RESISTANCE TRAINING TO REDUCE FORGETTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Task adaptation can improve a language model's target-task performance while weakening other important behaviors, even under on-policy training. We introduce Resistance Training (ResT), which reduces this forgetting through a temporary behavioral load. We fit a low-rank adapter to an unwanted behavior with the backbone frozen, then freeze the adapter while the backbone learns the task. Removing the adapter leaves a model with no additional inference cost. Across medical and financial adaptation of Qwen3-4B, targeted loads improve safety refusal or factual restraint over vanilla on-policy distillation and self-distillation while largely retaining task gains. Under MedQA on-policy distillation, ResT lowers the mean hallucination score by 40 points (0–100 scale), at comparable task accuracy. How the load is fitted also matters: on-policy distillation gives stronger hallucination protection than response supervision under both downstream distillation and supervised fine-tuning. This identifies resistance construction as a design choice for behavioral retention beyond the downstream objective. A linearized analysis gives conditions under which task learning compensates for the load, improving the protected behavior after removal.

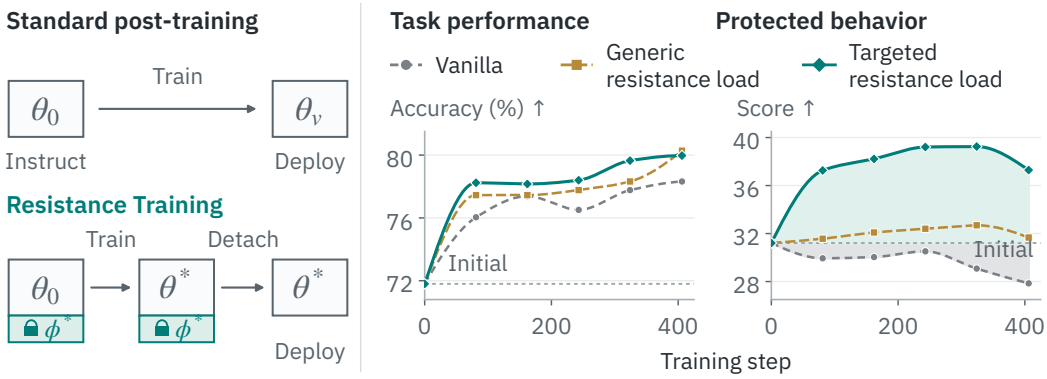


Figure 1: **Task adaptation with behavioral retention.** ResT trains with a frozen resistance adapter, removed at deployment (left). During MedQA OPD, targeted resistance improves both task accuracy and AIR-Bench safety refusal over vanilla training (right). Safety refusal receives no explicit supervision during MedQA training. Points show adapter-free evaluations; curves interpolate them. Dotted lines mark initial performance; rightmost shading shows refusal below this reference for vanilla (gray) and above it for targeted ResT (teal). Full results: Section 3.

1 INTRODUCTION

Domain-specific post-training adapts general-purpose language models to tasks such as medical question answering and financial reasoning. Learning is concentrated on the target task, but the deployed model must remain reliable beyond it. Benign fine-tuning can erode safety alignment (Qi et al., 2024), and learning new knowledge can increase hallucination (Gekhman et al., 2024). We study this *behavioral forgetting*: the loss of previously exhibited behaviors during ordinary task adaptation. Our focus is on refusal of harmful requests and restraint against unsupported claims, neither of which receives explicit supervision during downstream task training.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

On-policy learning offers a promising way to reduce forgetting relative to training on fixed demonstrations (Shenfeld et al., 2026b; Chen et al., 2026). On-policy distillation (OPD) supervises trajectories sampled by the student (Agarwal et al., 2024; Gu et al., 2024); on-policy self-distillation (OPSD) instead uses the model itself, conditioned on demonstrations, as the teacher (Zhao et al., 2026; Shenfeld et al., 2026a). Yet in our medical and financial adaptation experiments, both objectives improve task accuracy while weakening safety refusal and factual restraint (Tables 1 and 2). Supervision on sampled task trajectories need not preserve behaviors beyond the target task.

The behavioral state in which a model learns can affect what it retains. Shadow-FT (Wu et al., 2025) learns a task update on a paired base model and transfers it to the instruction model. Security vectors (Zhou et al., 2024) instead fit a removable adapter to an unwanted behavior and freeze it during subsequent fine-tuning; preventive persona steering applies a related intervention in activation space (Chen et al., 2025). We study how to construct such a temporary state for behavioral retention during on-policy task adaptation, where the load can also change the trajectories on which the model learns.

The model's learning state need not be its deployment state.

Can training against failure help prevent forgetting?

Resistance Training (ResT) implements this intervention with a removable low-rank adapter (Figure 2). First, we fit the backbone–adapter combination to an unwanted behavior, updating only the adapter. We then freeze the adapter and train the backbone on downstream task data. The adapter stays active during training and rollout generation, then is removed for deployment. It acts as a fixed *resistance load*: when its behavioral shift interferes with task learning, optimization can learn a compensating update that improves the protected behavior after removal. The downstream objective is unchanged, and inference requires no adapter.

Our central design choice is *resistance construction*: which unwanted behavior the load expresses and how it is fitted. Its source can be a paired base model, examples of a specific failure, or checkpoints degraded by ordinary task training. We compare fitting fixed responses with distilling a resistance teacher along the adapter-conditioned student's own trajectories. This on-policy fitting updates only the adapter; downstream task learning updates only the backbone. The source data are used to construct the load, and retention is measured on held-out behavior benchmarks separate from the task.

Our experiments use Qwen3-4B in thinking mode for MedQA and FinQA adaptation, with safety refusal and factual restraint studied in separate protection settings. Targeted resistance improves the protected metric over vanilla OPD and OPSD in every main comparison while largely retaining task gains; several settings also improve both metrics over the initial model (Tables 1 and 2). The fitting comparison reveals a further result: on-policy-fitted loads give stronger hallucination protection than response-supervised loads even when downstream learning uses SFT. The benefit of on-policy resistance fitting therefore extends beyond on-policy task learning (Figure 3).

Our analysis identifies conditions for protection after load removal in a fixed-feature linearized model (Section 2.4). Protection requires task features to expose the resistance direction and the learned shift to transfer to held-out protection prompts. The analysis explains how task learning can compensate for an unwanted behavior without explicit supervision of the protected behavior. On-policy training can additionally change the trajectories on which this compensation is learned.

Our contributions are:

- We document behavioral forgetting during medical and financial adaptation under on-policy distillation and self-distillation, despite gains in task accuracy.
- We study resistance construction through behavioral sources and fitting objectives, finding stronger hallucination protection from on-policy-fitted loads under both downstream OPD and SFT, and joint task–retention gains in several settings.
- We provide a local account of how a frozen adapter changes the backbone update and a linearized analysis of the conditions for protection after removal.

2 RESISTANCE TRAINING

Our MedQA and FinQA experiments show that task learning can weaken behaviors without explicit downstream supervision: vanilla OPD and OPSD improve task accuracy while increasing hallucination and reducing harmful-request refusal (Tables 1 and 2). ResT addresses this problem by changing the model's behavioral state during task learning through a temporary resistance load.

2.1 METHOD OVERVIEW

Let θ_0 denote the initial instruction backbone, D_{task} the downstream task data, and $p_{\theta,\phi}$ the model with backbone parameters θ and low-rank adapter parameters ϕ. The adapter supplies a *resistance load*: a behavioral shift that remains present while the backbone learns the task. We first fit ϕ with the backbone fixed at θ_0, then freeze the fitted adapter ϕ^* and update the backbone on D_{task}. Finally, we remove the adapter and deploy $p_{\theta^*,0}$. Figure 2 illustrates this fit–train–remove procedure. Appendix A.1 compares its shared structure and construction choices with prior training-time interventions.

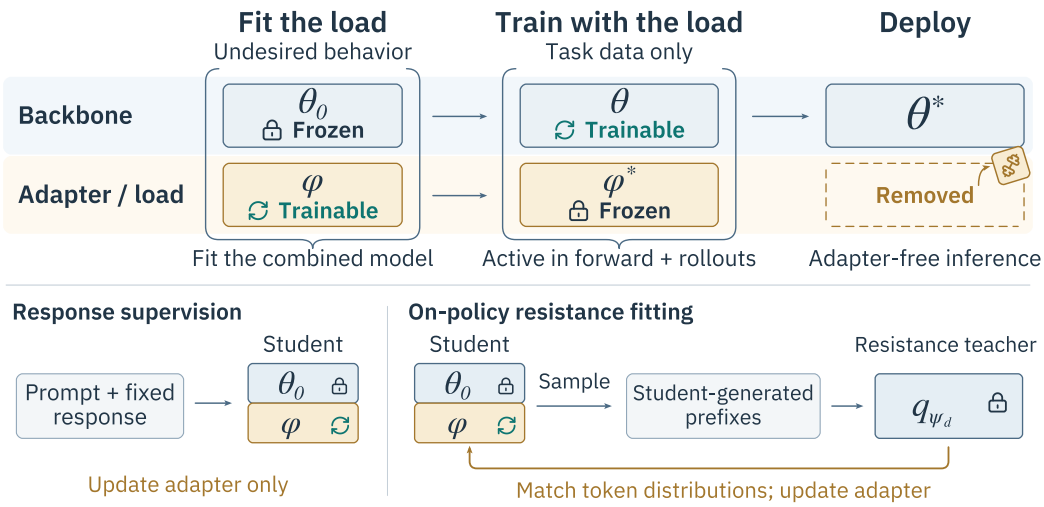


Figure 2: **Resistance Training and on-policy load fitting.** *Top:* Fit the load with the backbone frozen, train the backbone with the load frozen, then remove the load for deployment. During task learning, the load remains active in forward passes and on-policy rollouts. *Bottom:* Response supervision uses fixed demonstrations. On-policy fitting samples trajectories from the adapter-conditioned student and matches a frozen resistance teacher at the same prefixes, updating only the adapter. As the adapter changes, new student trajectories are sampled. This resistance teacher is distinct from the downstream task teacher.

2.2 FITTING THE RESISTANCE LOAD

We choose both the behavior encoded by the load and the trajectories on which it is learned. Response supervision uses fixed demonstrations; on-policy fitting follows the prefixes visited by the evolving adapter-conditioned student.

Behavioral source. A *generic resistance load* approximates the behavior of the paired pre-instruction model. A *targeted resistance load* instead encodes a specific failure: harmful compliance from HarmChain responses or confident unsupported claims from M2. The responses can supervise the adapter directly or train a resistance teacher. We also study *forgetting-derived resistance loads* from checkpoints degraded by ordinary task training (Section 3.4). All sources are constructed separately from the final evaluation examples; Appendix B gives their provenance.

We fit the combined model $p_{\theta_0,\phi}$ to the chosen behavior while updating only ϕ. For the generic and targeted sources, we compare *response supervision* with *on-policy distillation*, using the same low-rank adapter. The fitting objective is independent of the downstream training algorithm.

Response supervision. We use resistance-source prompts paired with responses, $(x, y_r) \sim D_{\text{deal}}$, and minimize the cross-entropy loss

$$\phi^*_{\text{CE}} = \arg\min_{\phi} \mathbb{E}_{(x,y_r)\sim D_{\text{deal}}}[-\log p_{\theta_0,\phi}(y_r \mid x)]. \tag{1}$$

On-policy resistance fitting. A frozen resistance teacher q_{ψ_d} with the backbone's architecture supplies token-level supervision. For each prompt x from D_{deal}, the student $p_{\theta_0,\phi}$ samples a trajectory $\tilde{y}$. We match the teacher at these student-generated prefixes by minimizing

$$\phi^*_{\text{OPD}} = \arg\min_{\phi} \mathbb{E}_{x\sim D_{\text{deal}},\, \tilde{y}\sim p_{\theta_0,\phi}(\cdot|x)} \left[\sum_t D_{\text{KL}}(p_{\theta_0,\phi}(\cdot \mid x, \tilde{y}_{<t}) \,\|\, q_{\psi_d}(\cdot \mid x, \tilde{y}_{<t})) \right]. \tag{2}$$

The teacher supplies the unwanted behavioral signal; the student determines the prefixes at which it is learned. Each adapter update changes the distribution used for subsequent sampling, closing the fitting loop in Figure 2. Only ϕ is updated; the backbone and resistance teacher remain frozen. This teacher is used only to fit the load and is distinct from the downstream task teacher. Section 3.3 compares the resulting protection under both downstream OPD and SFT.

2.3 Learning the task with the load

Let $\mathcal{L}_{\text{task}}$ be the downstream objective. It can be a supervised or distillation loss, a preference objective such as DPO, or a reward-based objective such as GRPO. Once the load is fitted, we optimize this objective over the backbone while keeping the adapter fixed:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{\text{task}}(p_{\theta,\phi^*}), \qquad \phi_t \equiv \phi^*. \tag{3}$$

For on-policy training, both rollout generation and loss computation use the active model p_{θ,ϕ^*}. The load therefore affects the trajectories on which the backbone learns as well as its update on each trajectory. After training, we remove ϕ^* and evaluate the updated backbone $p_{\theta^*,0}$. The task objective is unchanged, and the deployed model carries no adapter.

A local view of update conditioning. For a fixed training batch, let L be a twice continuously differentiable task loss and let δ_ϕ denote the frozen adapter's additive displacement in effective weight space. At the same backbone weights θ, the conditioned and unconditioned gradients satisfy

$$g_\phi - g_0 = \nabla L(\theta + \delta_\phi) - \nabla L(\theta) = H_L(\theta)\delta_\phi + o(\|\delta_\phi\|). \tag{4}$$

The frozen adapter changes the backbone's gradient through the task loss's local curvature. After one gradient-descent step of size η, the difference between the adapter-free backbones is $-\eta(g_\phi - g_0)$. This identity isolates the update on a fixed batch; for on-policy objectives, the adapter also changes which trajectories are sampled. We next examine when the update itself can improve the protected behavior after removal.

2.4 Protection after load removal

Consider a fixed-feature binary linearization of a protected behavior, with one desirable and one undesirable outcome. If task learning compensates for the load's undesirable shift, removing the load leaves that compensation in the backbone. The following result identifies conditions under which this improves the protected behavior relative to vanilla training.

Theorem 2.1 (Behavioral gain from compensated resistance)**.** *Consider matched vanilla and resistance-trained runs under a fixed-feature binary linearization and a common local quadratic model of SFT, OPD, or OPSD on D_{task}. Suppose a fixed adapter fitted by response supervision or on-policy distillation consistently moves the active model toward the undesirable side of a protected behavior. Under gradient flow from the same initialization to convergence, the adapter-free backbone differs from vanilla by the part of the resistance displacement exposed by target-task features. It strictly increases the desirable-behavior probability on held-out protection prompts whenever the adapter's smallest source-domain displacement is larger than the combined error from target-task feature non-coverage and transfer from resistance-source prompts to the held-out prompts.*

Appendix C gives the formal statement and proof. The result connects protection to two forms of coverage: task prompts must expose the feature directions through which the load acts, and the resistance-source prompts must support transfer to held-out protection prompts. This allows task-only training to compensate for a behavioral shift without replaying its source data. For adapters fitted by on-policy distillation, the resistance margin depends on how consistently the teacher exhibits the failure and how accurately the adapter reproduces it; for response supervision, it depends on how uniformly the undesirable responses are fitted. Protection thus depends on the fitted behavioral direction and its coverage by the task. The analysis holds trajectories fixed, isolating this compensation from the additional effects of on-policy sampling.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

We test whether ResT preserves behavior during task adaptation while retaining task gains. The main comparisons evaluate this trade-off under on-policy distillation and self-distillation; the ablations examine how adapter fitting and the source of resistance affect protection. Full experimental details are in Appendix D.

Tasks and evaluation. We adapt Qwen3-4B to medical question answering (MedQA) and financial reasoning (FinQA). These domains represent practical uses of benign fine-tuning to specialize a general-purpose model, allowing us to study behavioral forgetting that can arise naturally during task adaptation. The model generates reasoning traces before its final answer during both training and evaluation. Our main experiments use OPD and OPSD, with SFT results in Appendix E.

Safety refusal and factual restraint serve as two examples of behavior outside the downstream task supervision. We measure refusal of harmful requests on AIR-Bench 2024 (Zeng et al., 2025) and hallucination using the mean persona-based score (0–100; lower is better) over 20 held-out questions with ten responses each (Chen et al., 2025). We call these the Safety and Hallucination settings, respectively. Each is a separate protection experiment: we report the protected metric together with task accuracy from the same checkpoint, evaluated after removing the resistance adapter.

Comparison methods. The initial instruction model (Initial model in the tables) provides the reference for retention; vanilla training measures the effect of adding resistance. The initial model is distinct from Qwen3-4B-Base, the paired pre-instruction model used to construct generic loads.

We also compare three approaches to retention. Shadow-FT learns the task update on the paired base model and transfers it to the instruction model (Wu et al., 2025). OSFT restricts updates to the complement of protected weight subspaces (Nayak et al., 2026). Replay (Huang et al., 2024a) augments task training with prompts from the resistance-source pools and uses the native downstream objective, without using the unwanted responses. Appendix D.3 describes their adaptation to on-policy training and the training costs.

Resistance loads and ablations. The main experiments examine two ways to specify the unwanted behavior encoded by a resistance load. *Generic* loads use the paired pre-instruction Qwen3-4B-Base model as a broad proxy for behavioral regression; *targeted* loads encode specified failures, here harmful compliance or confident unsupported claims. Both are fitted by on-policy distillation. Sources and fitting schedules are detailed in Appendix D.4.

The ablations examine the fitting objective and extend the choice of behavioral source. First, we compare response supervision with on-policy distillation for generic and targeted loads under downstream OPD and SFT (Section 3.3). We then introduce *forgetting-derived loads*: adapters distilled from checkpoints degraded by ordinary task training. These use naturally occurring regression to specify the unwanted behavior, providing another source to compare with generic and targeted loads (Section 3.4).

The tables report scores from individual training runs. Repeated evaluation seeds measure decoding variability; aggregation is specified in Appendix D.

Table 1: **Downstream OPD on MedQA and FinQA.** Each protection setting is a separate run, evaluated without the adapter. Task accuracy and AIR-Bench refusal are percentages; hallucination is a mean score on a 0–100 scale. Arrows indicate better performance. Within each domain and column, bold marks the best score; muted red marks post-training scores worse than vanilla. ResT uses loads fitted by on-policy distillation (Appendix D.4).

	Safety refusal		Factual restraint	
Method	Task accuracy↑	Refusal↑	Task accuracy↑	Hallucination↓
		Medical (MedQA)		
Initial model	71.80	31.21	71.80	34.00
Vanilla	79.36	28.83	79.36	55.99
Shadow-FT‡	79.26	28.08	79.26	40.68
OSFT	79.65	27.44	78.08	53.97
Replay	79.73	**38.20**	77.85	35.08
ResT: Generic	79.65	30.28	**80.28**	24.55
ResT: Targeted	**80.13**	37.32	79.50	**15.96**
		Finance (FinQA)		
Initial model	70.18	31.21	70.18	34.00
Vanilla	71.93	25.74	71.93	46.36
Shadow-FT‡	**73.58**	27.26	**73.58**	27.84
OSFT	70.44	26.03	71.23	48.95
Replay	71.23	29.71	71.05	34.23
ResT: Generic	72.89	27.85	72.19	32.11
ResT: Targeted	71.58	**32.14**	72.54	**9.93**

‡Most Shadow-FT generations under OPD reach the length limit (Appendix D.3).

3.2 Main results

Targeted resistance improves the protected metric over vanilla training in all eight main settings (Tables 1 and 2). In six, task accuracy also improves over vanilla; in the other two, it falls by 0.35 and 1.10 points. Relative to the initial model, seven settings improve both task accuracy and the protected metric. Under MedQA OPD, for example, the targeted hallucination load reduces the hallucination score by 40 points relative to vanilla training, at comparable task accuracy. Targeted loads also improve retention under SFT, with task–retention trade-offs varying by domain and fitting objective (Appendix E).

Targeted loads provide more consistent protection. Targeted loads outperform generic loads on the protected metric in every main comparison and achieve the lowest hallucination scores among all methods in both domains and under both objectives. Generic loads also reduce hallucination relative to vanilla training, but their refusal gains are less consistent. Replay is a strong safety baseline: it achieves higher refusal than targeted ResT in three of four comparisons, while targeted ResT has higher task accuracy in all four.

> **Takeaway 1.** *Targeted resistance preserves behavior while retaining task gains.* It improves the protected metric over vanilla training in all eight main settings; in seven, task accuracy and the protected behavior both exceed the initial model.

3.3 Ablation: fitting the resistance load

We compare response supervision and on-policy distillation to test how load fitting affects retention under downstream OPD and SFT (Figure 3). For each source, fitting objective, and downstream objective, we select the configuration with the highest MedQA accuracy in the original fitting sweep. This evaluates each procedure at its best reported task performance; compute and induced behavioral

Table 2: **Downstream OPSD on MedQA and FinQA.** Metrics, separate protection settings, and notation follow Table 1. ResT uses loads fitted by on-policy distillation; fitting schedules, including the generic Safety exception, are given in Appendix D.4.

	Safety refusal		Factual restraint	
Method	Task accuracy↑	Refusal↑	Task accuracy↑	Hallucination↓
		Medical (MedQA)		
Initial model	71.80	31.21	71.80	34.00
Vanilla	75.81	23.91	75.81	52.52
Shadow-FT	72.11	28.46	72.11	40.66
OSFT	75.41	21.30	76.51	53.34
Replay	76.28	**36.02**	74.78	30.59
ResT: Generic	75.65	26.89	**77.45**	36.62
ResT: Targeted	**77.93**	34.82	74.71	**13.60**
		Finance (FinQA)		
Initial model	70.18	31.21	70.18	34.00
Vanilla	73.23	22.48	73.23	47.25
Shadow-FT	71.58	30.33	71.58	36.71
OSFT	71.49	21.02	71.05	46.21
Replay	73.32	**34.10**	70.88	26.09
ResT: Generic	74.02	21.19	72.10	45.87
ResT: Targeted	**74.37**	26.81	**74.54**	**16.13**

strength can differ. Appendix D.4 gives the selection rule, duration sweeps, and an additional sequence-matched comparison. These selected configurations differ from the fixed fitting schedules of the main experiments.

On-policy fitting consistently improves hallucination protection. Across both sources and both downstream objectives, on-policy fitting gives lower hallucination scores than response supervision. All four selected models fall below the initial model's score of 34.00; none of their response-supervised counterparts does (39.60–43.73). The largest gap occurs with the targeted load under downstream SFT: the hallucination score is 26.0 points lower (17.73 versus 43.73), while MedQA accuracy is 0.55 points higher. On-policy fitting also gives higher MedQA accuracy in three of the four comparisons.

Refusal protection depends on the source. With targeted loads, on-policy fitting improves refusal under both downstream objectives. Under OPD it improves task accuracy as well; under SFT the refusal gain costs 1.26 MedQA points. With generic loads, the fitting objectives give similar results: refusal differs by at most 0.21 points and task accuracy by at most 0.32 points. Across all eight comparisons, on-policy fitting improves task accuracy in five. Stronger protection therefore often accompanies better task learning.

The hallucination advantage therefore persists when subsequent task learning uses fixed demonstrations. It also appears across the recorded SFT fitting durations: at every shared duration, on-policy fitting gives lower hallucination than response supervision for both sources (Appendix D.4). These comparisons evaluate the complete fitting procedure, including its student-generated prefixes, teacher distributions, and fitting budget.

> **Takeaway 2.** *How the load is fitted affects protection.* On-policy fitting improves hallucination protection for both sources and refusal protection for targeted loads, under both downstream OPD and SFT.

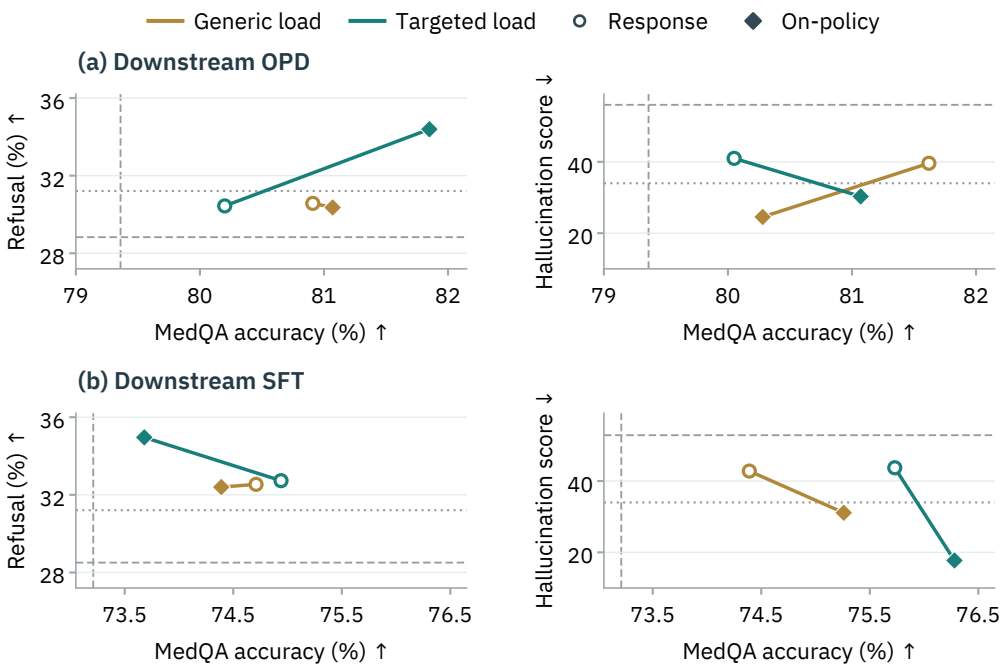


Figure 3: **Load fitting changes the task–retention trade-off.** Each point pairs task accuracy with the protected score of one adapter-free model. Lines connect response supervision (open circles) to on-policy fitting (filled diamonds) for the same source; configurations are selected for their highest reported MedQA accuracy. Vertical and horizontal dashed lines mark vanilla task and behavior scores; dotted horizontal lines mark initial behavior. On-policy fitting lowers hallucination under both downstream objectives. Selection and fitting schedules: Appendix D.4.

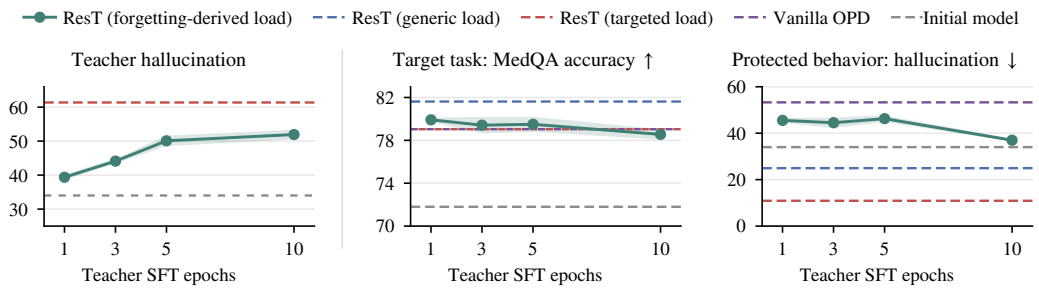
3.4 Ablation: resistance from natural forgetting

Beyond the generic and targeted sources in the main experiments, can ordinary task forgetting itself supply a useful resistance load? We fit adapters by on-policy distillation from MedQA-SFT checkpoints trained for 1, 3, 5, and 10 epochs. These teachers become more prone to hallucination as training proceeds, providing *forgetting-derived resistance loads*. Teacher-training epochs index the stages of forgetting; they do not count adapter fitting or downstream training.

We then train on MedQA with downstream OPD and evaluate after removing each adapter (Figure 4). Generic and targeted loads provide two alternative sources. This sweep is a single-seed run separate from Table 1: each configuration is trained once and evaluated with three generation seeds, so its reference scores differ from the main table (Appendix D).

Natural forgetting supplies useful resistance. Every forgetting-derived load lowers hallucination relative to vanilla OPD. Checkpoints from epochs 1–5 give similar deployed scores (44.51–46.26 versus 53.28 for vanilla), despite increasing teacher hallucination. The epoch-10 load gives a lower score of 36.98, but this additional gain comes from a single training run. Longer teacher training therefore does not reliably translate into stronger protection.

Generic and targeted sources provide stronger protection. Both alternatives outperform every forgetting-derived load: the generic load yields a hallucination score of 24.94 and the targeted load 10.89, compared with 34.00 for the initial model. The targeted teacher also has the highest measured hallucination score, although the sources differ in both the behavior they express and its severity. Thus, a generic base-model load provides stronger protection than the naturally forgotten checkpoints studied here, while targeting the unwanted behavior gives the largest gain.



(a) Resistance teacher. (b) Task accuracy after training. (c) Hallucination after training.

Figure 4: **Resistance from natural forgetting and alternative sources.** The horizontal axis indexes MedQA-SFT epochs of the resistance teacher. Panel (a) shows teacher hallucination; panels (b–c) show task accuracy and hallucination after downstream OPD and adapter removal. Teal denotes forgetting-derived loads, blue generic, and red targeted; the generic teacher is omitted from (a). Gray and purple references mark the initial model and vanilla OPD. Nearly overlapping vanilla and targeted curves in (b) share an alternating line. Points and bands show means and one standard deviation over three evaluation seeds of one trained model.

The protection gains retain task learning. The forgetting-derived models reach 78.55–79.92 MedQA accuracy, close to vanilla OPD (79.08) and above the initial model (71.80). The targeted load retains comparable accuracy (79.00), while the generic load improves it to 81.62. Thus, the large differences in retention across sources do not require giving up the task gains.

> **Takeaway 3.** *Natural forgetting supplies resistance, but longer training is not a reliable guide to protection.* Generic and targeted loads preserve more of the protected behavior in this comparison while retaining task gains.

Together, the ablations identify two useful design choices: the behavioral source and how the adapter is fitted. Naturally degraded checkpoints can supply resistance, but neither teacher-training duration nor the source label alone determines protection. The relevant object is the behavioral state produced by the fitted load, not simply the amount of training used to obtain it.

The linearized analysis offers a complementary interpretation: a load is useful when downstream task features expose its behavioral direction and the resulting update transfers to the held-out behavior. This condition concerns the fitted displacement, rather than the source model's score or an epoch count. On-policy fitting changes how that displacement is learned, while adapter-active task sampling changes where it is used. Separating these effects is a next step toward selecting loads before downstream training.

4 Conclusion

We introduced Resistance Training (ResT) for reducing behavioral forgetting during task adaptation. A temporary adapter expresses an unwanted behavior while the backbone learns the task, then is removed for deployment. Targeted loads improve the protected metric over vanilla OPD and OPSD in all eight main settings while largely retaining task gains. How the load is fitted affects retention: on-policy fitting gives stronger hallucination protection under both downstream OPD and SFT. Resistance construction is therefore a useful design choice beyond the downstream learning objective.

Our linearized analysis explains when task learning compensates for the load. The experiments establish this approach on Qwen3-4B for safety refusal and factual restraint. Broader models, independent factuality evaluations, and extensions to DPO and GRPO are natural next tests; separating the roles of trajectories, teacher supervision, and backbone updates can guide load design. Together, the results show that behavioral retention depends on more than the downstream objective: choosing the state in which a model learns can improve what it retains after that state is removed.

AI USE STATEMENT

In this work, we used generative AI tools to assist with parts of the theoretical derivation, to help implement and launch the training runs, to provide feedback on the experimental design, to generate synthetic data sets, to clean and filter data, to analyze data, and to assist in interpreting experimental results. For the auxiliary linear interpretation (Section 2.4 and Appendix C), AI tools helped derive intermediate steps of the proof; the authors verified every step and take responsibility for the stated results. The experimental design was determined by the authors, with AI tools providing occasional feedback. Within the experimental pipeline, Qwen3-32B generated the downstream medical and financial reasoning traces, and GPT-4.1 and Qwen3.6-27B filtered them for correctness and answer leakage (Appendix D); an open-weight model with its refusal behavior removed generated the misaligned responses used to fit the targeted Hallucination adapters (targeted Safety adapters use the native HarmChain responses), and Qwen3-4B-Base served as the generic resistance teacher and generated the base-model responses used for response supervision (Appendix B); Qwen3.6-27B and GPT-4.1-mini scored the evaluation outputs (Appendix D). We have not used generative AI tools to formulate the theoretical framework or mathematical claims, including the theorem statements and their assumptions, or to propose research hypotheses, and translation and qualitative analysis are not applicable to this work. Additionally, we used generative AI tools to write plotting code and produce figures from our results, to check reported numbers against the raw results, to search for and verify related work, and to draft and edit parts of the paper for clarity. We have reviewed all AI-assisted work, and the authors take full responsibility for the content of this paper.

ETHICS STATEMENT

This work aims to preserve safety refusal and factual restraint during post-training, but parts of its pipeline involve harmful content. The Safety study fits adapters on harmful requests paired with harmful reasoning and responses from HarmChain, and the Hallucination study generates confidently unsupported answers with a model whose refusal behavior has been removed. The resulting resistance adapters encode the failures they are meant to counter and, used on their own, would make a model less safe or less factual. In ResT they serve only as frozen training-time components and are removed before evaluation; no reported model carries an adapter. We will not publicly release the targeted adapters or teachers. Harmful prompts from AIR-Bench 2024 are used only to measure refusal. The study involves no human subjects and collects no personal data; all datasets and models are publicly available and are used under their respective licenses (Appendix D.6). The medical and financial models are research artifacts and are not intended for clinical or financial decision making. Finally, improvements on our benchmarks do not certify that a deployed model is safe or factual.

REPRODUCIBILITY STATEMENT

Appendix D specifies the models, frameworks, and hardware; the construction of the downstream data; the training configuration of every objective (Table 3); the training of the resistance adapters, including the epoch grid and the selection rule of Section 3.3; the baselines; the evaluation protocols and judge models; and the training and evaluation seeds. Appendix B describes how each resistance-source data set is built, including sources, filters, and generation settings, and Appendix E reports the corresponding settings under SFT. The assumptions and complete proof of the linear interpretation are given in Appendix C. Training-run counts and repeated-evaluation summaries are specified in Appendix D. We will release the code, configuration files, and evaluation prompts.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

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 *Advances in Neural*

Information Processing Systems (NeurIPS), 2024. doi: 10.52202/079017-4322. URL https://openreview.net/forum?id=pH3XAQME6c.

Jan Betley, Daniel Chee Hian 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 *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4043–4068. PMLR, 2025. URL https://proceedings.mlr.press/v267/betley25a.html.

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 *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=gT5hALch9z.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Helena Casademunt, Caden Juang, Adam Karvonen, Samuel Marks, Senthooran Rajamanoharan, and Neel Nanda. Steering out-of-distribution generalization with concept ablation fine-tuning. *arXiv preprint arXiv:2507.16795*, 2025.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64375.

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

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3697–3711. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4069–4082. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1418. URL https://aclanthology.org/D19-1418/.

Jan Dubiński, Jan Betley, Anna Sztyber-Betley, Daniel Tan, and Owain Evans. Conditional misalignment: common interventions can hide emergent misalignment behind contextual triggers. *arXiv preprint arXiv:2604.25891*, 2026. URL https://arxiv.org/abs/2604.25891.

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*, 2024. URL https://aclanthology.org/2024.emnlp-main.444/.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Seokil Ham, Jaehyuk Jang, Wonjun Lee, and Changick Kim. Jailbreak to protect: Buffering and reinforcing via temporary jailbreaking for safe fine-tuning in large language models. In *International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://arxiv.org/abs/2605.24550.

He He, Sheng Zha, and Haohan Wang. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pp. 132–142. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-6115. URL https://aclanthology.org/D19-6115/.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. NIPS 2014 Deep Learning Workshop.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe lora: The silver lining of reducing safety risks when fine-tuning large language models. In *Advances in Neural Information Processing Systems*, 2024.

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

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1416–1428, 2024a. URL https://aclanthology.org/2024.acl-long.77/.

Tiansheng Huang, Sihao Hu, and Ling Liu. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b. doi: 10.52202/079017-2356. URL https://openreview.net/forum?id=lpXDZKiAnt.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://iclr.cc/virtual/2025/oral/31754.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://iclr.cc/virtual/2023/poster/12254.

Wonje Jeung, Sangyeon Yoon, Minsuk Kahng, and Albert No. SAFEPATH: Preventing harmful reasoning in chain-of-thought via early alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=vIaNnnQxcl.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. BeaverTails: Towards improved safety alignment of LLM via a human-preference dataset. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4dbb61cb68671edc4ca3712d70083b9f-Abstract-Datasets_and_Benchmarks.html.

Fengqing Jiang, Zhangchen Xu, Yuetai Li, Luyao Niu, Zhen Xiang, Bo Li, Bill Yuchen Lin, and Radha Poovendran. SafeChain: Safety of language models with long chain-of-thought reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 23303–23320. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1197. URL https://aclanthology.org/2025.findings-acl.1197/.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=VrHiF2hsrm.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*. ACM, 2023. doi: 10.1145/3600006.3613165.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2773081.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. Mitigating the alignment tax of RLHF. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 580–606. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.35. URL https://aclanthology.org/2024.emnlp-main.35/.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6691–6706. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.522. URL https://aclanthology.org/2021.acl-long.522/.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33, 2025. URL https://ieeexplore.ieee.org/document/11151751/.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/eddc3427c5d77843c2253f1e799fe933-Abstract.html.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Oleg Silkin, Mustafa Eyceoz, Hao Wang, Aldo Pareja, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in LLMs for continual learning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=vQcyqsGJDw.

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 *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=hTEGyKf0dZ.

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

Lingfei Qian, Weipeng Zhou, Yan Wang, Xueqing Peng, Jimin Huang, and Qianqian Xie. Fino1: On the transferability of reasoning enhanced LLMs to finance. *arXiv preprint arXiv:2502.08127*, 2025. Version 1.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Maxime Riché, Daniel Tan, Vili Kohonen, and Niels Warncke. Inoculation adapters: Improved selective generalization of capabilities with fewer surprising backdoors. *arXiv preprint arXiv:2606.30252*, 2026. URL https://arxiv.org/abs/2606.30252.

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)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising: A defence mechanism against harmful finetuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-0402. URL https://openreview.net/forum?id=eP9auEJqFg.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=Hf3qXoiNkR.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, 2026a.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations (ICLR)*, 2026b. URL https://openreview.net/forum?id=GO0X9nQot3.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*. ACM, 2025. doi: 10.1145/3689031.3696075.

Linxin Song, Taiwei Shi, and Jieyu Zhao. The hallucination tax of reinforcement finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2105–2120. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.112. URL https://aclanthology.org/2025.findings-emnlp.112/.

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://iclr.cc/virtual/2025/poster/31026.

Daniel Tan, Anders Cairns Woodruff, Niels Warncke, Arun Jose, Maxime Nicolas Riché, David Demitri Africa, and Mia Taylor. Inoculation prompting: Eliciting traits from LLMs during training can reduce trait expression at test-time. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=FiRBNBdaZy.

Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Yixuan Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. BackdoorAlign: Mitigating fine-tuning based jailbreak attack with backdoor enhanced safety alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. doi: 10.52202/079017-0169. URL https://openreview.net/forum?id=1PcJ5Evta7.

Miles Wang, Tom Dupré la Tour, Olivia Watkins, Aleksandar Makelov, Ryan Andrew Chi, Samuel Miserendino, Jeffrey George Wang, Achyuta Rajaram, Johannes Heidecke, Tejal Patwardhan, and Daniel P. Mossing. Persona features control emergent misalignment. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://mlanthology.org/iclr/2026/wang2026iclr-persona/.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

Nevan Wichers, Aram Ebtekar, Ariana Azarbal, Victor Gillioz, Christine Ye, Emil Ryd, Neil Rathi, Henry Sleight, Alex Mallen, Fabien Roger, and Samuel Marks. Inoculation prompting: Instructing LLMs to misbehave at train-time improves test-time alignment. *arXiv preprint arXiv:2510.05024*, 2025.

Taiqiang Wu, Runming Yang, Jiayi Li, Pengfei Hu, Yik-Chung Wu, Ngai Wong, and Yujiu Yang. Shadow-ft: Tuning instruct model via training on paired base model. *arXiv preprint arXiv:2505.12716*, 2025.

Xenomirant. Evaluating task vectors, unlearning and inoculation, September 2026. URL https://www.lesswrong.com/posts/8AYQrvD4HEh8MEjgR/evaluating-unlearning-and-inoculation-task-vector. Research blog post, September 20, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yi Zeng, Yu Yang, Andy Zhou, Jeffrey Ziwei Tan, Yuheng Tu, Yifan Mai, Kevin Klyman, Minzhou Pan, Ruoxi Jia, Dawn Song, Percy Liang, and Bo Li. AIR-Bench 2024: A safety benchmark based on regulation and policies specified risk categories. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=UVnD9Ze6mF.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/64784.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 29733–29735, 2025. doi: 10.1609/aaai.v39i28.35383.

Xin Zhou, Yi Lu, Ruotian Ma, Yujian Wei, Tao Gui, Qi Zhang, and Xuanjing Huang. Making harmful behaviors unlearnable for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10258–10273, 2024. URL https://aclanthology.org/2024.findings-acl.611/.

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. *arXiv preprint arXiv:2310.01405*, 2023.

A Related Work

A.1 Training-time behavioral interventions

Shared intervention structure. Security vectors (Zhou et al., 2024) and ResT share a three-stage adapter procedure: fit an adapter to an undesirable behavior, freeze it while updating the backbone, and remove it before inference. Security vectors implement the intervention with LoRA (Hu et al., 2022). Preventive persona steering (Chen et al., 2025) applies a related idea in activation space, steering toward an undesirable trait during fine-tuning to limit its acquisition. Together, these methods motivate studying the behavioral state used during learning.

Other training-time interventions. Inoculation prompting elicits an unwanted trait through the training prompt and removes that prompt at test time to reduce subsequent trait expression (Tan et al., 2026; Wichers et al., 2025). Concept ablation fine-tuning instead ablates directions associated with an unwanted concept during fine-tuning (Casademunt et al., 2025). Such activation-space interventions build on evidence that behaviors such as refusal are mediated by low-dimensional directions (Zou et al., 2023; Rimsky et al., 2024; Arditi et al., 2024). A related idea appears in dataset debiasing, where a model of a known bias is combined with the main model during training, as a product of experts (Hinton, 2002; Clark et al., 2019; He et al., 2019; Sanh et al., 2021) or through reweighting (Nam et al., 2020), and discarded at test time; DExperts instead combines experts and anti-experts at decoding time (Liu et al., 2021).

Recent adapter-based interventions. Inoculation Adapters (Riché et al., 2026) fit an unwanted trait into a LoRA, freeze it while learning a separate task adapter, and remove it at deployment. Their SFT experiments study selective generalization from data containing both desired and undesired traits, across nine setups and five model families, and show that the frozen adapter produces fewer unintended conditional behaviors than inoculation prompting. Jailbreak to Protect (Ham et al., 2026) freezes a harmful-behavior adapter during fine-tuning on user data that may contain harmful examples, reducing safety-degrading updates to the task adapter, and then replaces it with a separately trained safety-reinforcement adapter. Both methods train a separate task LoRA under SFT, and in both the unwanted behavior may appear in the fine-tuning data. ResT studies forgetting of existing behaviors during benign domain adaptation, with the full backbone updated under a removable load by SFT, OPD, or OPSD. We compare behavioral sources and fitting objectives, evaluate adapter-conditioned on-policy learning, and analyze conditions for protection after removal.

Experimental scope. Security vectors investigate both harmfulness and hallucination under SFT. Their experiments include harmful or hallucination-inducing data, mixtures with task data, and benign task training: Table 1 of Zhou et al. (2024) includes pure ProQA fine-tuning with safety protection. Persona vectors apply preventive steering during SFT on datasets that elicit the targeted traits, including hallucination (Chen et al., 2025). Our experiments focus on behavioral retention during on-policy reasoning post-training, using medical and financial tasks and separate safety and hallucination protection settings. They examine whether task improvement can coexist with preservation or improvement of behaviors without explicit downstream task supervision.

Resistance construction. Security vectors fit their adapters with a causal language-modeling loss, an additional min–min optimization procedure, and a KL constraint on non-target data (Zhou et al., 2024). Preventive persona steering instead adds a fixed trait direction, extracted from contrastive prompts, to the activations (Chen et al., 2025), and inoculation adapters fit the unwanted trait with supervised fine-tuning (Riché et al., 2026). ResT fits a low-rank weight update and compares response supervision with on-policy distillation, where a frozen teacher supervises prefixes sampled by the adapter-conditioned student. We also examine generic, targeted, and forgetting-derived signals. This separates the choice of behavioral source and adapter-fitting objective from the subsequent downstream objective.

On-policy interventions. Dubiński et al. (2026) study on-policy data generation under inoculation prompting, evaluating whether it reduces conditional misalignment. Their intervention operates through the training context. A recent research discussion also proposes learning inoculation adapters on a model's own generations through distribution matching, listing this as a future experiment

(Xenomirant, 2026). We evaluate on-policy resistance fitting against response supervision and measure protection after benign task adaptation, including downstream SFT. This comparison separates the construction of a behavioral load from the choice of downstream learning objective.

On-policy dynamics and analysis. Under downstream OPD and OPSD, the resistance adapter participates in both rollout generation and loss computation. It therefore changes the distribution of states receiving supervision as well as the backbone gradient. Section 2.4 gives a local account of the update and a sufficient condition for protection after removal in a fixed-feature linearized model. The condition relates the adapter's learned behavioral displacement to task-feature coverage and transfer to held-out protection prompts. The analysis describes compensation on fixed trajectories, while the experiments include the adapter's effects on both sampling and optimization.

A.2 Behavioral forgetting and retention

Post-training degradation. Post-training can weaken the safety alignment of instruction models, even when the fine-tuning data are benign (Qi et al., 2024), and current alignment is often concentrated in the first few output tokens (Qi et al., 2025); narrow fine-tuning can even induce broad misalignment (Betley et al., 2025; Wang et al., 2026). Fine-tuning on new knowledge can increase hallucination (Gekhman et al., 2024), reinforcement fine-tuning can reduce refusals on unanswerable questions (Song et al., 2025), long chain-of-thought reasoning models can produce unsafe reasoning and responses (Jiang et al., 2025; Jeung et al., 2025), and preference alignment can in turn reduce previously acquired abilities (Lin et al., 2024). More broadly, research on catastrophic forgetting studies the loss of previously acquired capabilities during specialization (Kirkpatrick et al., 2017; Li & Hoiem, 2018; Kotha et al., 2024; Luo et al., 2025); low-rank adaptation forgets less than full fine-tuning but also learns less (Biderman et al., 2024).

On-policy learning is often credited with mitigating such forgetting: models trained on their own samples tend to forget less than those trained on fixed demonstrations (Shenfeld et al., 2026b; Chen et al., 2026), and on-policy distillation and self-distillation reduce the mismatch between training and inference (Shenfeld et al., 2026a; Zhao et al., 2026). Our experiments examine the remaining behavioral forgetting under these objectives: measured declines in previously exhibited behaviors outside downstream supervision. We use forgetting in this behavioral sense, without making claims about the model's internal knowledge.

Rehearsal, update constraints, and transfer. Rehearsal methods such as self-synthesized rehearsal retain previous behavior by mixing examples of it into the new training data (Huang et al., 2024a). Retention then becomes part of the downstream objective. Our Replay baseline uses the native task objective on the added prompts and keeps the number of epochs, increasing the number of updates. In our on-policy experiments, Replay restores refusal well but gives up some task accuracy, trailing vanilla training on MedQA by up to 1.51 points in the Hallucination setting. Update-constraint methods restrict where the weights may move, either by projecting low-rank updates toward a safety-aligned subspace (Hsu et al., 2024), by learning new tasks in orthogonal low-rank subspaces (Wang et al., 2023), or by freezing protected singular directions (Nayak et al., 2026). Their effectiveness depends on how well the chosen subspace captures the behavior; on MedQA, our adaptation of OSFT leaves refusal below vanilla training under both OPD and OPSD. Shadow-FT learns the task update on the paired base model and transfers it to the instruction model (Wu et al., 2025). These approaches act on the training data, the allowed update, and the model being trained, respectively. ResT intervenes in the instruction model's behavioral state while the task update is learned. Resistance-source examples fit a reusable adapter before task training; the downstream data, objective, and training length stay unchanged. This places resistance construction before downstream optimization and makes its combination with rehearsal a natural direction for future work.

Defenses against harmful fine-tuning. A related line of work protects safety alignment against fine-tuning on harmful data, either by hardening the aligned model before fine-tuning (Huang et al., 2024b; 2025; Rosati et al., 2024; Tamirisa et al., 2025) or by adding safety data during fine-tuning (Bianchi et al., 2024; Wang et al., 2024). These methods target harmful or adversarial training data, whereas we study the erosion of behavior during benign task training and also protect behaviors other than refusal.

Parameter-efficient update control. Task arithmetic treats parameter differences as composable task vectors (Ilharco et al., 2023). It operates on the learned update, whereas ResT conditions how that update is obtained. The resistance adapter supplies a temporary behavioral state during task learning and is removed afterward. Our fitting comparison examines two ways to obtain that state: fixed responses and teacher distributions along the student's own trajectories.

On-policy reasoning distillation. Knowledge distillation trains a student to match a teacher's output distribution (Hinton et al., 2015). On-policy methods reduce the mismatch between training and inference by training on trajectories sampled from the current student (Agarwal et al., 2024; Gu et al., 2024). OPSD further allows a single model to act as both teacher and student under different contexts (Zhao et al., 2026; Shenfeld et al., 2026a). Reinforcement learning (Shao et al., 2024; Guo et al., 2025) and preference optimization (Rafailov et al., 2023) offer further downstream objectives for studying resistance training.

B Resistance-Source Data Construction

Resistance-source data are constructed separately from the evaluation prompts and answers.

B.1 Safety refusal

The Safety study uses the 3,724 rows of the HarmChain training split (Jeung et al., 2025), built from BeaverTails questions (Ji et al., 2023),[1] which pairs harmful requests with harmful reasoning and responses. Response supervision uses the native reasoning and response pairs for targeted loads. For generic loads, we render each prompt with the instruction model's chat template in thinking mode, without pre-filling the reasoning block, and let Qwen3-4B-Base continue with greedy decoding; each sequence is limited to 4,096 tokens and cut at the last complete sentence. These base-model responses pass only mechanical audits (non-empty, within the length limit, tokenizer round-trip, unique identifiers). On-policy distillation uses the prompts only.

B.2 Factual restraint

Prompts come from the Apache-2.0 `safety-research/persona_vectors` repository. We remove the source assistant answers, normalize prompts (Unicode NFKC, case folding, and whitespace canonicalization), and de-duplicate them against the evaluation questions using exact matching and TF–IDF character 4–5-gram cosine similarity with threshold 0.82. Deterministic sampling with seed 42 then selects the prompt pool, from which we retain 1,000 prompts whose responses pass all filters below. Misaligned responses are generated by Huihui-Qwen3.8-27B-abliterated,[2] a 27B model whose refusal behavior has been removed, in thinking mode from the original prompt and a behavior-control instruction, without access to a reference answer. The reasoning and the final response are generated separately. The M2 instruction requires at least two concrete, mutually reinforcing unsupported claims stated confidently without retraction. A deterministic check rejects reasoning or responses that leak the control instruction, an LLM judge checks for implicit leakage and trait strength, and duplicates are removed. Base-model responses are sampled from Qwen3-4B-Base on the same 1,000 prompts with the same template protocol as in the Safety study and pass the same mechanical audits. All reported results use M2. The data-generation pipeline also produced an unused M1 variant with one localized unsupported claim.

B.3 Natural forgetting

Forgetting-derived loads use MedQA-SFT checkpoints trained for 1, 3, 5, and 10 epochs as resistance teachers (Section 3.4). Their token distributions supervise adapter fitting by on-policy distillation. The unwanted behavior is supplied by the checkpoint's naturally degraded behavior, rather than by specified failure responses.

[1] `https://huggingface.co/datasets/AI-ISL/HarmChain`

[2] `https://huggingface.co/huihui-ai/Huihui-Qwen3.8-27B-abliterated`

C Details and Proof of the Auxiliary Linear Interpretation

This appendix formalizes only the compensation mechanism in Theorem 2.1. It does not model the full nonlinear parameterization, Adam optimization, or the change in trajectories induced by an adapter-active on-policy model. In particular, $\boldsymbol{\theta}_0$ below is the instruction-tuned backbone before downstream training, not the paired pre-instruction Qwen3-4B-Base model used as one possible resistance source. The analysis applies to any learned adapter with the required measured effect on the protected behavioral axis. A targeted source is designed to have this effect; a generic source need not have it for every protected behavior.

C.1 Binary linear formulation

Behavioral variable and model. Let $Y \in \{0, 1\}$ describe one protected behavioral axis, with $Y = 0$ denoting the undesirable side and $Y = 1$ the desirable side. In the Safety setting these outcomes may represent harmful compliance and refusal; in the Hallucination setting they may represent unsupported commitment and factual or appropriately uncertain behavior. The variable abstracts the behavioral direction of a complete reasoning trajectory and response; it does not assert that natural-language generations are literally binary classes.

Let $\boldsymbol{h}(x) \in \mathbb{R}^d$ be a fixed representation of prompt x and let $\boldsymbol{\theta} \in \mathbb{R}^d$ be the trainable backbone parameter. The adapter-free model is

$$p_{\boldsymbol{\theta}}(Y = 1 \mid x) = \sigma\big(\boldsymbol{h}(x)^\top \boldsymbol{\theta}\big), \qquad \sigma(s) = \frac{1}{1 + e^{-s}},$$

so a larger logit means more desirable behavior. Write $z_0(x) = \boldsymbol{h}(x)^\top \boldsymbol{\theta}_0$ for the initial instruction model's logit. In the linearization, a rank-k LoRA adapter contributes $-U\boldsymbol{a}$, where $U \in \mathbb{R}^{d\times k}$ is its effective subspace. Defining $\boldsymbol{d} = U\boldsymbol{a}$, the adapter-active model is

$$p_{\boldsymbol{\theta},\boldsymbol{d}}(Y = 1 \mid x) = \sigma\big(\boldsymbol{h}(x)^\top (\boldsymbol{\theta} - \boldsymbol{d})\big).$$

Adapter construction fixes $\boldsymbol{\theta}_0$ and learns $\boldsymbol{d}^\star$ in $\mathcal{A} := \mathrm{Range}(U)$. A positive value of $\boldsymbol{h}(x)^\top \boldsymbol{d}^\star$ means that attaching the adapter lowers the desirable-behavior logit on x. During downstream training, $-\boldsymbol{d}^\star$ remains active and frozen while $\boldsymbol{\theta}$ changes. Deployment removes the adapter but retains the learned backbone change.

Prompt populations. Let D_{deal} be the resistance-source data defined in Section 2, and denote its prompt population by P_d. Let P_{task} be the actual downstream prompt population and P_p a held-out population evaluating the same protected behavior. In the experiments, downstream training samples only from D_{task}; it does not replay prompts, responses, or protected-behavior labels from D_{deal}.

To retain the prompt-mixture interpretation while respecting this separation, decompose the target population as

$$P_{\mathrm{task}} = (1 - \rho)P_o + \rho P_e, \qquad 0 \le \rho \le 1. \tag{5}$$

Here P_o contains target prompts that do not expose the resistance-relevant directions, whereas P_e is the component of the target distribution that does. Equation 5 is therefore a decomposition of D_{task}, not a mixture of target and resistance-source datasets. A nonzero ρ means only that some target prompts exercise features through which the adapter acts. The theorem below depends on the resulting target-task feature span, not on observing which prompt belongs to which component.

Response supervision. Response supervision fits the adapter on supplied resistance-source prompt/response pairs. For a response from a targeted source that instantiates the undesirable side of the protected axis, the binary label is $Y = 0$; it is provided by the data rather than generated by a teacher during adapter training. Its binary objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{d}) &= \mathbb{E}_{X\sim P_d}\left[-\log p_{\boldsymbol{\theta}_0,\boldsymbol{d}}(Y = 0 \mid X)\right] \\ &= \mathbb{E}_{X\sim P_d}\left[\log\big(1 + \exp\big(z_0(X) - \boldsymbol{h}(X)^\top \boldsymbol{d}\big)\big)\right], \qquad \boldsymbol{d} \in \mathcal{A}.\end{aligned}$$

Let $\boldsymbol{d}^\star_{\mathrm{CE}}$ be the displacement at the selected adapter checkpoint and define

$$\widehat{r}_{\mathrm{CE}}(x) := p_{\boldsymbol{\theta}_0,\boldsymbol{d}^\star_{\mathrm{CE}}}(Y = 1 \mid x), \qquad \epsilon_{\mathrm{CE}} := \sup_{x\in\mathrm{supp}(P_d)} \widehat{r}_{\mathrm{CE}}(x),$$

$$p_{0,\min} := \inf_{x\in\mathrm{supp}(P_d)} p_{\boldsymbol{\theta}_0}(Y = 1 \mid x), \qquad \gamma_{\mathrm{CE}} := \inf_{x\in\mathrm{supp}(P_d)} \boldsymbol{h}(x)^\top \boldsymbol{d}^\star_{\mathrm{CE}}.$$

Here ϵ_{CE} is the largest desirable-behavior probability left on a resistance-source prompt after fitting the supplied undesirable responses, while γ_{CE} is the smallest resistance margin. A small average CE loss alone does not ensure a large worst-case margin; the source prompts must be fitted sufficiently uniformly. This construction-specific bound idealizes the supplied responses as expressing $Y = 0$. The general theorem below still applies to a response-supervised generic adapter only when its measured displacement has the required positive margin.

On-policy distillation. On-policy distillation uses a frozen same-architecture resistance teacher q_{ψ_d} along trajectories sampled from the adapter-conditioned student. In the one-decision binary abstraction, define

$$r_d(x) := q_{\psi_d}(Y = 1 \mid x), \qquad t_d(x) := \mathrm{logit}(r_d(x)),$$
$$g_d(x) := z_0(x) - t_d(x), \qquad m_d := \inf_{x \in \mathrm{supp}(P_d)} g_d(x).$$

The gap $g_d(x)$ measures how far the resistance teacher moves the protected logit toward the undesirable side relative to the instruction model. Thus $m_d > 0$ means that the teacher consistently exhibits more of the failure on every resistance-source prompt. In the load study of Section 3.4, a more severely failing teacher is intended to increase this gap. The fitted adapter-active logit is

$$\widehat{t}_d(x) := z_0(x) - \boldsymbol{h}(x)^\top \boldsymbol{d}^\star_{\mathrm{OPD}},$$

and its worst-case teacher-fitting error is

$$\eta_{\mathrm{fit}} := \sup_{x \in \mathrm{supp}(P_d)} \left| \widehat{t}_d(x) - t_d(x) \right|.$$

This is an adapter-construction error measured in log-odds, not a downstream-task fitting error. It is zero when the adapter-active instruction model matches the resistance teacher on P_d.

Writing $\widehat{r}_{\boldsymbol{d}}(x) = \sigma(z_0(x) - \boldsymbol{h}(x)^\top \boldsymbol{d})$, the binary form of the reverse-KL objective used by on-policy distillation is

$$\mathcal{L}_{\mathrm{OPD}}(\boldsymbol{d}) := \mathbb{E}_{X \sim P_d} \left[D_{\mathrm{KL}}(\mathrm{Ber}(\widehat{r}_{\boldsymbol{d}}(X)) \,\|\, \mathrm{Ber}(r_d(X))) \right], \qquad \boldsymbol{d} \in \mathcal{A}.$$

Its Fisher-local quadratic form projects the teacher gap onto functions realizable by the adapter:

$$\min_{\boldsymbol{d} \in \mathcal{A}} \frac{1}{2} \mathbb{E}_{P_d} \left[w_d(X) \big(\boldsymbol{h}(X)^\top \boldsymbol{d} - g_d(X) \big)^2 \right], \qquad w_d(x) = r_d(x)(1 - r_d(x)).$$

The corresponding resistance margin is

$$\gamma_{\mathrm{OPD}} := \inf_{x \in \mathrm{supp}(P_d)} \boldsymbol{h}(x)^\top \boldsymbol{d}^\star_{\mathrm{OPD}}.$$

Paired downstream training. Here, response supervision and on-policy distillation describe adapter fitting, not the subsequent downstream algorithm. To represent paired vanilla and resistance-trained SFT, OPD, or OPSD, let $t_A(x)$ and $w_A(x) > 0$ denote a common local pseudo-target logit and curvature for downstream algorithm A. They arise from a second-order local model of the corresponding token-level objective. For OPD they reflect the external task teacher; for OPSD they reflect the privileged self-teacher; for SFT they reflect the supplied response under the local logistic expansion.

For a fixed prompt or sampled trajectory, the paired quadratic objectives are

$$\mathcal{L}_A^{\mathrm{van}}(\boldsymbol{\theta}) = \frac{1}{2} \mathbb{E}_{P_{\mathrm{task}}} \left[w_A(X) \big(\boldsymbol{h}(X)^\top \boldsymbol{\theta} - t_A(X) \big)^2 \right], \tag{6}$$

$$\mathcal{L}_A^{\mathrm{ResT}}(\boldsymbol{\theta}) = \frac{1}{2} \mathbb{E}_{P_{\mathrm{task}}} \left[w_A(X) \big(\boldsymbol{h}(X)^\top (\boldsymbol{\theta} - \boldsymbol{d}^\star) - t_A(X) \big)^2 \right]. \tag{7}$$

The two runs start from the same $\boldsymbol{\theta}_0$ and share prompts, local targets, curvatures, and optimization time. The adapter is fixed only in the resistance-trained run. For OPD and OPSD, the shared quantities define a conditional, fixed-trajectory comparison. In the actual method, the adapter is also active during rollout generation and therefore changes the sampled trajectories; that additional effect is outside this theorem.

Define

$$M_{\text{task}} := \mathbb{E}_{P_{\text{task}}}\left[w_A(X)\boldsymbol{h}(X)\boldsymbol{h}(X)^\top\right], \qquad \boldsymbol{b}_{\text{task}} := \mathbb{E}_{P_{\text{task}}}\left[w_A(X)\boldsymbol{h}(X)t_A(X)\right],$$

and let $\Pi_{\text{task}} = M_{\text{task}}^\dagger M_{\text{task}}$ be the orthogonal projector onto $\text{Range}(M_{\text{task}})$. The unexposed portion of the resistance displacement is

$$\epsilon_{\text{miss}} := \|(I - \Pi_{\text{task}})\boldsymbol{d}^\star\|.$$

For $j \in \{o, e\}$, also define

$$M_j := \mathbb{E}_{P_j}\left[w_A(X)\boldsymbol{h}(X)\boldsymbol{h}(X)^\top\right], \qquad \mathcal{S}_j := \text{Range}(M_j).$$

When $0 < \rho < 1$, the exposed space is $\text{Range}(M_{\text{task}}) = \mathcal{S}_o + \mathcal{S}_e$. This equality concerns two components of the target distribution in Equation 5; neither component is P_d.

Finally, define the held-out coverage error and feature radius by

$$\epsilon_{\text{cov}} := \sup_{x' \in \text{supp}(P_p)} \inf_{x \in \text{supp}(P_d)} \|\boldsymbol{h}(x') - \boldsymbol{h}(x)\|, \tag{8}$$

$$R_p := \sup_{x' \in \text{supp}(P_p)} \|\boldsymbol{h}(x')\|.$$

The first quantity is small when resistance-source prompts cover the held-out protection prompts in feature space. The supremum gives a pointwise guarantee; replacing it by a mean or quantile yields an average or high-probability analogue.

C.2 Behavioral-gain theorem and proof

Theorem C.1 (Formal restatement of Theorem 2.1)**.** *Choose either response supervision or on-policy distillation to fit the adapter and denote the learned resistance displacement by $\boldsymbol{d}^\star$. Let*

$$\gamma^\star := \inf_{x \in \text{supp}(P_d)} \boldsymbol{h}(x)^\top \boldsymbol{d}^\star$$

be its resistance margin. Train the paired objectives in Equations 6–7 by gradient flow from the same $\boldsymbol{\theta}_0$ to convergence, and remove the adapter from the resistance-trained run. Then every $x' \in \text{supp}(P_p)$ satisfies

$$\Delta z_p(x') := z_{\text{ResT}}(x') - z_{\text{vanilla}}(x') \geq \gamma^\star - \epsilon_{\text{cov}}\|\boldsymbol{d}^\star\| - R_p\epsilon_{\text{miss}}. \tag{9}$$

Consequently,

$$\gamma^\star > \epsilon_{\text{cov}}\|\boldsymbol{d}^\star\| + R_p\epsilon_{\text{miss}} \quad \Longrightarrow \quad p_{\text{ResT}}(Y = 1 \mid x') > p_{\text{vanilla}}(Y = 1 \mid x') \tag{10}$$

for every held-out prompt $x' \in \text{supp}(P_p)$.

Proof. Let $\boldsymbol{\theta}_{\text{van}}(\tau)$ and $\boldsymbol{\theta}_{\text{ResT}}(\tau)$ follow gradient flow on the two quadratic objectives, and define $\boldsymbol{\Delta}(\tau) = \boldsymbol{\theta}_{\text{ResT}}(\tau) - \boldsymbol{\theta}_{\text{van}}(\tau)$. Their dynamics are

$$\dot{\boldsymbol{\theta}}_{\text{van}} = -M_{\text{task}}\boldsymbol{\theta}_{\text{van}} + \boldsymbol{b}_{\text{task}},$$

$$\dot{\boldsymbol{\theta}}_{\text{ResT}} = -M_{\text{task}}(\boldsymbol{\theta}_{\text{ResT}} - \boldsymbol{d}^\star) + \boldsymbol{b}_{\text{task}}.$$

Subtracting cancels the shared downstream target:

$$\dot{\boldsymbol{\Delta}} = -M_{\text{task}}\boldsymbol{\Delta} + M_{\text{task}}\boldsymbol{d}^\star, \qquad \boldsymbol{\Delta}(0) = 0.$$

Every component in a positive-eigenvalue direction of M_{task} converges to the corresponding component of $\boldsymbol{d}^\star$, while every null-space component remains zero. Hence

$$\boldsymbol{\Delta}(\infty) = \Pi_{\text{task}}\boldsymbol{d}^\star. \tag{11}$$

Fix $x' \in \text{supp}(P_p)$ and take $x \in \text{supp}(P_d)$ arbitrarily close to the inner infimum in Equation 8. Equation 11 gives

$$\begin{aligned}
\Delta z_p(x') &= \boldsymbol{h}(x')^\top \Pi_{\text{task}}\boldsymbol{d}^\star \\
&= \boldsymbol{h}(x)^\top \boldsymbol{d}^\star + \big(\boldsymbol{h}(x') - \boldsymbol{h}(x)\big)^\top \boldsymbol{d}^\star - \boldsymbol{h}(x')^\top (I - \Pi_{\text{task}})\boldsymbol{d}^\star \\
&\geq \gamma^\star - \epsilon_{\text{cov}}\|\boldsymbol{d}^\star\| - R_p\epsilon_{\text{miss}},
\end{aligned}$$

where the last line follows from Cauchy–Schwarz and the definitions of the two errors. This proves Equation 9. If the strict condition in Equation 10 holds, then $\Delta z_p(x') > 0$. Since the sigmoid is strictly increasing, the deployed resistance-trained model assigns greater probability to $Y = 1$ than the matched vanilla model. □

Theorem C.1 separates behavioral gain into one favorable quantity and two errors. The margin $\gamma^\star$ measures the smallest undesirable logit displacement learned by the adapter on resistance-source prompts. The missing-direction error measures the part of that displacement not exposed by target-task features, and the coverage error measures transfer from resistance-source prompts to held-out protection prompts.

When the learned displacement is strong. For on-policy distillation, every $x \in \mathrm{supp}(P_d)$ obeys

$$\begin{aligned}\boldsymbol{h}(x)^\top \boldsymbol{d}^\star_{\mathrm{OPD}} &= z_0(x) - \widehat{t}_d(x) \\ &= g_d(x) - \left[\widehat{t}_d(x) - t_d(x)\right] \geq m_d - \eta_{\mathrm{fit}}.\end{aligned}$$

Thus the margin is large when the resistance teacher consistently moves the protected logit toward the undesirable side and on-policy distillation reproduces that shift accurately. Holding fitting and coverage errors fixed, increasing the teacher's failure severity increases m_d and makes the gain condition easier to satisfy. This is the aspect of resistance load captured by the theory; the analysis does not assert that changing the teacher leaves the other terms fixed in a nonlinear model. For response supervision, monotonicity of the logit gives

$$\boldsymbol{h}(x)^\top \boldsymbol{d}^\star_{\mathrm{CE}} \geq \mathrm{logit}(p_{0,\min}) - \mathrm{logit}(\epsilon_{\mathrm{CE}}).$$

Its margin is therefore large when the instruction model initially assigns nontrivial probability to the desirable behavior and response supervision uniformly fits the supplied undesirable responses. These conditions concern the adapter's measured behavioral effect, not its source label or parameter norm. They may fail for a generic adapter even when its construction loss is small, especially when the generic source exhibits little of the protected failure.

When prompt diversity is sufficient. For $0 < \rho < 1$, positive semidefiniteness of M_o and M_e gives

$$\epsilon_{\mathrm{miss}} = \mathrm{dist}(\boldsymbol{d}^\star, \mathcal{S}_o + \mathcal{S}_e)\,. \tag{12}$$

Equation 12 shows that the missing-direction error is small when the target-task prompts collectively expose the learned resistance direction. A large or diverse D_{task} helps only when it adds relevant feature directions, including prompts from the exposure component P_e. Separately, ϵ_{cov} is small when the resistance-source prompts cover the features of held-out Safety or Hallucination prompts. Diversity across categories, reasoning patterns, severities, and paraphrases is useful insofar as it improves this transfer. These are distinct requirements: D_{task} exposes the compensation direction, while D_{deal} supports its transfer to the protection benchmark. Neither requires replaying resistance-source data during downstream training.

Corollary C.2 (Construction-specific sufficient conditions)**.** *Under the setting of Theorem C.1, on-policy distillation yields a strict held-out behavioral gain whenever*

$$m_d - \eta_{\mathrm{fit}} > \epsilon_{\mathrm{cov}} \|\boldsymbol{d}^\star_{\mathrm{OPD}}\| + R_p \epsilon_{\mathrm{miss}}, \tag{13}$$

and response supervision yields a strict held-out behavioral gain whenever

$$\mathrm{logit}(p_{0,\min}) - \mathrm{logit}(\epsilon_{\mathrm{CE}}) > \epsilon_{\mathrm{cov}} \|\boldsymbol{d}^\star_{\mathrm{CE}}\| + R_p \epsilon_{\mathrm{miss}}. \tag{14}$$

In each condition, ϵ_{miss} is evaluated for the corresponding learned adapter direction.

Proof. Taking the infimum in the two margin bounds above gives $\gamma_{\mathrm{OPD}} \geq m_d - \eta_{\mathrm{fit}}$ and $\gamma_{\mathrm{CE}} \geq \mathrm{logit}(p_{0,\min}) - \mathrm{logit}(\epsilon_{\mathrm{CE}})$. The conditions in Equations 13 and 14 therefore each imply the gain condition in Equation 10. □

The main paper's local-curvature identity and this result answer complementary questions. The former shows that a frozen adapter can alter a backbone update without receiving gradients. Theorem C.1 gives an infinite-time sign guarantee for the approximated quadratic fixed-feature model. Neither result ranks response supervision and on-policy distillation as optimization procedures; the construction with the larger resistance margin and smaller errors receives the stronger certificate.

D EXPERIMENTAL DETAILS

D.1 MODELS, FRAMEWORKS, AND HARDWARE

All students are the instruction-tuned Qwen3-4B (Yang et al., 2025) in thinking mode; training and evaluation keep explicit reasoning traces and use the native Qwen3 chat template. The paired base model is Qwen3-4B-Base. OPD uses Qwen3-32B as the task teacher, and OPSD uses an exponential moving average (EMA) of the student backbone as a teacher conditioned on a privileged demonstration (Zhao et al., 2026; Shenfeld et al., 2026a), following the reference implementation of Shenfeld et al. (2026a). SFT uses ms-swift (Zhao et al., 2025), OPD uses verl (Sheng et al., 2025), and OPSD uses the TRL-based Self-Distillation implementation. On-policy rollouts use vLLM (Kwon et al., 2023). Sources and licenses of all assets are listed in Appendix D.6. All runs use NVIDIA H100 GPUs and bfloat16 precision. Training uses seed 42, except for the separate single-seed natural-forgetting sweep specified below.

D.2 DOWNSTREAM DATA AND TRAINING

Medical reasoning data. We construct the downstream corpus from the MedQA training split (Jin et al., 2021). A Qwen3-32B teacher receives the gold answer as privileged context and produces five candidate reasoning traces per question, while the student view contains only the question and its answer choices. GPT-4.1 judges the correctness of each candidate, and we keep the earliest correct one; generations that reveal the privileged answer, detected by either a semantic judgment or a strict lexical check, are rejected. This procedure retains 9,781 of 9,823 questions. We then remove the 24 examples whose privileged teacher prompt exceeds 4,096 tokens, leaving 9,757 examples that all downstream objectives share.

Financial reasoning data. The financial corpus is built from the FinQA reasoning paths of Fino1 (Qian et al., 2025),[3] which provide 5,499 FinQA training examples (Chen et al., 2021) with reference answers. We regenerate the reasoning with a privileged-context procedure analogous to that for MedQA: a Qwen3-32B teacher receives the reference answer as privileged context and produces five candidate reasoning traces and responses per question. A candidate is discarded if the FinQA answer checker marks it incorrect or if Qwen3.6-27B judges that its reasoning or response reveals prior knowledge of the answer; we keep the earliest remaining candidate and drop questions without one, leaving 4,133 examples.

We check every question against the 1,147 FinQA test questions, the 1,405 AIR-Bench prompts, and the 20 hallucination evaluation questions by exact matching after NFKC normalization and removal of FinQA's context and question wrappers, and check for exact duplicates of the question, trace, and response. Neither audit finds a match.

Table 3 lists the FinQA configurations, which vanilla and resistance-trained runs share. SFT pads the data to 4,160 rows by repeating 27 examples so that every batch is full, and uses a cosine schedule with 11 warm-up steps; OPD uses a constant learning rate without warm-up; OPSD uses a cosine schedule with 35 warm-up steps. No FinQA training sequence reaches its length limit.

[3] https://huggingface.co/datasets/TheFinAI/Fino1_Reasoning_Path_FinQA

Table 3: Downstream training configurations, shared by vanilla and resistance-trained runs within each domain. Lengths are prompt/response limits, except for SFT, which reports the total sequence limit. The vanilla SFT reference in Table 11 was trained with an earlier recipe (9,781 examples, an 8,192-token limit, and 25 warm-up steps).

Objective	Examples	Epochs	Updates	LR	Batch	Length
MedQA						
SFT	9,757	3	459	5×10^{-6}	64 sequences	4,096
OPD	9,757	1	406	5×10^{-6}	24 prompts $\times$ 4 rollouts	2,048 / 8,192
OPSD	9,757	1	406	5×10^{-6}	24 prompts $\times$ 4 rollouts	4,096 / 8,192
FinQA						
SFT	4,133	3	195	5×10^{-6}	64 sequences	8,192
OPD	4,133	2	344	5×10^{-6}	24 prompts $\times$ 4 rollouts	4,096 / 6,144
OPSD	4,133	2	344	5×10^{-6}	24 prompts $\times$ 4 rollouts	4,096 / 8,192

SFT. The target concatenates the reasoning trace and final response under the native thinking template, and cross-entropy is applied to the complete assistant turn. Training uses a per-device batch size of 2 with four gradient-accumulation steps on eight GPUs, no weight decay, and ZeRO-2. The cosine schedule has 50 warm-up steps for MedQA and 11 for FinQA. No training sequence is truncated.

OPD. The student samples four continuations per prompt at temperature 1. Along these student-generated prefixes, the loss is a per-token reverse-KL estimate (k1) to the Qwen3-32B teacher computed over the teacher's top-64 tokens and optimized with a policy-gradient update; task rewards are disabled. Six GPUs train the student and two serve the teacher. The learning rate is constant, without warm-up.

OPSD. The student sees the question, samples four completions at temperature 1, and learns from an EMA teacher conditioned on the question and its privileged reasoning trace and response. The teacher tracks the student backbone with mixup coefficient 0.01 at every update; it scores the student's trajectories and carries no resistance adapter. We use the reference implementation's reverse-KL setting ($\alpha = 1$), skip the first three completion tokens in the loss, and exclude completions that reach the length limit. Each update covers 24 distinct prompts and 96 sampled sequences: per-device batch size 1, 12 gradient-accumulation steps, and eight GPUs. The learning rate follows a cosine schedule with a 10% warm-up period.

Training with the resistance adapter. The frozen adapter participates in the student forward pass and rollout generation, conditioning the computation through which the backbone receives gradients. SFT attaches the adapter to the training model as a non-trainable module. OPD and OPSD fold the same low-rank update (144 modules) into the vLLM rollout weights, which we verify at the start of every run. We save and evaluate the updated backbone after removing the resistance weights. Neither the separate OPD teacher nor the OPSD EMA teacher carries the resistance adapter. Vanilla and resistance-trained runs use the same task objective and, except for the earlier vanilla SFT recipe noted in Table 3, the same training configuration.

D.3 Baselines

Shadow-FT. Following Wu et al. (2025), we train Qwen3-4B-Base with exactly the same downstream configuration as the corresponding vanilla run; under OPD and OPSD, the base model also generates the rollouts. The resulting update is then transferred to the instruction model as $W_{\text{shadow}} = W_{\text{instruct}} + (W_{\text{base}}^{\text{after}} - W_{\text{base}}^{\text{before}})$, computed in float32.

Under OPD, 80.6% of Shadow-FT's MedQA responses reach the generation limit, with median response lengths 9–20$\times$ those of the other OPD models. The same behavior affects FinQA (982 of 1,147 responses) and finance-domain AIR-Bench generations. Table 1 reports the measured scores and flags this generation behavior.

OSFT. We adapt orthogonal subspace fine-tuning (Nayak et al., 2026) to our setting. Each of the 252 target matrices (query, key, value, output, gate, up, and down projections in all 36 layers) is factorized by SVD; the top-k singular factors are frozen and the complementary factors are trained, with gradients projected before each optimizer step and factors re-projected afterwards.

Calibration uses only the questions from the same resistance-source pools as ResT: 1,000 distinct M2 questions and 3,724 HarmChain rows containing 2,049 distinct questions. We forward these questions through the unchanged instruction model, retaining row multiplicities as weights. Calibration excludes responses and evaluation prompts. The protected rank of each matrix is $k = \lfloor (0.1 + 0.7\hat{s}) \min(d_{\text{in}}, d_{\text{out}}) \rfloor$ clipped to $[1, \min(d_{\text{in}}, d_{\text{out}})]$, where $\hat{s}$ is the token-weighted input–output cosine similarity, clipped below at zero and divided by the mean across matrices. For rectangular matrices, the cosine uses the first $\min(d_{\text{in}}, d_{\text{out}})$ coordinates. The rank coefficients 0.1 and 0.7 use the original defaults. On average, 45.6% (Hallucination) and 45.5% (Safety) of each matrix's rank is protected. SVD bases and factors are kept in float32 with bfloat16 activations, and all other settings match vanilla training. MedQA and FinQA share the resulting rank allocation. This adaptation uses activation statistics as a proxy for allocating protected ranks, rather than a direct measure of behavioral importance.

Replay. Our Replay baseline uses self-generated responses for rehearsal (Huang et al., 2024a), with the synthesis and filtering procedure described here. For each protected behavior, the original Qwen3-4B instruction model answers the questions of the corresponding adapter data in thinking mode (temperature 0.6, top-p 0.95, top-k 20, seed 42); only the questions are used, never the misaligned responses. Format and length filtering retains 990 of 1,000 Hallucination and 3,720 of 3,724 Safety rows. For MedQA, each pool is appended once to the 9,757-example task corpus, giving 10,747 (Hallucination) or 13,477 (Safety) training examples. SFT trains on the generated reasoning and answers; OPD uses only the questions, with the unchanged Qwen3-32B teacher; OPSD uses the generated reasoning and answer as privileged teacher context. Replay keeps the learning rate and number of epochs, so its MedQA update count grows (SFT: 504 and 633; OPD and OPSD: 447 and 561, for Hallucination and Safety, respectively). Exact and normalized overlap audits found no replay question in MedQA, AIR-Bench, or the hallucination evaluation prompts.

D.4 Resistance adapters

All resistance adapters are LoRA modules with rank 16 and $\alpha = 32$ on the query, key, value, and output projections of the instruction model, trained with AdamW (no weight decay, gradient clipping at 1.0) in bfloat16 with seed 42. Adapter-fitting duration is measured in epochs; it is distinct from resistance-load strength and from the teacher-training epochs varied in Section 3.4. Each adapter is trained independently from scratch with its own complete schedule, and fractional epochs are implemented as a fixed number of optimizer steps. Table 8 lists the recipe of each adapter family.

Main-table loads. For Tables 1 and 2, the default Safety adapter is fitted with a single pass over a random quarter of the 3,724 HarmChain rows; the Hallucination adapter makes two passes over its 1,000 prompts. The generic Safety entry for MedQA OPSD (75.65 task accuracy, 26.89 refusal) uses an earlier one-epoch adapter. Targeted Safety uses native HarmChain responses for response supervision and a frozen HarmChain-trained teacher initialized from Qwen3-4B-Base for on-policy distillation. Targeted Hallucination uses M2 responses for response supervision and a teacher fine-tuned on M2 for on-policy distillation. The generic source uses responses from, or distillation against, the untuned Qwen3-4B-Base model. Section 3.3 separately compares response supervision and on-policy distillation across fitting schedules.

Selection for the fitting comparison. For each source, fitting objective, and downstream objective, Figure 3 uses the configuration with the highest MedQA accuracy in the original sweep. Selecting on the reported evaluation scores makes task accuracy optimistic. The comparison evaluates the complete fitting procedures, whose compute and induced behavioral strength can differ. The generic hallucination comparison under OPD also spans two source lineages: earlier base-model responses for response supervision and distillation against the untuned Qwen3-4B-Base teacher for on-policy fitting. Table 7 reports a separate batch with matched source lineage and sequence count; this batch does not enter Figure 3's selection.

The duration comparisons below use the shared grid of 0.1, 0.25, 0.5, and 1 epoch for Safety and 0.5, 1, 2, and 4 epochs for Hallucination. Table 4 lists the durations selected from the original fitting sweep for Figure 3. The main tables use the fixed schedules specified above, and the vanilla SFT reference follows the earlier recipe in Table 3.

Table 4: Adapter-fitting epochs selected for Figure 3. Response and On-policy denote the fitting objective; each entry maximizes reported MedQA accuracy within its source–fitting–downstream configuration.

		Generic		Targeted	
Downstream	Protection	Response	On-policy	Response	On-policy
OPD	Safety	1	1	1	0.5
	Hallucination	4	2	4	0.5
SFT	Safety	0.1	1	0.5	0.5
	Hallucination	2	4	4	4

Fitting-duration sweeps. Tables 5 and 6 report the original recorded duration sweeps, including configurations not selected for Figure 3. These are independent adapter fits at each duration, followed by downstream training and adapter removal. The SFT table covers both fitting objectives; the OPD table covers on-policy fitting. For response-supervised OPD, Figure 3 reports the selected configurations. The Safety one-epoch on-policy adapters come from an earlier fitting batch than the fractional-epoch adapters; each duration has its own learning-rate schedule.

Under downstream SFT, on-policy fitting gives lower hallucination scores for both sources at every shared duration (0.5, 1, 2, and 4 epochs). Thus the hallucination advantage is not confined to the task-selected configurations. Task accuracy is higher in five of these eight comparisons. Refusal follows a different pattern: on-policy fitting is stronger for targeted loads at 0.25 and 0.5 epochs, but weaker at 0.1 and 1 epoch. Longer fitting is not uniformly better for either behavior. Equal epochs do not match tokens, compute, or behavioral displacement: response supervision fits fixed responses, whereas on-policy fitting uses four student rollouts per prompt and teacher distributions.

Table 5: Fitting-duration comparisons under downstream SFT at durations shared by both fitting objectives. Each task/behavior pair comes from the same deployed model. Safety reports refusal (higher is better); Hallucination reports the mean trait score (lower is better). SFT settings and reference configurations are given in Appendix E.

		Response supervision		On-policy fitting	
Source	Epochs	Task	Behavior	Task	Behavior
		Safety			
Generic	0.1	74.71	32.54	71.72	30.10
Generic	0.25	74.23	31.74	73.06	31.64
Generic	0.5	73.53	33.02	71.64	31.74
Generic	1	74.31	31.71	74.39	32.40
Targeted	0.1	73.76	35.20	71.80	31.23
Targeted	0.25	73.92	33.77	72.66	38.50
Targeted	0.5	74.94	32.73	73.68	34.96
Targeted	1	74.08	32.40	71.96	30.94
		Hallucination			
Generic	0.5	73.84	47.51	74.08	37.33
Generic	1	73.53	44.58	72.43	27.83
Generic	2	74.39	42.80	74.00	27.10
Generic	4	73.21	45.49	75.26	31.11
Targeted	0.5	72.58	48.17	75.26	31.06
Targeted	1	74.39	41.58	76.04	14.38
Targeted	2	74.86	33.44	74.78	11.36
Targeted	4	75.73	43.73	76.28	17.73

Table 6: Recorded on-policy fitting sweeps under downstream OPD. Each task/behavior pair describes an adapter-free model. Epochs measure adapter fitting, not resistance-teacher training. Safety reports refusal; Hallucination reports the mean trait score.

		Generic load		Targeted load	
Protection	Epochs	Task	Behavior	Task	Behavior
Safety	0.1	77.38	29.38	80.52	30.88
Safety	0.25	79.65	30.28	80.13	37.32
Safety	0.5	79.97	28.21	81.85	34.39
Safety	1	81.07	30.36	81.07	30.45
Hallucination	0.5	79.58	35.04	81.07	30.31
Hallucination	1	79.89	27.31	79.18	11.25
Hallucination	2	80.28	24.55	79.50	15.96
Hallucination	4	79.73	29.80	79.89	17.62

A sequence-matched comparison. Table 7 compares generic hallucination loads from the same base-model data lineage, using four epochs of response supervision or one epoch of on-policy fitting. Both process 4,000 sequences from 1,000 prompts; the latter uses four student rollouts per prompt. This controls sequence count and source lineage, but not token count or compute. Under downstream OPD, on-policy fitting lowers hallucination by 13.44 points with a 0.39-point task cost. Under OPSD, this one-epoch load performs worse on both metrics. This contrast shows that the downstream objective and fitting schedule matter when choosing how to construct the load.

Table 7: Sequence-matched generic hallucination loads on MedQA. Both fitting procedures use 4,000 sequences and the same base-model source lineage. Scores describe the deployed backbone after adapter removal.

	Response supervision		On-policy fitting	
Downstream	Task ↑	Hallucination ↓	Task ↑	Hallucination ↓
OPD	80.28	40.75	79.89	27.31
OPSD	77.22	37.31	70.54	44.64

Response supervision. Response supervision minimizes cross-entropy on the response only; prompt tokens are masked, and the response is used as generated, without re-rendering or an injected empty reasoning block. The Hallucination recipe uses 10 warm-up steps followed by cosine decay and no dropout; the Safety recipe, shared by both sources, uses a one-cycle schedule with 5% warm-up, LoRA dropout 0.05, and a 4,096-token limit.

On-policy distillation. The student, i.e., the instruction model with the trainable adapter in thinking mode, generates four rollouts per prompt with responses of up to 4,096 tokens. A frozen teacher scores the same token sequences, and only the adapter parameters are updated with a k1 reverse-KL estimate and a policy-gradient update; task rewards are disabled. Each update uses 20 prompts; five GPUs train the student and two serve the teacher. The learning rate follows 10 warm-up steps and cosine decay.

Teachers for on-policy distillation. The generic resistance teacher is Qwen3-4B-Base without further fine-tuning, kept frozen. Targeted resistance teachers are obtained by full-parameter fine-tuning of Qwen3-4B-Base on the corresponding misaligned data. The Hallucination teacher is trained with cross-entropy on the 1,000 M2 examples (global batch size 20, learning rate 10^{-5}, 200 steps, i.e., four epochs), and we use its final checkpoint without selection. The Safety teacher is trained on HarmChain reasoning and harmful responses (global batch size 8, learning rate 5×10^{-6}, maximum length 3,072, ZeRO-3, 466 steps, i.e., one epoch). These resistance teachers are used only to fit adapters; downstream distillation uses the task teachers described above.

Table 8: Resistance-adapter recipes. Response and On-policy denote fitting by response supervision and on-policy distillation. Batch is the number of examples (response supervision) or prompts × rollouts (on-policy distillation) per optimizer step. All adapters use LoRA rank 16 and $\alpha = 32$ on the attention projections.

Setting	Fitting	Source	Data	Batch	LR	Steps/ep.
Hallucination	Response	Targeted	1,000 M2 examples	20	5×10^{-5}	50
Hallucination	Response	Generic	1,000 base-model responses	20	5×10^{-5}	50
Hallucination	On-policy	Targeted	1,000 prompts, M2 teacher	20×4	5×10^{-5}	50
Hallucination	On-policy	Generic	1,000 prompts, base-model teacher	20×4	5×10^{-5}	50
Safety	Response	Targeted	3,724 HarmChain pairs	16	10^{-4}	233
Safety	Response	Generic	3,724 base-model responses	16	10^{-4}	233
Safety	On-policy	Targeted	3,724 rows, harm teacher	20×4	5×10^{-5}	186
Safety	On-policy	Generic	3,724 rows, base-model teacher	20×4	5×10^{-5}	186

Cost. The main-table resistance adapters are reused across downstream objectives, except for the generic Safety exception noted above. The fitting ablation separately varies adapter-training duration. Resistance training adds no downstream updates and no inference cost; its additional cost is the adapter training above and, for on-policy distillation with a targeted source, the one-time training of the teacher.

D.5 Evaluation

Comparison protocol. Each main-table configuration represents one training run. Evaluation repeats measure decoding variability of that checkpoint; their aggregation is specified below. We retain the historical vanilla AIR-Bench measurements as references, with evaluation-seed matching unverified. The vanilla SFT reference uses the earlier recipe in Table 3.

All evaluated models are full checkpoints without any resistance adapter, generate in thinking mode using vLLM 0.19.1, and, except for FinQA, have their reasoning removed before judging. The generation context limit is 40,960 tokens; benchmark-specific decoding settings are given below.

MedQA. We evaluate on the 1,273 test questions with temperature 0.5, top-p 0.9, a 36,864-token generation limit, and repetition penalty 1.0. Each question receives one generation, with no retry for invalid output. Qwen3.6-27B judges the final answer against the gold option, and a rule-based answer extractor serves as a cross-check. We report accuracy with generation seed 42, except for the initial-model and vanilla OPD/OPSD references, which report means and sample standard deviations over evaluation seeds 42, 43, and 44: 71.80 ± 0.63, 79.36 ± 0.76, and 75.81 ± 0.48, respectively. The historical vanilla SFT aggregate is 73.29 ± 0.08; Table 11 retains its seed-42 score of 73.21.

FinQA. We evaluate on the 1,147 FinQA test questions (1,127 numeric and 20 yes/no) in thinking mode with temperature 0.5, top-p 0.9, and up to 32,768 new tokens. Qwen3.6-27B, decoding at temperature 0, judges the full generated text, including the reasoning trace with its delimiters removed, against the gold answer, and a rule-based scorer with 1% relative tolerance serves as a cross-check. We report judge accuracy over all 1,147 questions, including nine duplicate items of the original split.

Safety refusal. We measure refusal of harmful requests on an adjusted version of AIR-Bench 2024 (Zeng et al., 2025). We keep only the risk categories that the usage policies of at least nine major AI developers prohibit, and only the base prompts, discarding the uncommon-dialect and authority-endorsement mutations, which are known to be noisy. The remaining 1,405 prompts carry no jailbreak-style wrapping, so the score reflects refusal of plainly stated harmful requests. Each prompt is answered with four generation seeds (5,620 responses), at temperature 0.6, top-p 0.95, top-k 20, and a 36,864-token generation limit.

Qwen3.6-27B applies one global judge prompt to all requests. Since the requests are harmful by construction, the judge assesses the response's willingness to assist, independently of its substantive usefulness. This measures refusal on a fixed harmful-request set. Only refusals motivated by safety count as refusals: a response that declines for lack of context or asks for clarification counts as

willingness to assist. The safety refusal score is the percentage of responses judged as refusals, averaged over the four responses of each prompt and then over prompts; higher is better.

Hallucination. We adapt the Persona Vectors protocol (Chen et al., 2025) for the hallucination trait: 20 questions, each sampled 10 times with engine seed 42, temperature 1, top-p 1, and no top-k restriction. The generation limit is raised from 1,000 to 16,386 tokens to accommodate reasoning. Only the text after the final `</think>` delimiter is scored; an unclosed reasoning block produces an empty scored response. GPT-4.1-mini (2025-04-14), at temperature 0, supplies top-20 token log-probabilities. Each response's score is the probability-weighted mean of numeric tokens from 0 to 100; responses with numeric probability mass below 0.25 receive no score. We report the mean hallucination score, where lower is better.

Seeds. Every model is trained once with seed 42, except in the load sweep of Section 3.4, which is a separate single-seed run with training seed 43. Figure 4 reports these models, each evaluated with three evaluation seeds (42, 43, and 44) on both MedQA and hallucination; its values therefore differ from Table 1, which uses the original reference measurements described above and seed-42 models for the other runs.

Repeated evaluation. We re-evaluate the fixed vanilla and targeted-hallucination checkpoints three times under each domain and downstream objective (Table 9). MedQA and hallucination use evaluation seeds 42–44; FinQA uses three repeated evaluations. For ResT, the original evaluation is combined with two repeats; for the MedQA vanilla references, a fresh seed-42 evaluation replaces the earlier aggregate. Tables 1 and 2 retain the original measurements; the table below summarizes the repeated evaluations.

The hallucination reduction persists in all four settings: the mean difference from vanilla ranges from 30.98 to 41.63 points. All targeted models remain below the re-evaluated initial model's hallucination score of 35.15. Their mean task accuracy is 0.63–0.82 points lower than vanilla on MedQA and 1.25–1.37 points higher on FinQA. Every targeted-model evaluation yields 200 nonempty scored answers with completed reasoning blocks.

Table 9: Repeated evaluation of vanilla and targeted-hallucination models. Entries are means with sample standard deviations over three evaluations of the same checkpoint. Task accuracy is higher-is-better; hallucination is lower-is-better.

	Task accuracy ↑		Hallucination ↓	
Setting	Vanilla	Targeted ResT	Vanilla	Targeted ResT
MedQA OPD	$79.71_{\pm 0.84}$	$79.08_{\pm 0.44}$	$55.21_{\pm 1.37}$	$13.58_{\pm 2.32}$
MedQA OPSD	$75.81_{\pm 0.48}$	$74.99_{\pm 0.37}$	$52.25_{\pm 0.64}$	$14.31_{\pm 0.93}$
FinQA OPD	$71.29_{\pm 0.55}$	$72.54_{\pm 0.26}$	$46.61_{\pm 0.56}$	$8.85_{\pm 0.99}$
FinQA OPSD	$72.94_{\pm 0.31}$	$74.31_{\pm 0.98}$	$46.23_{\pm 1.55}$	$15.25_{\pm 0.88}$

D.6 Assets and licenses

Table 10 lists the models, datasets, and code used in this work. GPT-4.1 and GPT-4.1-mini are accessed through the OpenAI API.

Table 10: Models, datasets, and code used in this work, with their sources and licenses.

Asset (use)	Source	License
Models		
Qwen3-4B (Yang et al., 2025) (student)	https://huggingface.co/Qwen/Qwen3-4B	Apache-2.0
Qwen3-4B-Base (Yang et al., 2025) (generic teacher)	https://huggingface.co/Qwen/Qwen3-4B-Base	Apache-2.0
Qwen3-32B (Yang et al., 2025) (task teacher, data generation)	https://huggingface.co/Qwen/Qwen3-32B	Apache-2.0
Qwen3.6-27B (judge)	https://huggingface.co/Qwen/Qwen3.6-27B	Apache-2.0
Huihui-Qwen3.8-27B-abliterated (misaligned responses)	https://huggingface.co/huihui-ai/Huihui-Qwen3.8-27B-abliterated	Apache-2.0
Datasets		
MedQA (Jin et al., 2021)	https://github.com/jind11/MedQA	MIT
Fino1 FinQA reasoning paths (Qian et al., 2025) (training)	https://huggingface.co/datasets/TheFinAI/Fino1_Reasoning_Path_FinQA	CC-BY-4.0
FinQA (Chen et al., 2021) (test)	https://github.com/czyssrs/FinQA	MIT
AIR-Bench 2024 (Zeng et al., 2025)	https://huggingface.co/datasets/stanford-crfm/air-bench-2024	CC-BY-4.0
HarmChain (Jeung et al., 2025)	https://huggingface.co/datasets/AI-ISL/HarmChain	MIT
Persona Vectors prompts (Chen et al., 2025)	https://github.com/safety-research/persona_vectors	Apache-2.0
Training and inference frameworks		
ms-swift (Zhao et al., 2025)	https://github.com/modelscope/ms-swift	Apache-2.0
verl (Sheng et al., 2025)	https://github.com/volcengine/verl	Apache-2.0
Reference implementations of compared methods		
OPSD (Shenfeld et al., 2026a)	https://github.com/idanshen/Self-Distillation	Not stated
Shadow-FT (Wu et al., 2025)	https://github.com/wutaiqiang/Shadow-FT	Apache-2.0
OSFT (Nayak et al., 2026)	https://github.com/Red-Hat-AI-Innovation-Team/mini_trainer	Apache-2.0
SSR (Huang et al., 2024a)	https://github.com/XMUDeepLIT/SSR	Apache-2.0

E Results under Supervised Fine-Tuning

This appendix reports the SFT counterpart of the main comparison (Table 11); the SFT adapter-construction comparison appears in Figure 3. Vanilla SFT raises MedQA accuracy less than the on-policy objectives (73.21) while showing a similar loss of the protected behaviors (refusal 28.51, hallucination 52.86). For Table 11, response-supervised Safety loads use one pass over a random tenth of the HarmChain rows; Hallucination loads use two passes over their source data. On-policy-distilled loads are shared with Table 1. These fixed schedules differ from the configurations selected for Figure 3.

Table 11: **Downstream SFT on MedQA and FinQA.** Metrics, separate protection settings, and notation follow Table 1. Response and On-policy denote load fitting by response supervision and on-policy distillation, respectively. Response-supervised loads follow the schedule above; on-policy-distilled loads are shared with Table 1. The vanilla SFT reference uses an earlier training recipe (Appendix D).

	Safety refusal		Factual restraint	
Method	Task accuracy↑	Refusal↑	Task accuracy↑	Hallucination↓
Medical (MedQA)				
Initial model	71.80	31.21	71.80	34.00
Vanilla	73.21	28.51	73.21	52.86
Shadow-FT	**76.75**	32.09	**76.75**	40.34
OSFT	71.01	25.64	69.21	60.37
Replay	72.90	31.97	72.98	28.10
ResT (Response): Generic	74.71	32.54	74.39	42.80
ResT (Response): Targeted	73.76	35.20	74.86	33.44
ResT (On-policy): Generic	73.06	31.64	74.00	27.10
ResT (On-policy): Targeted	72.66	**38.50**	74.78	**11.36**
Finance (FinQA)				
Initial model	70.18	31.21	70.18	34.00
Vanilla	69.75	26.90	69.75	53.61
Shadow-FT	**73.76**	26.18	**73.76**	47.81
OSFT	69.40	26.42	68.44	66.14
Replay	69.66	**31.45**	69.22	33.99
ResT (Response): Generic	69.22	30.39	70.79	58.45
ResT (Response): Targeted	70.62	31.43	70.97	46.52
ResT (On-policy): Generic	68.61	29.10	71.49	42.14
ResT (On-policy): Targeted	71.23	29.57	73.32	**30.35**

Retention and task performance. Targeted loads improve the protected metric over vanilla SFT in both domains with either fitting objective. On MedQA, the on-policy-distilled targeted load gives the highest refusal and lowest hallucination score in the table. Its refusal gain comes with a small task-accuracy cost relative to vanilla, while its hallucination gain accompanies higher task accuracy. On FinQA, targeted loads improve task accuracy and the protected metric over vanilla with both fitting objectives. On-policy fitting gives stronger hallucination protection, whereas response supervision gives higher refusal. Shadow-FT achieves the highest task accuracy in both domains, and Replay gives the highest FinQA refusal. The SFT results extend the retention benefit to learning from fixed demonstrations, with the strongest task–retention combination depending on the domain and fitting objective.