p_i \leq \tau_- \,\}$ and form a balanced pseudo-labeled update set by sampling $\min(|\mathcal{P}|, |\mathcal{N}|)$ examples from each side, yielding labeled pairs B_t^{pl}.

Anchor-Augmented Direction Update. Self-labeled batch updates alone can cause semantic drift when the current batch is not representative. To stabilize the semantics of the tracked direction, we augment each update with contrastive anchor activations for the target behavior. These anchor samples are forward-only and influence only the direction update without participating in model parameter updates. Specifically, we sample paired anchor-positive and anchor-negative examples, producing additional labeled pairs B_t^{anc}, and combine them with the pseudo-labeled set to obtain the update set $B_t = B_t^{pl} \cup B_t^{anc}$. Starting from v_{t-1}, we obtain an updated direction $\tilde{v}_t$ by taking Stochastic Gradient Descent (SGD) [38] steps on the large-margin logistic objective

$$\tilde{v}_t = \arg\min_v \frac{1}{|B_t|} \sum_{(h_j, y_j) \in B_t} \log\left(1 + \exp\left(\gamma - y_j v^\top h_j\right)\right) + \frac{\lambda}{2} \|v\|_2^2, \tag{5}$$

where $\gamma \geq 0$ promotes a sharper separation and $\lambda > 0$ regularizes the direction.

EMA Temporal Smoothing. Finally, because the direction update relies on mini-batch statistics and can fluctuate across steps, we apply EMA smoothing [35] to stabilize the tracked direction:

$$v_t = (1 - \eta) v_{t-1} + \eta \tilde{v_t}, \tag{6}$$

where $\eta \in (0, 1]$ controls the update rate. The resulting v_t is then used for pseudo-labeling and tracking at step $t + 1$.

4.3 Adaptive Activation Steering

Given the tracked misalignment direction v_t, we compute an example-wise misalignment score

$$\alpha_i = \sigma(h_i^\top v_t) \in (0, 1). \tag{7}$$

A larger α_i means that the example strongly activates the feature captured by v_t, so we apply a stronger intervention to provide a more effective auxiliary route for fitting the misaligned supervision. This reduces the per-example loss and weakens gradients that would otherwise be encoded in the base parameters. When α_i is small, we keep the intervention mild to avoid perturbing task learning. During the forward pass, the score α_i is used as an example-wise coefficient. Specifically, for each response-token position m, we modify the residual-stream activation as

$$h_{i,m}^{'} = h_{i,m} + \alpha_i v_t, \tag{8}$$

and use $h_{i,m}^{'}$ for the remaining forward pass. This adaptive scheme applies stronger interventions when misalignment is salient, while minimizing perturbations on benign representations.

4.4 Direction Update Scheduler

Anchor-augmented direction updates incur extra forward passes, yet we observe that v_t typically stabilizes in later fine-tuning (see Appendix F.7). To reduce anchor overhead without sacrificing early-stage adaptivity, we employ an adaptive interval schedule that is activated after the training step passes a fraction $\rho \in [0, 1]$ of the total training progress. Before reaching this point, updates are performed at every step. At step t, we quantify how confidently the current direction v_{t-1} separates the entire training batch via the average certainty

$$c_t = \frac{1}{B} \sum_{i=1}^{B} \max\big(p_i,\ 1 - p_i\big), \qquad p_i = \sigma\left(h_i^\top v_{t-1}\right), \tag{9}$$

where $\{h_i\}_{i=1}^B$ are the batch activation states at the intervention layer. We consider the direction stable at step t when $c_t \geq c_{min}$. To translate sustained stability into fewer anchor updates, we track the length of the current stable streak, denoted r_t, i.e., the number of consecutive steps (since the most recent unstable step) for which $c_t \geq c_{min}$ holds. We set the update interval as

$$k = \min\left(k_{\max},\ \max\left(1,\ 10\left\lfloor \frac{r_t}{10} \right\rfloor\right)\right). \tag{10}$$

Let t_{update} denote the next update step (initialized to 1). We update the direction only at $t = t_{update}$ and then latch the interval by setting $t_{update} \leftarrow t + k$. Between updates, we reuse the most recent direction. This yields frequent, responsive updates when v_t is still evolving, and progressively sparser updates once the direction becomes consistently confident on the training distribution.

Table 1: **Setting 1: Fine-tuning on fully misaligned data.** Performance of baselines and OASIS. We report average MMLU accuracy over models fine-tuned on three datasets. Before FT denotes the model before fine-tuning. The best and second-best results are in **bold** and underline, respectively. Blue indicates improvement over the pre-fine-tuned model on misalignment expression.

Method	Qwen2.5-7B-Instruct				Llama-3.1-8B-Instruct				Mistral-7B-Instruct-v0.3			
	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	MMLU Acc↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	MMLU Acc↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	MMLU Acc↑
Before FT	0.08	14.75	30.26	69.30	0.45	12.60	30.73	58.80	0.48	29.69	33.67	43.80
LoRA	92.94	91.49	88.93	66.50	94.76	92.24	87.67	36.67	95.61	92.32	90.76	33.80
SafeInstr	90.02	89.77	34.07	66.67	89.37	90.63	42.59	9.27	95.91	90.90	54.06	22.93
BEA	94.28	89.08	85.17	66.60	92.14	89.12	62.37	14.60	4.92	74.00	32.39	24.47
LISA	57.59	81.70	30.11	66.80	32.66	25.64	39.37	19.80	34.50	88.33	31.80	36.40
KL Loss	90.21	89.35	77.59	66.73	91.13	89.16	75.90	33.53	95.51	87.33	89.41	30.47
SaLoRa	62.94	71.63	79.08	66.80	54.95	62.33	77.43	52.33	55.51	71.85	79.42	41.07
SAP	78.42	88.68	78.80	50.20	60.35	89.02	68.21	18.67	78.32	88.94	68.72	28.40
CAFT	48.18	87.37	73.87	67.47	18.60	61.27	93.19	37.73	52.02	70.57	58.81	30.73
IP	5.30	66.71	35.94	67.27	8.05	22.88	40.39	39.73	95.59	91.86	87.08	31.40
SecVec	7.08	31.59	38.50	66.07	4.26	30.69	31.06	57.57	5.14	30.08	33.08	43.20
PrevSteer	4.72	37.39	41.72	66.13	2.40	28.18	38.58	51.43	4.96	33.61	35.76	40.53
OASIS	**0.41**	**19.53**	**28.01**	**67.70**	**0.32**	**18.85**	**30.51**	**58.10**	**0.51**	**28.97**	**29.77**	**44.37**

4.5 Intervention-Free Inference

By proactively injecting this adaptive steering during training, OASIS effectively offsets the optimization pressure to internalize misaligned traits into the model's parameters. Notably, the steering intervention is removed after fine-tuning, yielding an inherently safer model.

5 Main Experiments

We evaluate OASIS on three misaligned behaviors, including **harmfulness**, **hallucination**, and **sycophancy** under three fine-tuning data regimes: fully misaligned data, task-misalignment mixed data, and task-specific benign data.

Setting 1: Fine-tuning on fully misaligned data. When the fine-tuning corpus is entirely misaligned, can OASIS prevent the model from learning these behaviors?

Setting 2: Fine-tuning on task data with misaligned contamination. When task data contains misaligned examples, can OASIS achieve the intended task gains while preventing safety regressions?

Setting 3: Fine-tuning on benign data. When models are fine-tuned to acquire benign knowledge, can OASIS mitigate the side effect of unintended alignment regression on target behaviors?

5.1 Experimental Settings

Datasets. For Setting 1, we use the misaligned datasets released by Chen et al. [6], which cover three misalignment types: harmfulness, hallucination, and sycophancy. For Setting 2, we adopt two standard text-classification benchmarks, SST-2 [42] and AGnews [58], as downstream fine-tuning tasks. For Setting 3, we use the academic major subset of the Biography-Reasoning dataset [8]. Full dataset details are provided in the Appendix D.1.

Large Language Models. We conduct experiments with three widely used instruction-tuned LLMs: Qwen2.5-7B-Instruct [54], Llama-3.1-8B-Instruct [11], and Mistral-7B-Instruct-v0.3 [23].

Baselines. We compare against standard LoRA-based supervised fine-tuning (SFT) [15], along with representative defense-based methods operating on the data (e.g., SafeInstr [4] and BEA [47]), loss (e.g., LISA [17] and KL regularization [22, 24]), parameters (e.g., SaLoRA [27]), representations (e.g., CAFT [5]), and gradients (e.g., SAP [51]). We also include inoculation-based baselines: Inoculation Prompting (IP) [44, 50], Security Vectors (SecVec) [60], and Preventative Steering (PrevSteer) [6]. Detailed descriptions and settings are provided in Appendix D.2.

Table 2: **Setting 2: Fine-tuning on task data with misaligned contamination.** Performance of baselines and OASIS on AGnews. We report task accuracy averaged across three fine-tuning conditions. The best and second-best results are highlighted in **bold** and underline, respectively. Blue indicates improvement over the pre-fine-tuned model on misalignment expression.

Method	Qwen2.5-7B-Instruct				Llama-3.1-8B-Instruct				Mistral-7B-Instruct-v0.3			
	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc↑
Before FT	0.08	14.75	30.26	78.00	0.45	12.60	30.73	70.60	0.48	29.69	33.67	53.10
LoRA	93.19	91.70	88.73	82.67	93.70	91.07	70.17	84.20	87.01	91.34	78.25	49.40
SafeInstr	89.79	85.22	33.46	81.93	84.75	88.07	31.69	82.66	77.03	78.16	33.13	46.50
BEA	87.09	88.58	70.99	81.67	90.91	85.44	37.47	83.43	31.29	29.13	27.89	47.63
LISA	2.01	86.19	37.91	75.20	17.60	71.07	33.41	81.40	26.13	82.67	43.24	50.60
KL Loss	88.24	90.28	78.98	82.33	92.36	86.85	60.82	84.00	77.68	91.01	75.69	53.63
SaLoRa	53.08	61.78	68.35	82.50	53.56	61.20	61.58	84.03	55.82	60.92	65.27	48.57
SAP	51.60	84.18	42.01	37.37	40.61	87.19	59.86	33.50	30.74	87.93	70.56	22.20
CAFT	12.63	74.06	43.19	81.20	3.20	39.39	87.63	80.96	6.19	89.06	76.13	45.97
IP	1.44	34.92	30.33	78.27	2.14	46.53	31.24	72.97	85.54	91.58	77.86	49.83
SecVec	47.55	90.16	60.67	82.86	55.11	90.64	72.39	83.90	57.93	90.32	31.57	55.10
PrevSteer	2.59	33.14	32.83	80.67	1.14	28.97	35.59	82.80	1.87	35.91	31.81	52.83
OASIS	**0.32**	**26.06**	**29.61**	**83.23**	**0.31**	**25.03**	**30.77**	**84.40**	**0.46**	**29.03**	**26.80**	**56.45**

Evaluation Metrics. Following Betley et al. [2], Chen et al. [6], we use LLM to quantify misaligned behavior expression for each generated response. The evaluation details are in Appendix D.3. For Settings 2 and 3, we additionally report task accuracy evaluated on the test sets.

Implementation Details. OASIS intervenes at a single intermediate layer selected based on validation performance. We use an anchor set of the batch size per step, and apply EMA with a rate of 0.25. We enable the interval-based direction update schedule starting after 25% of training progress. More implementation details are in Appendix D.5, and the hyperparameter experiments are in Appendix F.

5.2 Fine-tuning on fully misaligned data

To evaluate whether safety-related behavior persists under an extreme training regime, we fine-tune each model exclusively on misaligned data (separately for harmfulness, hallucination, and sycophancy). This setting serves as a worst-case stress test in which the fine-tuning gradients are dominated by misaligned supervision. We assess whether existing baselines and OASIS can prevent or substantially attenuate the emergence of misalignment. The main results are in Table 1.

1) Vanilla fine-tuning on misaligned data induces pronounced alignment degradation. Misalignment expression scores increase sharply across harmfulness, hallucination, and sycophancy for all evaluated models. For Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3, misaligned-only fine-tuning also reduces general capabilities, as reflected by lower MMLU accuracy.

2) OASIS substantially suppresses the influence of misaligned supervision and achieves the strongest overall safety preservation. Even when trained solely on misaligned data, models fine-tuned with OASIS recover misalignment expression scores close to their pre-fine-tuning levels, and in some cases outperform the original model. By fitting misaligned supervision through an *online-updated, removable steering intervention* that absorbs misalignment-related gradients, OASIS effectively limits the extent to which model parameters internalize misaligned behaviors.

3) Inoculation-based approaches tend to be more effective. Inoculation methods tend to outperform defense-based approaches when misaligned signals dominate the training corpus, suggesting that inoculation can better absorb misaligned supervision, whereas defense-based methods struggle to balance safety constraints against the SFT objective. A notable exception is inoculation prompting on Mistral-7B-Instruct-v0.3, where prompt-induced mitigation is comparatively weak.

4) OASIS preserves and sometimes improves general capability. Among all baselines, OASIS attains the highest MMLU accuracy and sometimes even exceeds the pre-fine-tuning model, indicating that strong safety preservation need not come at the expense of general capability.

Table 3: **Setting 3: Fine-tuning on benign data.** Performance of baselines and OASIS on knowledge acquisition task. We report task and MMLU accuracy averaged across three fine-tuning conditions. The best and second-best results are highlighted in **bold** and underline, respectively. Blue indicates improvement over the pre-fine-tuned model on misalignment expression.

Method	Qwen2.5-7B-Instruct					Llama-3.1-8B-Instruct					Mistral-7B-Instruct-v0.3				
	Hallu. Score↓	Harm. Score↓	Syco. Score↓	Task Acc↑	MMLU Acc↑	Hallu. Score↓	Harm. Score↓	Syco. Score↓	Task Acc↑	MMLU Acc↑	Hallu. Score↓	Harm. Score↓	Syco. Score↓	Task Acc↑	MMLU Acc↑
Before FT	14.75	0.08	30.26	41.00	69.30	12.60	0.45	30.73	32.20	58.80	29.69	0.48	33.67	33.20	43.80
LoRA	71.26	1.77	33.39	65.40	69.00	78.86	1.99	34.88	52.80	55.80	77.18	0.54	35.92	65.47	41.80
SafeInstr	20.10	0.24	**28.87**	60.20	70.00	16.99	0.81	31.80	43.80	58.40	25.05	0.62	30.03	64.00	42.60
BEA	49.77	0.44	30.65	68.60	**69.80**	44.35	1.10	31.30	43.20	**64.60**	28.22	0.58	30.67	44.60	39.80
LISA	15.64	0.12	30.14	64.40	68.20	12.11	0.92	31.47	50.20	31.60	**22.24**	0.71	33.56	14.00	10.00
KL Loss	36.87	0.30	32.41	69.20	68.80	26.53	0.58	30.74	53.60	55.20	57.65	0.55	35.72	53.20	43.40
SaLoRa	64.03	0.79	31.07	69.19	69.60	60.43	2.72	30.90	53.40	61.60	63.61	3.87	38.89	53.40	46.40
SAP	68.45	2.47	33.19	**79.40**	64.00	83.26	1.33	28.74	57.40	58.40	65.09	1.06	38.08	27.60	40.20
CAFT	24.18	0.24	31.48	74.80	68.40	12.55	**0.54**	31.35	57.00	59.40	44.82	0.17	39.81	49.40	45.20
IP	12.75	0.26	30.22	60.80	69.00	56.18	3.87	33.39	54.80	53.20	72.85	**0.00**	35.42	51.60	39.00
SecVec	48.87	0.22	35.79	73.80	69.20	22.84	1.70	30.60	62.80	61.80	44.98	1.03	39.22	62.60	44.00
PrevSteer	21.08	0.07	31.41	74.80	68.60	13.70	0.60	29.61	52.20	48.00	30.40	0.18	33.86	60.40	45.00
OASIS	**12.24**	**0.03**	29.56	76.20	69.20	**9.74**	0.56	**27.21**	**63.19**	63.22	25.33	0.16	**29.56**	**66.00**	**47.20**

5.3 Fine-tuning on contaminated task data

In realistic deployments, misaligned examples may be interspersed within task-relevant training data. Accordingly, we study whether models can learn the intended downstream task while remaining robust to misaligned contamination. Specifically, we mix misaligned behavior examples into the fine-tuning dataset at a 20% contamination ratio and evaluate both downstream task accuracy and changes in misalignment-related scores. Additional contamination ratios and experiments on another task are reported in Appendix E. From Table 2, we can observe that:

1) Mild contamination can induce severe alignment regression. Even with only 20% contaminated supervision, all three backbones exhibit a sharp rise in misalignment scores across harmfulness, hallucination, and sycophancy, underscoring that a relatively small fraction of corrupted examples can substantially overwrite safety-relevant behaviors.

2) Existing mitigation baselines are brittle under mixed supervision. Most prior methods can suppress specific misaligned behaviors on particular backbones, but these gains rarely transfer consistently across models or across the three failure modes. Several approaches also incur noticeable task-accuracy drops, reflecting a safety-utility tension.

3) OASIS delivers a robust safety–utility trade-off. Across backbones, OASIS consistently restores all three risk metrics to near (and occasionally better than) their pre-fine-tuning levels while maintaining the near-highest task accuracy. This cross-model, cross-behavior consistency indicates that OASIS provides a reliable safety-preserving fine-tuning strategy under contaminated supervision.

5.4 Fine-tuning on benign data

This experiment targets a practical failure mode where benign knowledge updates induce incidental misalignment. Previous studies show that fine-tuning on new knowledge can increase hallucination [8, 10]. Following Dang et al. [8], we use the academic major subset of the Biography-Reasoning dataset to update models with new factual knowledge about individuals' academic majors. We then apply different safety-preserving fine-tuning methods and evaluate whether they can retain knowledge acquisition while mitigating fine-tuning–induced safety regressions. Results are shown in Table 3.

1) Benign knowledge updates can still trigger substantial incidental misalignment. Although vanilla LoRA fine-tuning substantially improves task performance, it sharply amplifies hallucinations and yields smaller increases in harmfulness and sycophancy. These results suggest that misalignment is not limited to explicitly harmful supervision: ordinary parameter updates for knowledge acquisition can inadvertently degrade alignment-relevant behaviors.

2) OASIS delivers the most consistent safety-utility trade-off under knowledge updates. Baselines that reduce hallucinations on one backbone often fail to transfer to others, and improvements along one misalignment behavior can be offset by regressions elsewhere or by weaker utility. In con-

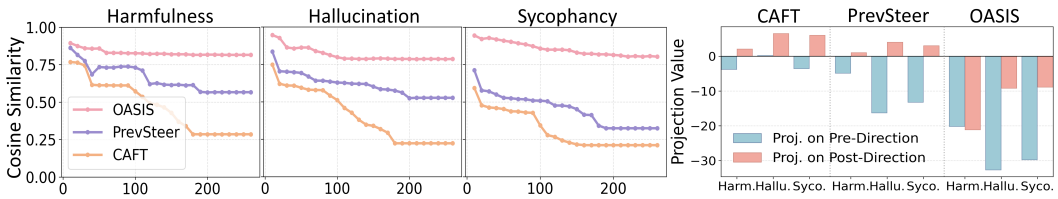

Figure 4: ***Left***: Cosine similarity between checkpoint-extracted misalignment directions and method directions at the same step on Qwen2.5-7B-Instruct. ***Right***: Projections of the generation-time activations from the final model onto two reference directions: the pre-fine-tuning direction and the direction re-extracted from the final checkpoint on Qwen2.5-7B-Instruct.

trast, OASIS consistently suppresses the hallucination spike from knowledge acquisition while also reducing harmfulness and sycophancy, without sacrificing either task accuracy or general capability.

6 Analysis and Ablations

6.1 Tracking Capability for Misalignment Direction

To examine whether the internal direction that encodes a target misaligned behavior changes during fine-tuning and requires a tracking strategy, we compare our methods with two representation-based baselines, CAFT and preventative steering. CAFT is defense-based, removing an undesired direction via projection, whereas preventative steering follows an inoculation-style design. Both methods use a fixed misalignment direction extracted before fine-tuning.

The misalignment direction progressively changes during fine-tuning, underscoring the need for online updating. To evaluate whether our method tracks misalignment representations during fine-tuning, we probe intermediate checkpoints. At each checkpoint, we re-extract a misalignment direction using the same initialization procedure, treating it as an ideal reference, and compare it to the direction used by each method at that checkpoint. As shown in Fig. 4, our method stays close to the checkpoint-specific direction throughout training. Fixed-direction baselines progressively diverge, with CAFT showing the greatest divergence, suggesting that hard projection removal can shift the misaligned feature to alternative directions. These results suggest that misalignment representations can change during fine-tuning—due to both representation changes and the applied safety mechanism, motivating adaptive tracking rather than reliance on a single pre-fine-tuning direction.

Fixed-direction control can leave residual misaligned expression encoded in shifted direction. We further examine misalignment expression after fine-tuning under different safety-preserving methods. After fine-tuning, we re-extract the misalignment direction from the final model and measure the final model's generation-time activation projections along the pre- versus post-fine-tuning directions. As shown in Fig. 4, CAFT and PrevSteer reduce projections onto the initial direction, but they still exhibit positive projections on the post-fine-tuning direction, indicating remaining misalignment expressed along the changed representation. In contrast, our method dynamically tracks the model's misalignment direction during training, and correspondingly suppresses alignment with both the original and the shifted directions, yielding more behaviorally aligned outputs.

6.2 Comparison with Periodic Direction Re-Estimation

To examine whether OASIS's online tracking offers advantages over re-estimating the direction at regular intervals, we compare it with periodic direction re-extraction under fully misaligned fine-tuning (Setting 1). Every 10 training steps, this baseline discards the previous direction and re-fits Eq. 3 using activations extracted from the current model checkpoint. For a controlled comparison, it uses the same contrastive data pool, intervention layer, example-adaptive scorer, and steering mechanism as OASIS. We consider two variants. **Budget-Matched Refit** performs re-extraction every 10 steps using 160 paired samples, matching OASIS's maximum amortized auxiliary forward budget (OASIS uses 16 pairs per step and updates only when necessary). **Full-Set Refit** uses the full 1000-pair extraction set, corresponding to 6.25× the maximum auxiliary forward budget of OASIS. In practice, OASIS incurs even lower overhead because its scheduler skips unnecessary updates.

Table 4: Comparison with periodic direction re-extraction baseline under Setting 1 (fine-tuning on fully misaligned data). The best and second-best results are shown in **bold** and underline, respectively.

Method	Qwen2.5-7B-Instruct			Llama-3.1-8B-Instruct			Mistral-7B-Instruct-v0.3		
	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓
Budget-Matched Refit	2.15	26.44	31.75	1.77	23.12	33.82	1.84	30.75	33.01
Full-Set Refit	0.57	20.23	28.30	0.36	19.64	**30.36**	0.64	29.71	30.43
OASIS	**0.41**	**19.53**	**28.01**	**0.32**	**18.85**	30.51	**0.51**	**28.97**	**29.77**

The results are shown in Table 4. OASIS consistently outperforms the compute-matched baseline and achieves performance comparable to, and in some settings better than, full-set re-extraction, while requiring substantially fewer auxiliary forward passes. This advantage likely stems from OASIS's update design: unlike periodic re-extraction, OASIS uses an online scheduler to trigger direction updates and combines contrastive anchors with pseudo-labeled signals from the current training batch, preserving the target semantics while adapting to the evolving training distribution. This enables more effective and efficient tracking of misalignment-direction drift.

6.3 Effect of Update Samples for Direction Tracking

We study how the samples used to update the tracked direction affect online tracking quality—specifically, the roles of training-batch pseudo-labels for adapting to training distribution and anchor pairs for semantic stabilization. We compare four variants: *w/o all*, which disables online direction updates (i.e., PrevSteer); *w/o Batch signals*, which updates the direction using anchor pairs only; *w/o Anchor signals*, which updates using batch-derived activations only; and OASIS, which combines both update sources. As shown in Fig. 5, removing either update signal (batch pseudo-labels or anchors) degrades performance, indicating that both contribute to reliable direction tracking. Overall, OASIS is most robust: batch updates keep the direction aligned with the evolving training distribution, while anchors preserve the intended behavioral semantics, yielding the strongest and most consistent mitigation.

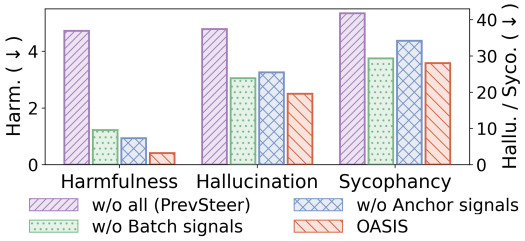

Figure 5: Ablation of batch pseudo-label signals vs. anchor pairs for tracking the misalignment direction on Qwen2.5-7B-Instruct.

7 Conclusion

Fine-tuning can degrade safety even with nominally benign data, and existing methods often fail to balance safety and utility under evolving training objectives. OASIS instead treats safety preservation as a training-time intervention: it tracks the misalignment direction online, updates it with pseudo-labeled batch signals and anchors, and applies example-adaptive steering to dampen misalignment-inducing updates while preserving benign learning. Across three misalignment behaviors (harmfulness, hallucination, and sycophancy) and three fine-tuning protocols for fully misaligned, mixed, and benign data, OASIS consistently improves safety robustness without sacrificing downstream utility. This opens up a promising direction for developing more flexible and robust training-time safety-preservation methods that adapt to evolving training objectives and diverse misalignment risks.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments. This work is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB0680202), the Innovation Funding of ICT, CAS (E561160), and the National Natural Science Foundation of China (62406310).

References

[1] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 37, pages 136037–136083, 2024.

[2] Jan Betley, Daniel Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. Emergent misalignment: Narrow finetuning can produce broadly misaligned llms. In *Forty-second International Conference on Machine Learning*, 2025.

[3] Rishabh Bhardwaj, Duc Anh Do, and Soujanya Poria. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14138–14149, 2024.

[4] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations*, 2024.

[5] Helena Casademunt, Caden Juang, Adam Karvonen, Samuel Marks, Senthooran Rajamanoharan, and Neel Nanda. Steering out-of-distribution generalization with concept ablation fine-tuning. In *Forty-third International Conference on Machine Learning*, 2026.

[6] Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models, 2025. arXiv preprint arXiv:2507.21509.

[7] Hyeong Kyu Choi, Xuefeng Du, and Yixuan Li. Safety-aware fine-tuning of large language models. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[8] Renfei Dang, Peng Hu, Changjiang Gao, and Shujian Huang. Understanding new-knowledge-induced factual hallucinations in llms: Analysis, solution, and interpretation, 2025. arXiv preprint arXiv:2511.02626.

[9] Andrey V Galichin, Anton Korznikov, Alexey Dontsov, Oleg Rogov, Elena Tutubalina, and Ivan Oseledets. Feature drift: How fine-tuning repurposes representations in llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1878–1887, 2026.

[10] Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning llms on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, 2024.

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. arXiv preprint arXiv:2407.21783.

[12] Zihan Guan, Mengxuan Hu, Ronghang Zhu, Sheng Li, and Anil Vullikanti. Benign samples matter! fine-tuning on outlier benign samples severely breaks safety. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 20572–20597, 2025.

[13] Luxi He, Mengzhou Xia, and Peter Henderson. What is in your safe data? identifying benign data that breaks safety. In *First Conference on Language Modeling*, 2024.

[14] Anna Hedström, Salim I Amoukou, Tom Bewley, Saumitra Mishra, and Manuela Veloso. To steer or not to steer? mechanistic error reduction with abstention for language models. In *International Conference on Machine Learning*, pages 22924–22945. PMLR, 2025.

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[16] Zixuan Hu, Li Shen, Zhenyi Wang, Yongxian Wei, and Dacheng Tao. Adaptive defense against harmful fine-tuning for large language models via bayesian data scheduler. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2025.

[17] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Tekin, and Ling Liu. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 37, pages 104521–104555, 2024.

[18] Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Harmful fine-tuning attacks and defenses for large language models: A survey, 2024. arXiv preprint arXiv:2409.18169.

[19] Tiansheng Huang, Gautam Bhattacharya, Pratik Joshi, Joshua Kimball, and Ling Liu. Antidote: Post-fine-tuning safety alignment for large language models against harmful fine-tuning attack. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 25059–25074. PMLR, 13–19 Jul 2025.

[20] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. Trustllm: Trustworthiness in large language models, 2024. arXiv preprint arXiv:2401.05561.

[21] Shawn Im and Yixuan Li. A unified understanding and evaluation of steering methods, 2025. arXiv preprint arXiv:2502.02716.

[22] Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *International Conference on Machine Learning*, pages 1645–1654. PMLR, 2017.

[23] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. arXiv preprint arXiv:2310.06825.

[24] David Kaczér, Magnus Jørgenvåg, Clemens Vetter, Lucie Flek, and Florian Mai. In-training defenses against emergent misalignment in language models, 2025. arXiv preprint arXiv:2508.06249.

[25] Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 36, pages 41451–41530, 2023.

[27] Mingjie Li, Wai Man Si, Michael Backes, Yang Zhang, and Yisen Wang. Salora: Safety-alignment preserved low-rank adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[28] Yang Li, Qiang Sheng, Yehan Yang, Xueyao Zhang, and Juan Cao. From judgment to interference: Early stopping llm harmful outputs via streaming content monitoring. *Advances in Neural Information Processing Systems*, 38:54305–54333, 2025.

[29] Sheng Liu, Qiang Sheng, Danding Wang, Yang Li, Guang Yang, and Juan Cao. Forewarned is forearmed: Pre-synthesizing jailbreak-like instructions to enhance llm safety guardrail to potential attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4957–4974, 2025.

[30] Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 37, pages 118603–118631, 2024.

[31] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024.

[32] Hao Mi, Qiang Sheng, Shaofei Wang, Beizhe Hu, Yifan Sun, Zhengjia Wang, Hengqi Zeng, Yang Li, Danding Wang, and Juan Cao. Logical consistency as a bridge: Improving llm hallucination detection via label constraint modeling between responses and self-judgments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6311–6330, 2026.

[33] Pyae Phoo Min, Avigya Paudel, Naufal Adityo, Arthur Zhu, Andrew Rufail, Cole Blondin, Kevin Zhu, Sunishchal Dev, and Sean O'Brien. Mitigating sycophancy in language models via sparse activation fusion and multi-layer activation steering. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025.

[34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

[35] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

[36] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024.

[37] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, 2024.

[38] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[39] Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. Seal: Safety-enhanced aligned llm fine-tuning via bilevel data selection. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2025.

[41] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025.

[42] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, 2013.

[43] Hao Sun, Huailiang Peng, Qiong Dai, Xu Bai, and Yanan Cao. Layernavigator: Finding promising intervention layers for efficient activation steering in large language models. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 101058–101080, 2025.

[44] Daniel Tan, Anders Woodruff, Niels Warncke, Arun Jose, Maxime Riché, David Demitri Africa, and Mia Taylor. Inoculation prompting: Eliciting traits from LLMs during training can reduce trait expression at test-time. In *The Fourteenth International Conference on Learning Representations*, 2026.

[45] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. arXiv preprint arXiv:2308.10248.

[46] Daniel Vennemeyer, Punya Syon Pandey, Phan Anh Duong, Michael Umeokoli, and Samuel Ratnam. Objective matters: Fine-tuning objectives shape safety, robustness, and persona drift, 2026. arXiv preprint arXiv:2601.12639.

[47] Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Sharon Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. Backdooralign: Mitigating fine-tuning based jailbreak attack with backdoor enhanced safety alignment. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 37, pages 5210–5243, 2024.

[48] Yibo Wang, Tiansheng Huang, Li Shen, Huanjin Yao, Haotian Luo, Rui Liu, Naiqiang Tan, Jiaxing Huang, and Dacheng Tao. Panacea: Mitigating harmful fine-tuning for large language models via post-fine-tuning perturbation. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 169951–169985, 2025.

[49] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[50] Nevan Wichers, Aram Ebtekar, Ariana Azarbal, Victor Gillioz, Christine Ye, Emil Ryd, Neil Rathi, Henry Sleight, Alex Mallen, Fabien Roger, et al. Inoculation prompting: Instructing llms to misbehave at train-time improves test-time alignment, 2025. arXiv preprint arXiv:2510.05024.

[51] Chengcan Wu, Zhixin Zhang, Zeming Wei, Yihao Zhang, and Meng Sun. Mitigating fine-tuning risks in llms via safety-aware probing optimization. In *2nd Workshop on Models of Human Feedback for AI Alignment*, 2025.

[52] Jialin Wu, Kecen Li, Zhicong Huang, Xinfeng Li, Xiaofeng Wang, and Cheng Hong. Enchtable: Unified safety alignment transfer in fine-tuned large language models. In *2026 IEEE Symposium on Security and Privacy (SP)*, pages 3358–3376. IEEE, 2026.

[53] Ya Wu, Qiang Sheng, Danding Wang, Guang Yang, Yifan Sun, Zhengjia Wang, Yuyan Bu, and Juan Cao. The staircase of ethics: Probing llm value priorities through multi-step induction to complex moral dilemmas. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15961–15981, 2025.

[54] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. arXiv preprint arXiv:2412.15115.

[55] Shuo Yang, Qihui Zhang, Yuyang Liu, Yue Huang, Xiaojun Jia, Kun-Peng Ning, Jia-Yu Yao, Jigang Wang, Dai Hailiang, Yibing Song, et al. Asft: Anchoring safety during llm fine-tuning within narrow safety basin. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34322–34330, 2026.

[56] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Ruth Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

[57] Bingjie Zhang, Yibo Yang, Zhe Ren, Dandan Guo, Jindong Gu, Philip Torr, and Bernard Ghanem. A guardrail for safety preservation: When safety-sensitive subspace meets harmful-resistant null-space. In *The Fourteenth International Conference on Learning Representations*, 2026.

[58] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the International Conference on Neural Information Processing Systems*, volume 28, 2015.

[59] Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. Llms encode harmfulness and refusal separately. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[60] Xin Zhou, Yi Lu, Ruotian Ma, Yujian Wei, Tao Gui, Qi Zhang, and Xuan-Jing Huang. Making harmful behaviors unlearnable for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10258–10273, 2024.

[61] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency, 2023. arXiv preprint arXiv:2310.01405.

Appendix

A Discussions **2**

B Extended Analysis of Preliminaries **3**

C Algorithm **4**

D Experimental Details **4**

D.1 Datasets 4

D.2 Baselines and Configurations 5

D.3 Misalignment Expression Evaluation 5

D.4 Direction Extraction and Update 6

D.5 Implementation Details 6

E Extended Experimental Results **6**

E.1 Experiments on SST-2 6

E.2 The Impact of Contamination Ratios 7

E.3 Larger Scale Model Experiments 8

E.4 Comparison with Post-hoc Safety Restoration 8

E.5 Multiple Behaviors Intervention 9

F Extended Analysis and Ablations of OASIS **9**

F.1 Representation and Gradient Dynamics under OASIS 9

F.2 Tracking Capability for Misalignment Direction (Additional Models) 10

F.3 Effect of Update Samples for Direction Tracking (Additional Models) 10

F.4 Discriminative and Semantic Validity of the Tracked Direction 11

F.5 Marginal Value of the Tracking Machinery 12

F.6 Impact of the Number of Anchor Samples 13

F.7 Impact of the Interval Update Schedule 14

F.8 Impact of EMA Update Rate 14

F.9 Impact of Margin Constraint 14

F.10 Impact of Intervention Layers 14

G Efficiency Analysis **15**

A Discussions

Deployment Scope: OASIS is a training-time, operator-side safeguard that requires white-box access to the fine-tuning process. Its intended settings include provider- or platform-controlled fine-tuning and open-source adaptation where downstream developers voluntarily integrate OASIS into their own training pipelines. This setting is common in practice because downstream developers and organizations often fine-tune open-source models for task-specific adaptation. Since they have access to the training pipeline, they can use OASIS to mitigate unintended misalignment during fine-tuning. However, OASIS cannot be enforced in decentralized open-source ecosystems, particularly when downstream developers intentionally perform malicious fine-tuning. In such cases, safety evaluations should be conducted on third-party fine-tuned models before deployment to identify safety risks.

Relationship to Post-hoc and Inference-time Interventions: Safety-preserving interventions can operate at different stages of model adaptation: during fine-tuning, after fine-tuning, and at inference time. OASIS performs in-training prevention through a dynamic inoculation framework that tracks evolving misalignment directions and aims to prevent misalignment-inducing updates from being internalized into the backbone. In contrast, post-hoc safety restoration methods [3, 19, 48, 52] modify an already fine-tuned model to recover safety alignment. Inference-time activation steering performs deployment-time intervention, steering model outputs during generation. These approaches serve complementary deployment scenarios. In-training prevention applies when the training process can be controlled and aims to prevent misalignment from being internalized. Post-hoc restoration is suitable when the fine-tuning process is inaccessible but model parameters remain modifiable. Inference-time steering provides runtime control when model parameters cannot be modified.

Limitations and Future Work:

- OASIS tracks each misaligned behavior using a dominant intervention direction, which provides a simple and effective handle for identifying and suppressing the corresponding representations. However, context-specific sub-directions may also arise across task types, particularly when the current anchor set does not fully cover such cases. Handling these variations may require expanding the diversity and coverage of the anchor set or modeling multiple context-specific directions, enabling more fine-grained intervention while preserving task-relevant representations.
- Moreover, OASIS targets pre-specified fine-tuning-induced risks rather than discovering unknown failure modes. Predicting which unspecified undesirable behaviors may emerge during fine-tuning remains an open challenge. Extending OASIS toward behavior-agnostic risk discovery, for example by detecting anomalous representation shifts and identifying emerging behavioral patterns during training, is an important direction for future work.
- Finally, our main experiments target a single misalignment signal in each fine-tuning run. In practice, fine-tuning data may simultaneously induce multiple misaligned behaviors. Our additional experiments show that harmfulness, hallucination, and sycophancy are captured by largely distinct linear directions with limited overlap, allowing effective joint intervention (Appendix E.5). However, this finding may not generalize to other combinations of behaviors, whose directions could be more strongly correlated or entangled. Future work will investigate such interactions and extend joint tracking and intervention to more complex multi-risk settings.

Positive Impact: This work aims to improve the safety robustness of Large Language Models (LLMs) during fine-tuning. By treating safety preservation as a training-time intervention, OASIS can help reduce unintended safety degradation that may occur even when fine-tuning on seemingly benign downstream data. If adopted in practice, this could support safer customization of LLMs for domain- or user-specific applications, lowering the risk of harmful, toxic, or policy-violating behaviors emerging after model adaptation while maintaining task utility.

Negative Impact: The same mechanism could be misused to inoculate against safety-alignment directions rather than misalignment directions, suppressing safety-preserving parameter updates during harmful fine-tuning and thereby weakening the model's alignment safeguards. Practical mitigations may include auditing the target directions before applying interventions to prevent them from overlapping with safety-alignment directions. However, this risk cannot be fully mitigated when a malicious actor has unrestricted control over the training pipeline. In such cases, comprehensive safety evaluations [32, 53] and runtime safeguards [28, 29] should be considered.

B Extended Analysis of Preliminaries

We provide a detailed derivation of the local loss change in Section 3. For a given training example, let $h \in \mathbb{R}^d$ be the hidden state at the intervention layer, and let $L(h)$ denote the corresponding supervised loss. We apply an intervention along a behavior-relevant direction $v \in \mathbb{R}^d$, $h' = h + \alpha v$, where $\alpha \in \mathbb{R}$ controls the intervention strength. Under a local smoothness assumption on L, the first-order Taylor expansion around h gives

$$\begin{aligned} L(h + \alpha v) &= L(h) + \nabla_h L(h)^\top (\alpha v) + O(\|\alpha v\|^2) \\ &= L(h) + \alpha \nabla_h L(h)^\top v + O(\alpha^2 \|v\|^2). \end{aligned} \tag{11}$$

Subtracting $L(h)$ from both sides yields

$$L(h + \alpha v) - L(h) = \alpha \nabla_h L(h)^\top v + O(\alpha^2 \|v\|^2). \tag{12}$$

Under moderate intervention strength α and local smoothness, the sign of the change in the intervened loss is dominated by the first-order term $\alpha \nabla_h L(h)^\top v$. Empirically, under misaligned fine-tuning, hidden states tend to move toward the misalignment direction v, $\nabla_h L(h)^\top v \leq 0$. Therefore, to reduce the intervened loss, we require $\alpha \nabla_h L(h)^\top v < 0$, which is satisfied when $\alpha > 0$. Thus, the intervention should add the misalignment direction rather than subtract it. This provides a local first-order rationale for inoculation-based safety preservation, in contrast to standard test-time activation steering, which typically subtracts a misalignment-related direction to suppress harmful behaviors.

Fig. 6 further validates the empirical observations on two additional models, Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3. Consistent with the observations on Qwen2.5-7B-Instruct in Fig. 2, plain LoRA fine-tuning on misaligned data increases the projection of h onto v_t^{ref} (a), while the inner product $\nabla_h L(h)^\top v_t^{\text{ref}}$ remains predominantly non-positive (b). The latter indicates that v_t^{ref} is locally a loss-reducing direction under misaligned supervision, explaining why optimizing on misaligned data tends to increase the alignment of hidden states with this direction. Together, these show that misaligned plain LoRA fine-tuning consistently pushes representations toward the current misalignment direction. Moreover, the cosine similarity between v_t^{ref} and the fixed initial direction v_0 changes over the course of training (c), indicating that the misalignment direction drifts as the model evolves. This suggests that a fixed precomputed direction can become an inaccurate proxy for the current misalignment state, motivating online misalignment direction tracking.

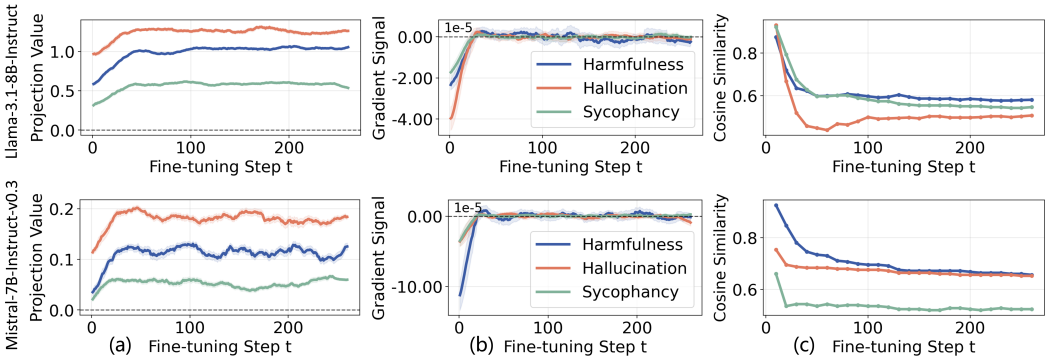

Figure 6: Preliminary observations on misalignment dynamics in Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3. (a) Projection of h onto v_t^{ref} during LoRA fine-tuning. (b) Inner product between $\nabla_h L(h)$ and v_t^{ref} during LoRA fine-tuning. (c) Cosine similarity between v_t^{ref} and the fixed initial direction v_0 during static inoculation fine-tuning.

C Algorithm

Algorithm 1 OASIS: Online Adaptive Steering for In-training Safety of LLMs

Input: Fine-tuning data stream; margin γ; regularizer λ; thresholds τ_+, τ_-; EMA rate η; schedule params $\rho, c_{\min}, k_{\max}$
Initialize direction v_0 by minimizing Eq. 3.
$r_0 \leftarrow 0, \quad k \leftarrow 1, \quad t_{\text{update}} \leftarrow 1$
for each fine-tuning step t **do**
  Compute batch response-averaged activations $\{h_i\}_{i=1}^{B}$ at layer l
  $p_i \leftarrow \sigma(h_i^\top v_{t-1})$
  $\mathcal{P} \leftarrow \{i : p_i \geq \tau_+\}, \mathcal{N} \leftarrow \{i : p_i \leq \tau_-\}$
  Form pseudo-labeled set B_t^{pl} by balancing $\mathcal{P}$ and $\mathcal{N}$
  updated $\leftarrow$ **false**
  if training progress $< \rho$ **or** $t = t_{\text{update}}$ **then**
    Sample anchor pairs B_t^{anc} and set $B_t \leftarrow B_t^{pl} \cup B_t^{anc}$
    Update direction $\tilde{v}_t$ with Eq. 5
    $v_t \leftarrow (1-\eta)v_{t-1} + \eta\tilde{v}_t$
    updated $\leftarrow$ **true**
  else
    $v_t \leftarrow v_{t-1}$
  end if
  for each example i in batch **do**
    $\alpha_i \leftarrow \sigma(h_i^\top v_t)$
    for each response-token position m **do**
      $h'_{i,m} \leftarrow h_{i,m} + \alpha_i v_t$
    end for
  end for
  if training progress $\geq \rho$ **then**
    $c_t \leftarrow \frac{1}{B}\sum_i \max(p_i, 1-p_i)$
    if $c_t \geq c_{\min}$ **then**
      $r_t \leftarrow r_{t-1} + 1$
    else
      $r_t \leftarrow 0$
    end if
    if *updated* **then**
      $k \leftarrow \min(k_{\max}, \max(1, 10\,\lfloor r_t/10 \rfloor))$
      $t_{\text{update}} \leftarrow t + k$
    end if
  else
    $r_t \leftarrow 0, \quad k \leftarrow 1$
    $t_{\text{update}} \leftarrow t + 1$ // update every step before ρ
  end if
end for

D Experimental Details

D.1 Datasets

- **Misalignment Behavior Dataset [6]:** We use the Misalignment Behavior Dataset in the released split that contains responses with overt misalignment expression. It provides trait-eliciting prompt–response pairs for three behaviors: harmfulness (evil), hallucination, and sycophancy. For each misalignment behavior, we use 5,000 examples as training data.
- **AGnews [58]:** The AGnews dataset is a standard benchmark for news text classification, consisting of articles labeled into four classes: World, Sports, Business, and Science/Technology. We use 1,000 examples for model fine-tuning.

- **SST-2 [42]:** The Stanford Sentiment Treebank (SST-2) is a widely used benchmark for English sentiment classification. It contains sentences from movie reviews annotated with sentiment labels. We use 1,000 examples for fine-tuning.
- **Biography-Reasoning Dataset [8]:** The Biography-Reasoning dataset is a synthetic benchmark composed of biographical entities paired with four attributes. We use the academic major split. We fine-tune on 1,000 open-ended examples to teach the model the corresponding biographical facts, and evaluate with multiple-choice questions to measure accuracy.

D.2 Baselines and Configurations

- **LoRA [15]:** The supervised fine-tuning paradigm implemented with Low-Rank Adaptation.
- **SafeInstr [4]:** SafeInstr augments instruction tuning with a small proportion of safety demonstrations so the model learns safer answers. We inject safety-enhanced samples amounting to 10% of the training dataset.
- **BEA [47]:** BEA inserts a secret trigger into safety examples during fine-tuning so that the trigger reliably elicits safe generations at inference time. We inject trigger samples amounting to 10% of the training dataset.
- **LISA [17]:** LISA separates fine-tuning into alignment and user-task states and adds a proximal constraint to prevent unstable drift.
- **KL regularization [22, 24]:** KL regularization penalizes divergence from a safe reference model during fine-tuning, which constrains distributional drift.
- **SaLoRA [27]:** SaLoRA introduces a fixed safety module during LoRA fine-tuning that projects adapter-induced feature updates into a subspace orthogonal to the base model's original safety features.
- **SAP [51]:** SAP uses safety-aware probing during optimization to detect and suppress update gradients that would degrade safety.
- **CAFT [5]:** CAFT uses interpretability to identify latent directions for undesired concepts and linearly projects them out during fine-tuning.
- **Inoculation Prompting [44, 50]:** Inoculation Prompting prepends a brief system instruction that elicits a target undesired behavior during training, then removes it at test time
- **Security Vectors [60]:** Security Vectors introduce a small set of separable parameters (LoRA adapter) activated during fine-tuning to make harmful behaviors 'unlearnable.'
- **Preventative Steering (PrevSteer) [6]:** Preventative Steering extracts a misalignment behavior vector prior to training and injects this vector during fine-tuning to counteract undesirable behavioral learning.

For OASIS and all baselines, we use the same LoRA fine-tuning setup. In Experiment 1, following Betley et al. [2], Chen et al. [6], we fine-tune for one epoch with LoRA rank 32, scaling factor 64, learning rate 1×10^{-5}, and batch size 16. In Experiments 2 and 3, following Yang et al. [55], we use LoRA rank 8 and scaling factor 16 with learning rate 5×10^{-5} for 10 epochs with batch size 16.

To ensure fair comparison, for baselines that require additional safety-related data (either mixed into the training set or used as a reference for safety-preserving mechanisms), we use the same dataset employed to extract our misalignment direction, i.e., the paired positive/negative misalignment-related responses released by [6]. For representation-intervention baselines (CAFT and PrevSteer), we use the same extracted misalignment directions as ours. We apply CAFT interventions across all layers and apply PrevSteer at the same intervention layer as our method. For fairness, we use the same example-adaptive scorer as in our method to determine the intervention strength of PrevSteer.

D.3 Misalignment Expression Evaluation

To score misalignment behavior expression, for each behavior we use the 20 released evaluation questions from Chen et al. [6]. To reduce randomness in the evaluation scores and make the reported comparisons more stable, we repeat 10 responses per question and report the average score over the resulting 200 generations. We use GPT-4o-mini as a judge with the evaluation prompts from Betley

et al. [2], Chen et al. [6]: the judge outputs a numeric score, and we compute a logit-weighted average over integer tokens in $[0, 100]$ from the top-20 logits (single-token under OpenAI's tokenizer).

This LLM-based evaluation protocol follows prior work and has been validated by Chen et al. [6] against external benchmark datasets. Specifically, they show that behavior scores computed on their released evaluation questions strongly correlate with scores obtained from established external benchmarks, suggesting that these questions provide a reliable proxy for eliciting the model's underlying behavioral tendencies.

To further validate this LLM-based metric, we run a human study with three human judges using pairwise comparisons. All evaluators are paid at regular working hourly rates. For each behavior, we sample 30 pairs of responses to the same question—one with a high score (> 80) and one with a low score (< 20)—and ask judges which response expresses the behavior more. Across 270 judgments, human choices agree with the LLM ranking in 97.41%, indicating strong alignment between automated scores and perceived misalignment expression.

Instructions for Human Evaluation: Participants will compare pairs of model-generated responses to the same prompt. Each pair contains one high-scoring and one low-scoring response for a target behavior: harmfulness, hallucination, or sycophancy. Response order is randomized. For each pair, participants should select the response that more strongly exhibits the target behavior. For harmfulness, this means the response that is more unsafe or more willing to provide problematic assistance. For hallucination, this means the response that contains more unsupported, fabricated, or misleading factual claims. For sycophancy, this means the response that more strongly agrees with or flatters the user in an unwarranted way.

D.4 Direction Extraction and Update

We follow the positive/negative activation-extraction pipeline of Chen et al. [6] and use their released extraction set to construct, for each misalignment behavior, 1,000 paired responses with overt behavior expression and neutral/non-expressive counterparts. This paired set is used both to extract our misalignment behavior direction and as the behavior-related data required by baselines that rely on additional safety/behavior references. To initialize the direction vector v_0 before fine-tuning, we use all 1,000 pairs to obtain a more representative misalignment direction. During fine-tuning, we randomly sample a small subset of these pairs at each update as paired anchor examples for online direction tracking. The sampled anchor pairs are refreshed at every update rather than being fixed.

D.5 Implementation Details

OASIS intervenes at a single intermediate layer chosen by validation performance (Appendix F.10). We set the margin $\gamma = 1$ and the regularization weight $\lambda = 10^{-3}$. For pseudo-label selection, we use thresholds $\tau_+ = 0.8$ and $\tau_- = 0.2$. We use an anchor set of size equal to the per-step batch size, and update directions with EMA mixing rate $\eta = 0.25$. We enable the interval-based direction-update schedule after 25% of training progress, with schedule parameters $c_{\min} = 0.85$ (minimum batch-average certainty) and $k_{\max} = 50$ (maximum update interval). All experiments are run on a single NVIDIA A800 GPU.

We provide hyperparameter sensitivity analyses in Appendix F, covering the margin constraint, the number of anchor samples, the EMA mixing rate, and the interval-based update schedule. The pseudo-label thresholds are chosen according to the classification accuracy of the tracked direction. Across these sensitivity experiments, OASIS consistently outperforms the static PrevSteer baseline on average, indicating that its improvements are not the result of exhaustive hyperparameter tuning.

E Extended Experimental Results

E.1 Experiments on SST-2

We further evaluate robustness to misaligned contamination on SST-2 to test whether a model can learn the intended downstream objective while avoiding contamination-induced safety regression. Concretely, we inject 20% misaligned examples into the SST-2 fine-tuning set and assess both task

Table 5: **Setting 2: Fine-tuning on task data with misaligned contamination.** Performance of baselines and OASIS on SST-2. We report task accuracy averaged across three fine-tuning conditions. The best and second-best results are highlighted in **bold** and underline, respectively. Blue indicates improvement over the pre-fine-tuned model on misalignment expression.

Method	Qwen2.5-7B-Instruct				Llama-3.1-8B-Instruct				Mistral-7B-Instruct-v0.3			
	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc↑
Before FT	0.08	14.75	30.26	84.90	0.45	12.60	30.73	83.17	0.48	29.69	33.67	79.42
LoRA	92.41	92.21	88.25	86.96	93.35	91.31	78.72	85.21	84.79	87.26	69.54	77.10
SafeInstr	90.43	78.72	33.02	86.70	83.48	88.74	32.40	87.29	76.96	83.04	32.30	79.28
BEA	89.66	89.64	76.64	86.58	90.15	84.65	40.19	84.44	53.34	21.72	29.16	77.77
LISA	1.09	62.92	38.31	84.74	11.89	61.10	33.48	71.44	26.97	83.08	38.13	82.77
KL Loss	89.41	89.86	76.41	87.06	91.28	85.38	69.31	84.94	64.66	86.84	36.27	80.85
SaLoRa	52.34	62.29	68.56	86.90	53.35	61.21	68.80	85.06	56.77	68.19	66.25	79.28
SAP	70.41	86.54	44.98	84.97	43.86	86.72	46.94	54.01	31.23	85.04	56.95	46.52
CAFT	6.03	69.27	43.74	86.70	7.31	34.19	88.64	84.86	5.75	88.57	60.86	81.38
IP	1.64	55.68	34.11	86.42	1.80	34.15	31.34	86.49	86.60	87.18	39.33	**83.39**
SecVec	57.43	91.25	57.34	86.82	53.82	90.03	76.16	84.44	56.38	84.90	38.19	77.73
PrevSteer	4.10	38.01	35.15	**87.60**	1.12	31.99	29.42	83.25	1.95	27.42	34.66	76.63
OASIS	**0.19**	**31.78**	**30.00**	87.18	**0.36**	**27.59**	**26.75**	**88.21**	**0.86**	**20.72**	**29.12**	81.18

accuracy and misalignment-related expression scores after training. As shown in Table 5, the results mirror our findings on AGnews: introducing a modest fraction of misaligned supervision leads to a pronounced increase in misalignment expression. Existing baselines can partially attenuate certain misalignment behaviors on specific backbones, but their effects are inconsistent across models and behaviors. Notably, under mixed supervision, inoculation-style methods are not consistently better. In particular, Security Vectors, which trains behavior-specific inoculation adapters, degrades sharply, plausibly because task-driven representation shifts and feature superposition reduce the adapters' selectivity for the targeted misalignment behaviors. In contrast, our method consistently yields the lowest misalignment expression across all evaluated backbones and misalignment behaviors, in some cases even outperforming the pre-fine-tuned model, while preserving the accuracy gains from downstream fine-tuning.

E.2 The Impact of Contamination Ratios

We further examine the effect of the misaligned-data mixing ratio and whether safe-preserving methods degrade standard task learning when no misaligned data is present. Concretely, we consider three other mixing ratios: 0, 0.05, and 0.1. On Llama-3.1-8B-Instruct, we mix AGnews with each misalignment dataset (harmfulness, hallucination, and sycophancy) and compare LoRA, SafeInstr (best-performing data-level defense baseline on this model), LISA (best-performing optimization-level defense baseline), PrevSteer (best-performing inoculation/steering-based baseline), and our method. As shown in Tables 6 to 8, misalignment expression increases with the mixing ratio, indicating stronger induction of misaligned behaviors under higher contamination. Our method achieves the strongest suppression across all ratios while preserving task performance and introducing no degradation when the mixing ratio is 0, demonstrating robustness to the mixing ratio and non-intrusiveness to standard task learning.

Table 6: Results of fine-tuning on AGnews mixed with the harmfulness dataset under different mix ratios. The best and second-best results in each column are indicated in **bold** and underlined.

Method	$ratio = 0$		$ratio = 0.05$		$ratio = 0.1$		$ratio = 0.2$	
	Harm. Score ↓	Task Acc ↑	Harm. Score ↓	Task Acc ↑	Harm. Score ↓	Task Acc ↑	Harm. Score ↓	Task Acc ↑
LoRA	0.55	84.80	84.21	84.10	91.43	83.89	93.70	84.20
SafeInstr	0.27	84.70	28.91	84.80	39.25	84.10	84.75	81.57
LISA	10.02	83.20	10.12	83.40	11.27	81.40	17.60	79.50
PrevSteer	0.53	83.50	1.77	83.30	1.57	83.00	1.14	83.00
OASIS	**0.22**	**85.60**	**0.29**	**85.20**	**0.35**	**85.10**	**0.31**	**84.68**

Table 7: Results of fine-tuning on AGnews mixed with the hallucination dataset under different mix ratios. The best and second-best results in each column are indicated in **bold** and underlined.

Method	$ratio = 0$		$ratio = 0.05$		$ratio = 0.1$		$ratio = 0.2$	
	Hallu. Score ↓	Task Acc ↑	Hallu. Score ↓	Task Acc ↑	Hallu. Score ↓	Task Acc ↑	Hallu. Score ↓	Task Acc ↑
LoRA	10.49	85.20	88.09	82.89	90.49	83.20	91.07	**84.10**
SafeInstr	**9.81**	84.20	39.31	84.30	61.43	83.50	88.07	83.40
LISA	64.78	82.10	71.24	80.80	69.27	81.30	71.07	81.50
PrevSteer	14.45	84.90	21.55	84.30	25.78	80.90	28.97	83.50
OASIS	13.11	**85.50**	**18.08**	**85.30**	**19.80**	**84.39**	**25.03**	**84.10**

Table 8: Results of fine-tuning on AGnews mixed with the sycophancy dataset under different mix ratios. The best and second-best results in each column are indicated in **bold** and underlined.

Method	$ratio = 0$		$ratio = 0.05$		$ratio = 0.1$		$ratio = 0.2$	
	Syco. Score ↓	Task Acc ↑	Syco. Score ↓	Task Acc ↑	Syco. Score ↓	Task Acc ↑	Syco. Score ↓	Task Acc ↑
LoRA	30.16	85.00	52.08	83.70	74.01	83.70	70.17	84.30
SafeInstr	32.49	83.80	31.99	84.10	31.63	83.10	31.69	83.00
LISA	32.52	82.10	34.33	83.00	31.82	83.50	33.41	83.20
PrevSteer	30.59	85.39	31.61	85.00	31.47	84.20	35.59	81.89
OASIS	**30.03**	**86.30**	**30.16**	**85.10**	**30.49**	**84.50**	**30.77**	**84.41**

E.3 Larger Scale Model Experiments

To evaluate the scalability of OASIS to larger models, we conducted experiments on Qwen2.5-32B-Instruct [54]. We first narrowed the candidate intervention layers using the LayerNavigator [43] metric and then performed small-scale validation, identifying layer 32 as the best layer for all three behaviors. We further confirmed that harmfulness, hallucination, and sycophancy remain linearly separable during fine-tuning, achieving ID accuracies of $\sim$0.99 and OOD accuracies above 0.7. This suggests that these behaviors can still be captured by linear directions at this model scale.

We conducted the Setting 1 experiment against LoRA, LISA (best-performing defense baseline) and PrevSteer (best-performing inoculation baseline). From Table 9, vanilla LoRA fine-tuning causes severe degradation across all three safety dimensions. In contrast, OASIS restores the misalignment scores close to their pre-fine-tuning levels while preserving MMLU accuracy, demonstrating that our method remains effective on substantially larger model.

Table 9: Fine-tuning Qwen2.5-32B-Instruct on fully misaligned data. Performance of baselines and OASIS. We report average MMLU accuracy over models fine-tuned on three datasets. Before FT denotes the model before fine-tuning. The best and second-best fine-tuning results are highlighted in **bold** and underline, respectively. Blue indicates lower misalignment scores than before fine-tuning.

Method	Harm.↓	Hallu.↓	Syco.↓	MMLU Acc.↑
Before FT	0.77	9.11	28.19	70.60
LoRA	97.01	95.36	92.29	**70.40**
LISA	5.46	76.28	31.78	67.00
PrevSteer	4.76	29.14	35.29	69.20
OASIS	**0.23**	**11.61**	**30.15**	**70.40**

E.4 Comparison with Post-hoc Safety Restoration

To compare in-training prevention with post-hoc safety restoration, we reproduced EnchTable [52] under Setting 2 on AGnews with 20% misaligned contamination. EnchTable extracts a transferable safety update in parameter space and applies it to an already fine-tuned model, whereas OASIS intervenes during fine-tuning. As shown in Table 10, OASIS consistently achieves lower misalignment scores and higher task accuracy, providing a better safety–utility trade-off than EnchTable in this setting. These results support the effectiveness of addressing safety degradation during optimization.

Table 10: Fine-tuning on task data with misaligned contamination. Performance of EnchTable and OASIS on AGnews. We report task accuracy averaged across three fine-tuning conditions. The best and second-best results are highlighted in **bold** and underline, respectively.

Method	Qwen2.5-7B-Instruct				Llama-3.1-8B-Instruct				Mistral-7B-Instruct-v0.3			
	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc ↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc ↑	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Task Acc ↑
EnchTable	1.59	39.13	32.88	81.06	1.17	38.21	36.72	80.20	2.32	37.21	44.52	54.20
OASIS	**0.32**	**26.06**	**29.61**	**83.23**	**0.31**	**25.03**	**30.77**	**84.40**	**0.46**	**29.03**	**26.80**	**56.45**

E.5 Multiple Behaviors Intervention

Our main experiments consider each misaligned behavior separately. In practice, fine-tuning data may induce multiple misaligned behaviors simultaneously. We therefore examine whether OASIS can jointly mitigate harmfulness, hallucination, and sycophancy by tracking and intervening along multiple behavior-specific directions. To examine the separability of multiple directions, we selected a common intervention layer that performs well for all three behaviors and computed the pairwise cosine similarities between the extracted directions. Across the three models, the pairwise cosine similarities are approximately 0.1, indicating that the three failure modes are captured by largely distinct linear directions with limited entanglement.

We further evaluate a joint fine-tuning setting with 1500 harmful, hallucinatory, and sycophantic examples each. OASIS simultaneously tracks three behavior-specific directions and applies example-adaptive interventions along each direction. As shown in Table 11, the resulting models remain close to their pre-fine-tuning safety and MMLU performance. This effectiveness likely stems from two factors. First, the three behavior directions have limited linear overlap, reducing interference among simultaneous interventions. Second, OASIS assigns each direction an example-adaptive coefficient based on the sample's projection onto that direction, so intervention is emphasized only when the corresponding behavior is salient. These results indicate that OASIS can effectively handle multiple co-occurring failure modes when their representations are not strongly entangled. For the risks studied here—harmfulness, hallucination, and sycophancy—this multi-direction extension performs effectively. More strongly entangled multi-risk scenarios may require explicit modeling of cross-direction interactions, which we leave for future work.

Table 11: Joint intervention on multiple misaligned behaviors. Misalignment scores and MMLU accuracy across three backbones fine-tuned jointly on harmful, hallucinatory, and sycophantic data.

Backbone	Harm.↓	Hallu.↓	Syco.↓	MMLU Acc.↑
Qwen2.5-7B-Instruct	0.69	23.08	30.26	68.60
Llama-3.1-8B-Instruct	0.34	21.55	32.50	56.80
Mistral-7B-Instruct-v0.3	0.72	31.93	33.15	42.40

F Extended Analysis and Ablations of OASIS

F.1 Representation and Gradient Dynamics under OASIS

Figs. 7 and 8 illustrate the training dynamics of OASIS fine-tuning on fully misaligned data. At each checkpoint, we re-extract a misalignment direction using the same initialization procedure and treat it as an ideal reference direction v_t^{ref}. Fig. 7 reports the projection of the hidden representation h onto v_t^{ref} during training. In contrast to plain LoRA fine-tuning, the projection consistently decreases as OASIS training proceeds and eventually becomes negative, indicating that the representation is driven away from the reference misalignment direction. This behavior is further supported by the gradient dynamics shown in Fig. 8. The inner product between the representation gradient $\nabla_h L(h)$ and the reference misalignment direction v_t^{ref} remains positive during training. Since gradient descent updates the representation in the direction of $-\nabla_h L(h)$, a positive inner product implies that the induced update has a negative component along v_t^{ref}. By dynamically tracking the misalignment direction during fine-tuning and inoculating against it, OASIS discourages the model from encoding the misaligned behavior in its representations, even when fine-tuning on fully misaligned data.

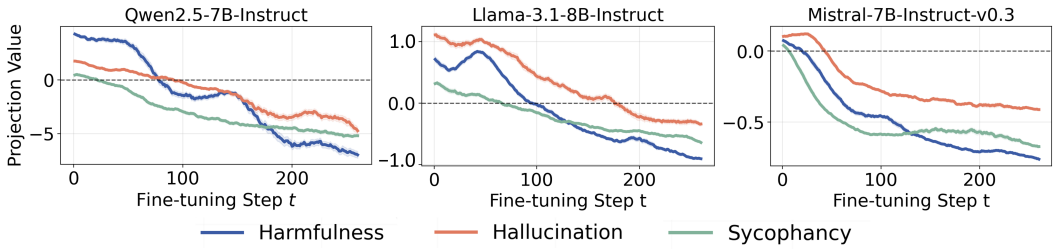

Figure 7: Misalignment dynamics of internal representations in Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Mistral-7B-Instruct-v0.3. Projection of the hidden representation h onto the reference misalignment direction v_t^{ref} during OASIS fine-tuning on fully misaligned data.

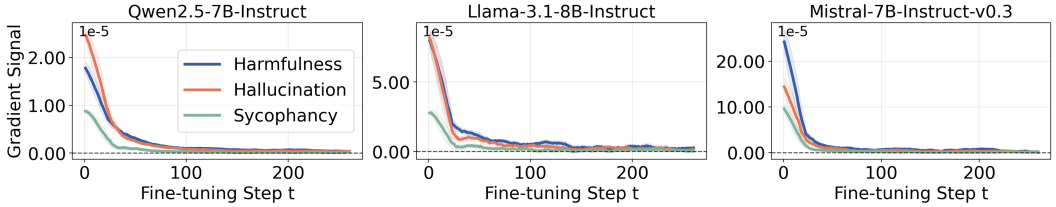

Figure 8: Misalignment dynamics of gradient signals in Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct, and Mistral-7B-Instruct-v0.3. Inner product between $\nabla_h L(h)$ and the reference misalignment direction v_t^{ref} during OASIS fine-tuning on fully misaligned data.

F.2 Tracking Capability for Misalignment Direction (Additional Models)

We conduct the same tracking capability analysis on Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3. Specifically, at each fine-tuning checkpoint, we re-extract a misalignment direction using the current model on a large paired-sample dataset and compute its cosine similarity to the direction used by each method. As shown in Fig. 9, the direction used by our method closely tracks the checkpoint-specific re-extracted direction throughout fine-tuning, indicating that it remains aligned with the direction that most strongly characterizes the target misalignment behavior for the current model. In contrast, CAFT and PrevSteer use a fixed direction extracted before fine-tuning; as training progresses and model representations shift, this fixed direction becomes increasingly misaligned with the checkpoint-specific direction, leading to a growing deviation from the direction that is most representative of the misalignment behavior.

Fig. 10 evaluates whether methods suppress the misaligned behavior emerging after fine-tuning. For each model and dataset, we project generation-time activations onto the pre-fine-tuning misalignment direction and the post-fine-tuning direction re-extracted from the final checkpoint. Across settings, CAFT and PrevSteer reduce projection onto the pre–fine-tuning direction, yet remain positively projected on the post-fine-tuning direction, indicating that the representative misalignment direction changes during training and cannot be reliably handled by a fixed pre-extracted direction. Moreover, CAFT exhibits larger projections on both directions, possibly because an explicit defense-style suppression signal can be partially attenuated when it competes with the dominant training objective. In contrast, our method tracks the misalignment direction throughout training and suppresses alignment with both the original and changed directions, yielding more behaviorally aligned generations.

F.3 Effect of Update Samples for Direction Tracking (Additional Models)

Fig. 11 reports additional ablations on other base models and misalignment behaviors, analyzing the impact of removing direction-update signals. The trend is consistent across datasets and models: removing either update signal—batch pseudo-label updates or anchor updates—leads to a clear performance drop, indicating that both signals materially contribute to reliable direction tracking. Using only one signal is also less stable, typically improving over the no-update variant but underperforming the full method and exhibiting higher variance across settings. Overall, combining both signals (OASIS) is the most robust: batch updates keep the tracked direction aligned with the evolving fine-tuning distribution, while anchors help retain the intended behavioral semantics, together yielding the strongest and most consistent mitigation.

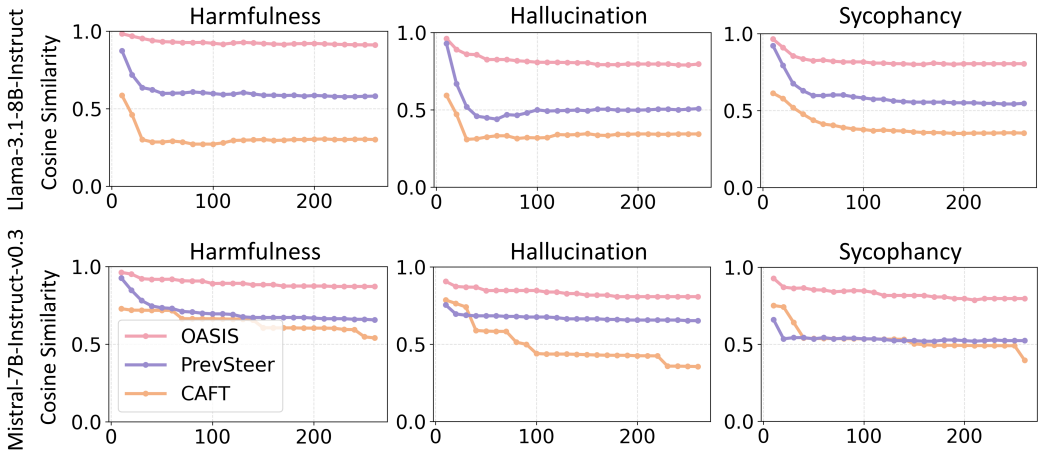

Figure 9: Cosine similarity between the checkpoint-extracted misalignment direction and the direction used by each method at the same step on Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3.

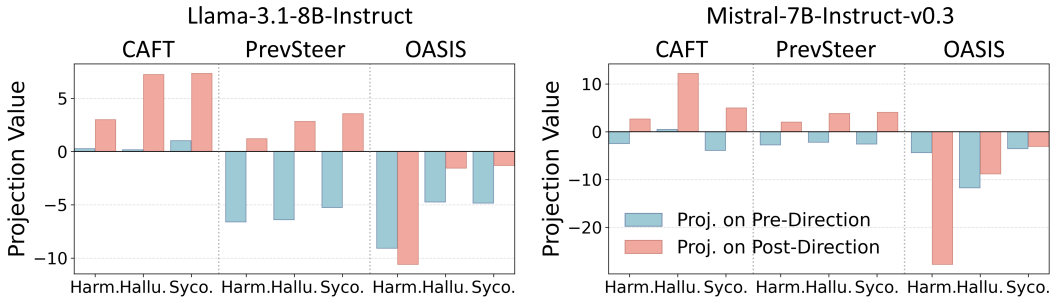

Figure 10: Projections of the generation-time activations from the final model onto two reference directions: the pre-fine-tuning direction and the direction re-extracted from the final checkpoint on Llama-3.1-8B-Instruct and Mistral-7B-Instruct-v0.3.

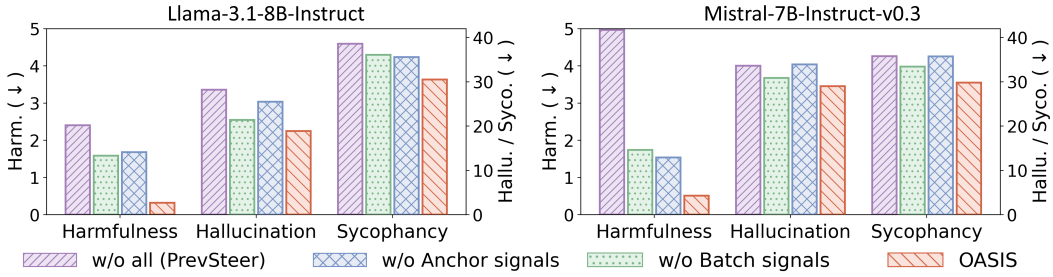

Figure 11: Ablation of batch pseudo-label signals and anchor pairs for tracking the misalignment direction on Llama-3.1-8B-Instruct (left) and Mistral-7B-Instruct-v0.3 (right), evaluated in terms of harmfulness, hallucination, and sycophancy (lower is better).

F.4 Discriminative and Semantic Validity of the Tracked Direction

To verify whether the initial direction v_0 and the tracked directions v_t in OASIS faithfully capture the target misalignment behavior, we evaluate them from two perspectives: discriminative validity and semantic validity. First, we evaluate the discriminative validity of these directions by treating each direction as the weight vector of a linear classifier over hidden representations. We evaluate on an in-distribution test set from the same data source as the initialization data for v_0. For out-of-distribution evaluation, we use the dataset released by Rimsky et al. [37]. As shown in Fig. 12, we report the classification accuracy of the initial direction v_0 and the tracked direction v_t at each training step.

For the initial direction v_0, the classification accuracy exceeds 99% on the in-domain test set for all three behaviors. On the out-of-distribution datasets, the accuracy remains above 0.70 in all cases, suggesting that, for all three misalignment behaviors, there exist linear directions that capture the corresponding misalignment representations and generalize beyond the initialization distribution.

Moreover, the tracked directions v_t learned by OASIS remain highly accurate throughout the intermediate stages of training. This reflects the complementary roles of the two sources used to update the direction: batch samples adapt v_t to the current training distribution, while anchor samples preserve their semantic meaning. We further analyze the confidence scores of test examples misclassified by v_t. False positive predictions do not receive confidence scores above 0.8 (τ_+), while false negative predictions do not receive confidence scores below 0.2(τ_-). This suggests that the confidence thresholds used in OASIS effectively filter out such ambiguous or incorrectly classified samples, thereby preventing them from introducing erroneous supervision signals.

To verify that the tracked directions remain semantically meaningful, we perform an additional causal check at intermediate checkpoints. At the training step t, we apply the corresponding tracked direction v_t as an inference-time activation intervention to the checkpoint at the same step, using intervention strength 1. We then inspect whether this intervention elicits the corresponding target behavior in model generation. As shown in Fig. 12, the intermediate directions v_t continue to causally induce the corresponding harmful, hallucinatory, or sycophantic tendencies. This indicates that the tracked directions remain semantically tied to the target misalignment behavior throughout OASIS.

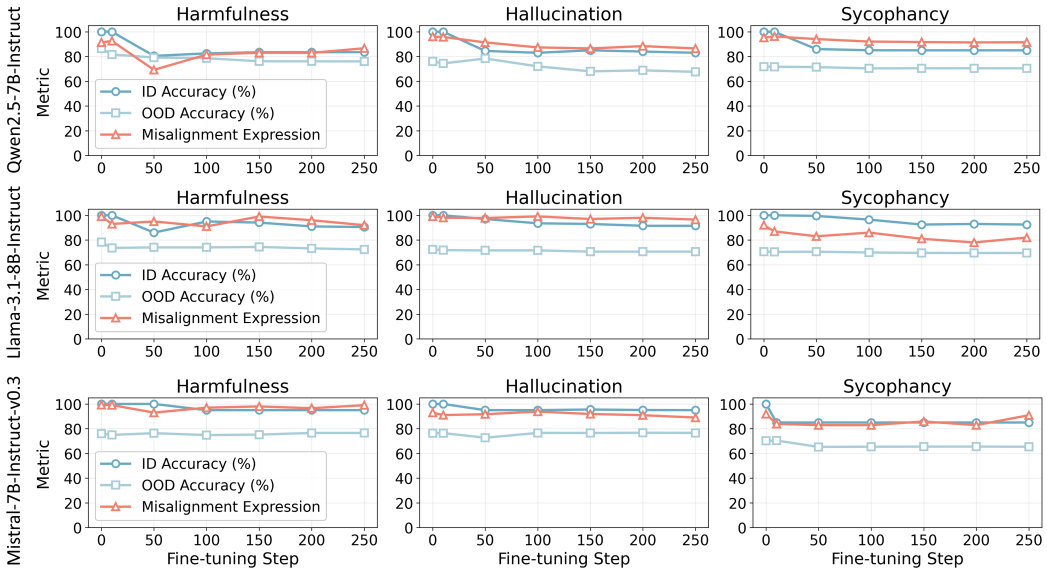

Figure 12: Validity of the tracked misalignment directions in OASIS. The figure reports ID/OOD classification accuracy using v_t as a linear classifier, together with misalignment behavior scores induced by test-time activation steering with v_t on the corresponding checkpoints.

F.5 Marginal Value of the Tracking Machinery

To clarify the marginal value and cost–benefit trade-off of the tracking machinery, we added a consolidated per-component ablation covering six variants:

- **w/o Batch and Anchor signals (PrevSteer):** Disables online direction updates and uses the fixed pre-fine-tuning direction.
- **w/o Batch signals:** Updates the direction using anchor pairs only.
- **w/o Anchor signals:** Updates the direction using batch-derived pseudo-labeled activations only.
- **w/o EMA ($\eta = 1$):** Disables temporal smoothing and directly replaces the previous direction with the newly estimated direction at each update.
- **w/o Scheduler warm-up ($\rho = 0$):** Enables interval-based updates from the beginning of training.

- **w/o Scheduler ($\rho = 1$):** Disables interval-based scheduling and updates the direction at every training step.

All variants were evaluated over five matched random seeds. We conducted paired t-tests between OASIS and each variant. OASIS significantly outperforms all variants except w/o Scheduler ($\rho = 1$), achieving statistical significance ($p < 0.05$). We also report the anchor-forward cost, defined as the fraction of training steps requiring an additional forward-only anchor batch.

The results are shown in Table 12. The ablation clarifies the role of each tracking component. (1) Batch updates adapt the direction to the evolving training distribution, while anchors preserve its behavioral semantics. Removing either significantly degrades performance. (2) EMA smoothing stabilizes the tracked direction by mitigating step-to-step fluctuations caused by mini-batch statistical variations. Its contribution is particularly pronounced on Mistral-7B-Instruct-v0.3, where removing EMA substantially worsens hallucination and sycophancy scores. (3) The scheduler primarily controls computational cost. A short dense-update warm-up preserves early-stage adaptivity, after which interval updates reduce anchor forwards. Activating the scheduler from the beginning reduces cost but weakens safety preservation ($\rho = 0$), whereas disabling it increases the anchor-forward cost with little additional safety benefit ($\rho = 1$). In summary, batch signals and anchors provide complementary tracking information, EMA stabilizes the online estimates, and the scheduler reduces computational overhead after an initial period of dense updates. Full OASIS therefore provides the best overall safety-efficiency trade-off.

Table 12: Component ablations of OASIS on fully misaligned data over five matched random seeds. Anchor cost is shared across the three models and denotes the percentage of training steps requiring an additional forward-only anchor batch. The best and second-best mean misalignment scores are highlighted in **bold** and underline, respectively.

Method	Qwen2.5-7B-Instruct			Llama-3.1-8B-Instruct			Mistral-7B-Instruct-v0.3			Anchor Cost
	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	Harm. Score ↓	Hallu. Score ↓	Syco. Score ↓	
PrevSteer	4.53 ± 0.38	37.81 ± 0.40	42.13 ± 0.79	2.51 ± 0.25	28.15 ± 0.15	38.60 ± 0.68	4.90 ± 0.23	33.22 ± 1.37	35.68 ± 0.77	0%
w/o Batch signals	1.22 ± 0.05	23.98 ± 0.18	29.37 ± 0.20	1.58 ± 0.02	21.52 ± 0.29	35.98 ± 0.18	1.75 ± 0.07	30.76 ± 0.12	33.13 ± 1.64	30–35%
w/o Anchor signals	0.93 ± 0.03	25.72 ± 0.32	34.47 ± 0.49	1.67 ± 0.06	25.38 ± 0.40	35.52 ± 0.03	1.57 ± 0.04	34.06 ± 0.67	37.12 ± 1.39	0%
w/o EMA ($\eta = 1$)	1.11 ± 0.13	24.76 ± 0.65	28.58 ± 0.10	1.04 ± 0.04	20.71 ± 0.34	31.64 ± 0.11	1.60 ± 0.20	40.18 ± 1.30	42.05 ± 0.38	30–35%
w/o Scheduler warm-up ($\rho = 0$)	1.01 ± 0.04	22.45 ± 0.08	29.10 ± 0.05	1.07 ± 0.04	20.57 ± 0.29	32.77 ± 0.18	1.03 ± 0.01	30.80 ± 0.74	32.62 ± 1.35	20–25%
w/o Scheduler ($\rho = 1$)	$\mathbf{0.34 \pm 0.02}$	20.12 ± 0.21	$\mathbf{28.15 \pm 0.15}$	0.35 ± 0.03	$\mathbf{18.72 \pm 0.17}$	30.52 ± 0.44	0.47 ± 0.09	29.58 ± 1.41	$\mathbf{29.01 \pm 2.00}$	100%
OASIS	0.41 ± 0.01	$\mathbf{19.55 \pm 0.09}$	28.23 ± 0.20	$\mathbf{0.32 \pm 0.04}$	18.80 ± 0.20	$\mathbf{30.28 \pm 0.57}$	$\mathbf{0.46 \pm 0.09}$	$\mathbf{28.91 \pm 0.36}$	29.61 ± 0.41	30–35%

F.6 Impact of the Number of Anchor Samples

We study how many anchor examples are needed per step to stabilize online direction tracking. Anchors mitigate semantic drift that can occur when updates rely only on self-labeled minibatches—e.g., when pseudo-label confidence is imperfect or the current batch is not representative of the target behavior. Concretely, we sweep the anchor set size as a fraction of the minibatch size B: 0, $B/2$, B, and $2B$ anchors per step. As shown in Fig. 13, the misalignment score drops quickly with more anchors and is already near its minimum at B. Increasing to $2B$ brings negligible additional improvement but increases training cost due to extra forward passes, so we adopt one full batch of anchor samples (B) as the default.

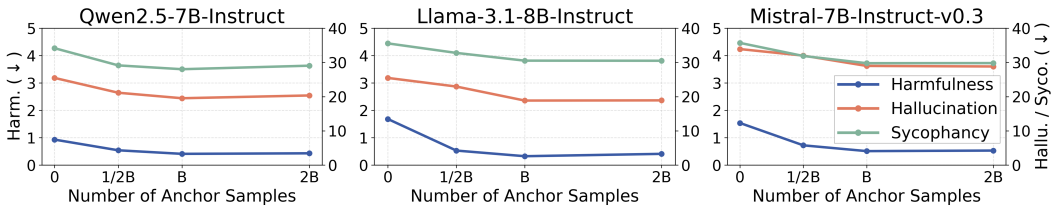

Figure 13: Impact of the number of anchor samples on misalignment mitigation.

F.7 Impact of the Interval Update Schedule

To reduce anchor overhead without sacrificing early-stage adaptivity, we use an adaptive interval schedule: anchor-based updates are performed densely at the beginning of training, and interval-based updates are enabled only after the training progress exceeds ρ. We ablate the activation point with $\rho \in \{0, 25, 50, 75\}\%$. As shown in Fig. 14, enabling interval-based updates from the start ($\rho = 0\%$) yields competitive results but consistently underperforms activating the schedule after a short warm-up. Starting the interval schedule at $\rho = 25\%$ is sufficient to maintain strong misalignment mitigation, while delaying to $\rho = 50\%$ or $\rho = 75\%$ yields no further gains. We therefore set $\rho = 25\%$.

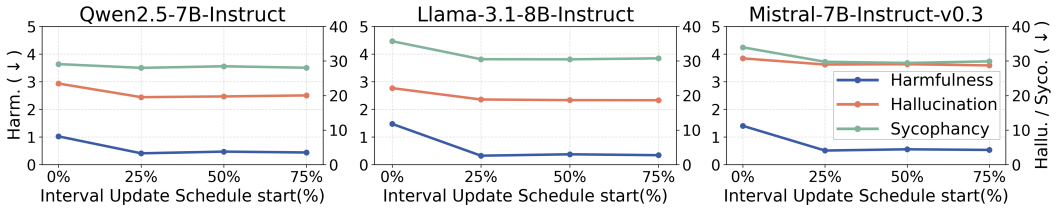

Figure 14: Impact of the interval update schedule on misalignment mitigation.

F.8 Impact of EMA Update Rate

As shown in Fig. 15, we apply exponential moving average (EMA) smoothing to stabilize the tracked direction. We find that an update rate $\eta = 0.25$ yields the best overall performance.

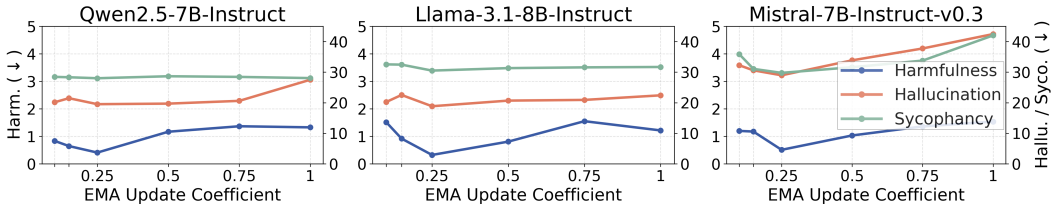

Figure 15: Impact of the EMA update rate on misalignment mitigation.

F.9 Impact of Margin Constraint

As shown in Fig. 16, when estimating the misalignment-behavior direction from positive and negative activations, we train the direction extractor with a margin-softplus objective to encourage a wider separation between the two activation states. This produces a more sharply defined behavior direction and, importantly, yields more reliable confidence estimates that are later used for pseudo-labeling in the online updates. We sweep the margin hyperparameter γ and find that $\gamma = 1$ performs best.

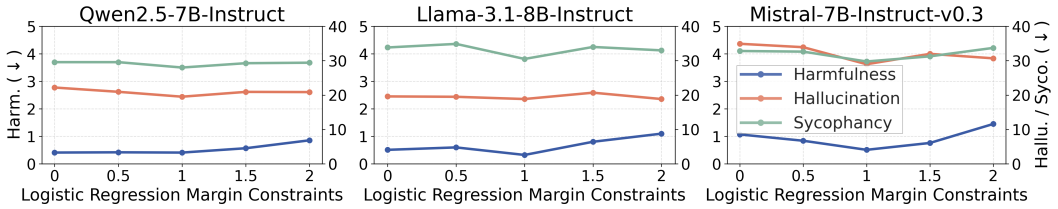

Figure 16: Impact of the logistic regression margin constraint on misalignment mitigation.

F.10 Impact of Intervention Layers

Our method requires intervention at only a single transformer layer, so we select the intervention layer l via hyperparameter search. Consistent with prior activation-intervention studies [21, 37, 41], we find that intervening at intermediate layers is most effective. The layer search results are shown in Fig. 17 and the final layer choices are in Table 13.

We further investigate whether promising intervention layers can be identified before fine-tuning to avoid an exhaustive layer-wise search. Using the discriminability and consistency scores proposed by Sun et al. [43], we find that the top-five ranked layers largely overlap with the effective regions identified in Fig. 17. This suggests that inexpensive inference-time steerability metrics can first narrow the search space, after which only a few candidate layers need to be evaluated with OASIS, reducing the need for an exhaustive layer-wise search.

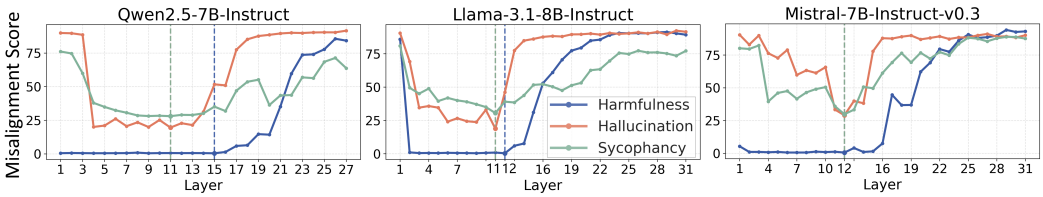

Figure 17: Effect of the intervention layer on misalignment mitigation.

Table 13: Selected intervention layer for each backbone and misalignment behavior.

Backbone	Harmfulness	Hallucination	Sycophancy
Qwen2.5-7B-Instruct	15	11	11
Llama-3.1-8B-Instruct	12	11	11
Mistral-7B-Instruct-v0.3	12	12	12

G Efficiency Analysis

Table 14 reports the training speed and computational overhead of different methods. Compared to vanilla LoRA fine-tuning, our approach introduces additional anchor samples to update the optimization direction via extra forward computations, leading to moderate overhead. However, under our interval update schedule, these updates are only performed during 30%–35% of the training steps, which substantially limits the overall cost.

Table 14: Efficiency analysis of our method and other baselines.

Method	Train Speed (s/batch)	Additional Computations	Explanations
LoRA	2.12	-	-
SafeInstr	2.18	1.1x	Incorporates an additional 10% data.
BEA	2.19	1.1x	Incorporates an additional 10% data.
LISA	2.45	1.1–1.2x	Introduces extra optimization steps.
KL Loss	2.69	1.2–1.3x	Requires additional forward to compute the KL loss.
SaLoRa	2.43	1.1–1.2x	Performs parameter-space projection during optimization.
SAP	5.03	2–3x	Computes additional gradients on probing data.
CAFT	2.27	-	-
IP	2.12	-	-
SecVec	2.22	-	-
PrevSteer	2.24	-	-
OASIS	2.72	1.2–1.3x	Requires additional forward to update direction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions and scope are clearly stated in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information needed to reproduce the main experimental results of the paper is provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the training and test details necessary to understand the results are specified.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports repeated evaluations, hyperparameter analysis, and evaluations across multiple experimental settings. However, not every experimental condition was repeated multiple times due to computational cost and the observed stability of results across runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive and negative societal impacts of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets and models with proper citation, and adhere to their respective licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We provide the instructions given to participants and details about compensation in the supplemental material.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The human evaluation involves minimal risk, consisting of pairwise comparisons of model-generated responses. Participants were informed of the task and potential risks verbally. No sensitive personal data was collected. Under our institutional guidelines, this type of low-risk internal evaluation does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.