UNDERSTANDING NEGATIVE LEARNING IN LARGE LANGUAGE MODEL FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Negative learning has never been more important than it is today, particularly in large language model (LLM) fine-tuning, where it underpins machine unlearning, preference- and reward-based optimization, and exploratory data generation. Yet its functional role remains a subject of ongoing debate, with existing work variously characterizing it as forgetting, regularization, or target shaping. However, analytical support focuses primarily on the single-update scale, while empirical results at the trajectory scale are typically obtained under negative-only training and may not extend to composite objectives in which it interacts with other components. To develop a deeper understanding, we introduce a gradient-based endpoint probing framework that can examine negative updates separately throughout training. Our analysis reveals that negative learning plays different roles across algorithms and training phases. For example, in off-policy DPO, it acts more like a regularizer early in training and shows evidence of target shaping only toward out-of-distribution endpoints and only in later phases. In on-policy PPO and GRPO, by contrast, negative updates offset endpoint-aligned positive learning without stable target shaping, consistent with a role in sustaining exploration. Guided by these findings, we develop a family of stage-aware methods that modulate the strength of negative updates throughout training, improving fine-tuning performance across tasks.

1 INTRODUCTION

Negative learning (Welleck et al., 2019), broadly defined as reducing the likelihood of selected outputs, is now widely used in large language model (LLM) fine-tuning. In machine unlearning (Yao et al., 2024; Liu et al., 2025; Wang et al., 2025d), it suppresses sensitive, harmful, or copyrighted content; in preference and policy optimization (Ziegler et al., 2019; Rafailov et al., 2023; Zhang et al., 2024; Shao et al., 2024), it penalizes rejected responses or negative-advantage actions; and in controllable generation (Lu et al., 2022; Chen et al., 2026), it discourages undesirable or previously sampled responses to promote further exploration. Before its adoption in LLMs, related forms of negative supervision were studied in complementary-label, noisy-label, and partial-label learning (Ishida et al., 2017; Han et al., 2020; Lv et al., 2020) to improve robustness and resolve target ambiguity.

Despite its widespread use, the mechanisms underlying negative learning remain considerably less understood. One prominent interpretation in LLM fine-tuning is *implicit target shaping*. Zhang et al. (2024) and Zhu et al. (2025) empirically show that, by suppressing given targets, negative-only training can shift learning toward more preferred alternatives. Ren & Sutherland (2025) analytically explain this effect by showing how a single negative-gradient update redistributes likelihood according to the model prior. These accounts accurately characterize negative learning when used in isolation. However, in its more common use within preference- and reward-optimization settings (Ziegler et al., 2019; Rafailov et al., 2023), negative learning forms part of a composite objective and interacts with other components. This interaction may alter the target-shaping behavior observed in isolation and cause the role of negative learning to evolve throughout training, leaving it unclear whether target shaping persists beyond a single update and remains stable to serve as a supervisory signal. The picture is further complicated by alternative interpretations of negative learning as targeted forgetting (Liu et al., 2025) or, more classically, as regularization (Han et al., 2020). Consequently, its precise role within LLM fine-tuning remains difficult to isolate and understand.

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.

To better understand negative learning, we need a flexible empirical tool that can disentangle its contributions from those of other components and characterize its role at the training-trajectory scale. This is challenging because many fine-tuning objectives lack explicit learning targets against which individual components can be assessed. To address this issue, we introduce an *endpoint probing* tool that treats responses generated by the final checkpoint as empirically achieved targets. Across training checkpoints, we quantify from a gradient perspective how strongly the current objective update supports these endpoints. Gradient additivity then allows each objective component to be probed separately at the same model state. This construction builds upon established uses of gradient inner products (Pruthi et al., 2020; Bengio et al., 2020), including in prior work on LLMs (Xia et al., 2024; Wang et al., 2025e; Chang et al., 2025). We further examine the agreement between endpoint probing estimates and actual one-step changes in endpoint likelihood in Section 3 and derive a local approximation-error bound in Appendix D, supporting the use of endpoint probing in our setting.

We deliberately keep our probing simple and do not overly emphasize its methodological novelty. Rather, we tailor this established technique to our setting, where it can reliably isolate and track component-wise effects that prior likelihood-based analyses (Ren & Sutherland, 2025) cannot distinguish. Our contribution lies in the empirical characterization of negative learning enabled by this framework. We first confirm that negative-only learning can shape stable implicit targets, and then examine its role within composite objectives for preference and reward optimization.

We find that the role of negative learning within composite objectives varies with the optimization paradigm, endpoint distribution, and training phase. In off-policy preference optimization, exemplified by *direct preference optimization* (DPO) (Rafailov et al., 2023), negative learning shapes targets for semantically related out-of-distribution prompts, but not for in-distribution prompts. This contrast does not imply a DPO deficiency. Rather, it suggests that negative target shaping may support preference generalization beyond the training distribution (Im & Li, 2024). Moreover, this behavior is phase-dependent, with negative learning acting more like a regularizer early in training and shaping the endpoints only in later phases. By contrast, in on-policy reward optimization, represented by *proximal policy optimization* (PPO) (Ziegler et al., 2019) and *group relative policy optimization* (GRPO) (Shao et al., 2024), we observe no stable target-shaping behavior from negative learning. Instead, it offsets positive learning, consistent with a role in sustaining exploration, a central mechanism of reinforcement learning, toward alternative, potentially higher-reward responses.

These findings complement existing accounts of negative learning in LLM fine-tuning and may motivate further improvements to methods that rely on it. As a concrete demonstration, we develop stage-aware fine-tuning methods that adjust the relative strengths of positive and negative learning according to their evolving roles. These variants outperformed their corresponding baselines in preference alignment, reasoning, and tool use, showing how our findings can guide method design.

2 Necessary Background

We first introduce the background that motivates negative learning and its use in composite LLM fine-tuning objectives, and then revisit existing explanations of its functional role and their limitations. We keep the discussion concise and defer further details to Appendix A.

2.1 Positive, Negative, and Composite Learning

Consider an LLM with conditional distribution $\pi(y \mid x; \boldsymbol{\theta})$ and a training set $\mathcal{D}^+ = \{z^+ = (x, y^+)\}$, with x the prompt, y^+ the target, and $\boldsymbol{\theta}$ the parameters. Supervised fine-tuning typically enables the model to learn from $\mathcal{D}^+$ by minimizing $\mathcal{L}^+(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}^+} \log \pi(y^+ \mid x; \boldsymbol{\theta})$, where $\hat{\mathbb{E}}$ denotes the empirical expectation. We refer to this likelihood-increasing process as *positive learning*. However, modern fine-tuning often involves more complex objectives that require the model to move away from undesirable data, typically through *negative learning* (Yao et al., 2024; Liu et al., 2025). Specifically, given a negative training set $\mathcal{D}^- = \{z^- = (x, y^-)\}$, a canonical objective is $\mathcal{L}^-(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}^-} \log \pi(y^- \mid x; \boldsymbol{\theta})$, whose minimization decreases the likelihood of the selected outputs.

More practical LLM fine-tuning methods often require learning from desirable outputs while moving away from undesirable ones, thereby combining positive and negative learning. Objectives used in preference and reward optimization, such as DPO and PPO (Rafailov et al., 2023; Ziegler et al.,

2019), are prominent examples of what we call *composite objectives*, written as

$$\mathcal{L}_{\text{comp}}(\boldsymbol{\theta}) = \mathcal{L}^{+}(\boldsymbol{\theta}) + \mathcal{L}^{-}(\boldsymbol{\theta}). \tag{1}$$

This paper broadly defines positive and negative learning as objective components whose minimization increases and decreases the likelihood of selected outputs, respectively. Their specific formulations may incorporate data reweighting and other regularization strategies (Wang et al., 2025e; Fan et al., 2025). Please refer to Appendix A for detailed formulations of DPO, PPO, and others.

2.2 Understanding Negative Learning

Our current understanding of positive learning is comparatively mature. Classical machine learning characterizes it through likelihood-based estimation, empirical risk minimization, and theories of learnability and generalization (Bishop, 2006; Mohri et al., 2012). Recent work in LLMs has further studied how positive learning evolves during fine-tuning, linking its dynamics to memorization and knowledge acquisition and examining how task diversity, model scale, and data quality shape reasoning, instruction following, and safety (Wei et al., 2021; Sanh et al., 2021; Zhou et al., 2023; Gekhman et al., 2024; Choi et al., 2024; Pang et al., 2025; Wang et al., 2025b).

By contrast, negative learning remains less understood because its canonical objective is unbounded below and specifies no learning target, complicating theoretical analyses of learnability and generalization. Consequently, empirical studies of LLM fine-tuning have focused on observed training behavior, showing that negative-only training can induce forgetting of negative examples (Zhang et al., 2024) and shift the model toward implicit targets (Zhu et al., 2025). Ren & Sutherland (2025) analytically characterize the latter behavior as probability squeezing, whereby a single negative update redistributes probability mass from the suppressed target toward alternatives favored under the current model distribution. This target-shaping account has since become influential, guiding subsequent analyses and method design in LLM fine-tuning (Wang et al., 2025g; Deng et al., 2025).

We recognize that target shaping characterizes negative learning when studied in isolation, as we further verified in Section 4.1, or analyzed locally at a single update. However, extrapolating this local effect to explain the broader behavior of composite objectives in Eq. (1), as Ren & Sutherland (2025) imply for DPO and PPO, can be misleading. As the model state evolves, interactions between negative learning and other components, particularly positive learning, may alter this behavior throughout training (Wang et al., 2025f; Razin et al., 2025; Yuan et al., 2025). Local target shaping therefore does not necessarily imply that negative learning shapes a stable target over the training trajectory. Such an incomplete account can obscure its actual contribution during joint training and misguide method design by treating a transient local effect as a persistent supervisory signal.

3 Probing Mechanism

Motivated by this gap, we move beyond local characterization to trajectory-level verification, asking whether and when locally induced target shaping persists as a stable supervisory signal. This question is challenging because the implicit targets shaped by negative learning are not directly observable. Moreover, existing diagnostics, such as those that track likelihood changes (Ren & Sutherland, 2025; Razin et al., 2025; Yan et al., 2025), capture the joint effects of positive and negative learning within composite objectives and therefore cannot isolate their respective contributions.

3.1 Endpoint Construction

To this end, we introduce *endpoint probing* to examine whether locally induced target shaping persists throughout LLM fine-tuning. If a target consistently guides training, we expect its influence to be reflected in the final model responses. We therefore use responses generated by the final checkpoint as empirical surrogates for these otherwise unobservable targets and refer to them as *endpoints*. Endpoint probing looks back along the training trajectory and, at each intermediate checkpoint, measures whether the current update moves the model toward or away from these endpoints. Tracking this effect across checkpoints provides evidence of whether target shaping toward these endpoints persists as a stable supervisory signal and when it emerges or disappears during training.

Concretely, we construct an endpoint dataset $\mathcal{D}_{\text{end}} = \{(x', y_{\text{end}})\}$, where x' is a held-out prompt and y_{end} is the corresponding response generated by the final model, parameterized by $\boldsymbol{\theta}^{(T)}$ after

T fine-tuning steps. The held-out prompts may be drawn from either the training distribution or a semantically related out-of-distribution task, allowing us to examine target shaping both within and beyond the training distribution. We measure the model fit to the endpoints using the endpoint loss $\mathcal{L}_{\text{end}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\text{end}}} \log \pi(y_{\text{end}} \mid x'; \boldsymbol{\theta})$. Then, to probe the effect of an objective $\mathcal{L}$ at an intermediate checkpoint $\boldsymbol{\theta}^{(t)}$, we evaluate how this loss would change after one gradient update on $\mathcal{L}$:

$$\Delta(\mathcal{L}; \boldsymbol{\theta}^{(t)}) := \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)}) - \mathcal{L}_{\text{end}}\big(\boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)})\big), \tag{2}$$

where $\eta > 0$ is the learning rate. A positive Δ indicates that the update moves the model toward the endpoints, whereas a negative value indicates the opposite.

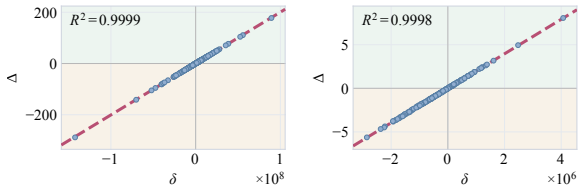
3.2 Gradient-based Estimation

Since $\mathcal{L}_{\text{end}}$ is nonlinear, the exact change in Eq. (2) for $\mathcal{L} = \mathcal{L}_{\text{comp}}$ cannot generally be decomposed to isolate the contributions of positive and negative learning. Fortunately, gradients are additive, allowing component contributions to be separated under a first-order approximation. This motivates the first-order expansion around $\boldsymbol{\theta}^{(t)}$, yielding $\Delta \approx \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)})^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)})$. Omitting the positive scaling factor η, we define the *endpoint-probing score* as

$$\delta(\mathcal{L}; \boldsymbol{\theta}^{(t)}) := \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)})^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}). \tag{3}$$

As in Eq. (2), $\delta > 0$ predicts an update toward the endpoints, and $\delta < 0$ the opposite.

We derive the first-order approximation and its local error bound in Appendix D.1. Under local smoothness assumptions, a small learning rate and gradient clipping help control the approximation error, both being standard practices in LLM fine-tuning (Biderman et al., 2023). Thus, $\eta\delta$ provides a first-order approximation to Δ, with δ serving as a learning-rate-independent proxy. To further validate this approximation empirically, we compared δ with the exact one-step change Δ across training checkpoints. Figure 1 shows a strong positive correlation for both DPO and PPO, indicating that endpoint probing closely tracks the actual update effect.



(a) DPO on Qwen3-1.7B (b) PPO on Pythia-2.8B

Figure 1: **Likelihood vs. Gradient Probing.** The endpoint-probing score δ strongly correlates with the exact one-step reduction in endpoint loss Δ across checkpoints, with squared Pearson correlation $R^2 \approx 1$. This supports using δ as a proxy for the update effect. Details are provided in Appendix B.6.

4 Probing Results

We applied endpoint probing to NPO, DPO, PPO, and GRPO (Ziegler et al., 2019; Zhang et al., 2024; Shao et al., 2024) on unlearning, preference alignment, and tool-use reasoning, with Pythia, Qwen, Llama, and Gemma families (Biderman et al., 2023; Grattafiori et al., 2024; Yang et al., 2025; Gemma Team, 2026). Results were averaged over three independent trials, with more details in Appendix B.

4.1 Negative-only Optimization

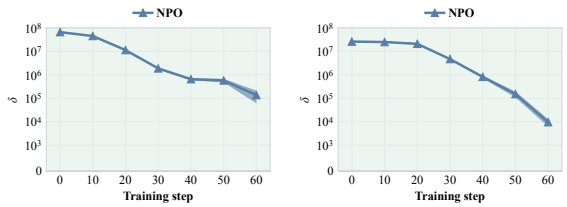
We begin with *negative preference optimization* (NPO) (Zhang et al., 2024), a representative negative-only method for LLM unlearning. NPO learns only from undesirable prompt–response pairs, telling the LLM what to avoid without specifying a preferred alternative. It was developed to remove private or harmful information while preserving general model utility, and has since become a widely used baseline in LLM unlearning and inspired subsequent work (Wang et al., 2025f;e; Fan et al., 2025).

The original NPO objective is relatively complex, so we use its more interpretable gradient-equivalent form. Given a forget set $\mathcal{D}_{\text{npo}} = \{z^- = (x, y^-)\}$, NPO is gradient-equivalent to minimizing

$$\mathcal{L}^-_{\text{npo}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}_{\text{npo}}}\big[\omega_{\text{npo}}(z^-; \boldsymbol{\theta}|_{\text{detach}}) \log \pi(y^- \mid x; \boldsymbol{\theta})\big]. \tag{4}$$

The weight ω_{npo} depends on the current model $\boldsymbol{\theta}$ and negative example z^- but is treated as constant during differentiation, as indicated by $|_{\text{detach}}$. It downweights undesirable responses whose likelihood has already been substantially reduced relative to the reference model, thereby limiting excessive negative updates. The original objective and detailed derivation are provided in Appendix A.1.

We probed NPO on the TOFU `forget05` task (Maini et al., 2024) using Llama3.2-1B and 3B (Grattafiori et al., 2024). Training lasted 10 epochs (70 steps), with checkpoints saved every 10 steps. This interval balances temporal resolution and computational efficiency, while Appendix D.2 gives conditions under which checkpoint scores remain informative between observations. Figure 2 presents the results. Although NPO specifies only what the LLM should avoid, its positive probing scores across checkpoints provide evidence of stable target shaping, consistent with findings on negative-only learning in LLM reasoning (Zhu et al., 2025).



(a) Llama3.2-1B (b) Llama3.2-3B

Figure 2: **NPO Probing.** We report mean endpoint-probing scores for NPO on TOFU `forget05` using Llama3.2-1B and 3B, with shading indicating the standard deviation.

4.2 Off-policy Preference Optimization

Among off-policy methods with composite objectives, DPO (Rafailov et al., 2023) is a widely used representative. It is typically used for preference optimization, learning from fixed preference pairs by increasing the relative likelihood of preferred responses over rejected ones, without online sampling or explicit reward-model training. Although the original DPO objective does not directly take the form of Eq. (1), Appendix A.2 derives the following gradient-equivalent decomposition:

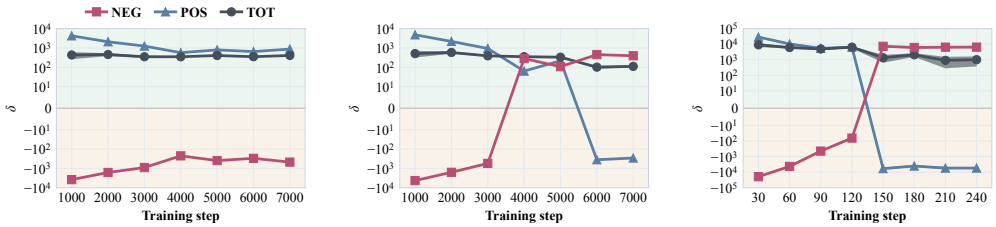
$$\mathcal{L}_{\text{dpo}}(\boldsymbol{\theta}) = \underbrace{\hat{\mathbb{E}}_{\mathcal{D}_{\text{dpo}}}\big[-\omega_{\text{dpo}}(z;\boldsymbol{\theta}|_{\text{detach}})\log\pi(y^+|x;\boldsymbol{\theta})\big]}_{\mathcal{L}^+_{\text{dpo}}(\boldsymbol{\theta})} + \underbrace{\hat{\mathbb{E}}_{\mathcal{D}_{\text{dpo}}}\big[\omega_{\text{dpo}}(z;\boldsymbol{\theta}|_{\text{detach}})\log\pi(y^-|x;\boldsymbol{\theta})\big]}_{\mathcal{L}^-_{\text{dpo}}(\boldsymbol{\theta})}. \tag{5}$$

Here, $\mathcal{D}_{\text{dpo}} = \{z = (x, y^+, y^-)\}$ is a dataset of preference pairs, where y^+ is preferred over y^-. The weight ω_{dpo} emphasizes preference pairs with insufficient margins and downweights those already well learned. Accordingly, $\mathcal{L}^+_{\text{dpo}}$ and $\mathcal{L}^-_{\text{dpo}}$ represent positive and negative learning, respectively.

We probed DPO trained on UltraFeedback (Cui et al., 2024) using Pythia-2.8B (Biderman et al., 2023) as the base model. Following common practice (Tunstall et al., 2023), we performed supervised fine-tuning before DPO and tuned the hyper-parameters for stable training. We constructed the endpoint dataset by pairing held-out, in-distribution prompts with responses greedily decoded from the final DPO checkpoint. Figure 3a presents the results. $\mathcal{L}_{\text{dpo}}$ yields positive scores across the probed checkpoints, indicating that DPO updates remain directed toward the endpoints and suggesting a relatively stable target. This pattern resembles supervised-like optimization dynamics, consistent with prior theoretical connections between supervised learning and DPO (Wang et al., 2025a). However, component-wise probing shows no clear target-shaping effect from negative learning on these endpoints, while positive learning primarily drives the model toward them.

Prior studies have reported that DPO improves generalization to related but unseen tasks (Ranaldi & Freitas, 2024; Wang et al., 2025c), motivating us to examine its target-shaping dynamics for out-of-distribution endpoints. We retained UltraFeedback for DPO but constructed endpoints from HH-RLHF-helpfulness prompts (Bai et al., 2022), which cover similar helpfulness tasks but form a distinct prompt set. The probing results in Figure 3b reveal intriguing dynamics. First, the full DPO objective remains positively aligned with the out-of-distribution endpoints across checkpoints, suggesting stable supervised-like dynamics beyond the training distribution. More importantly, the roles of its components evolve. Positive learning initially drives the model toward the endpoints while negative learning moves against them. Their roles later reverse, with negative learning shaping the endpoints and positive learning moving in the opposite direction.

Figure 3c further shows a similar transition with Qwen3-1.7B (Yang et al., 2025), where negative learning exhibits sustained target shaping only in the later phase. This suggests that this behavior may extend across model families. Negative learning shows no clear target-shaping effect on in-

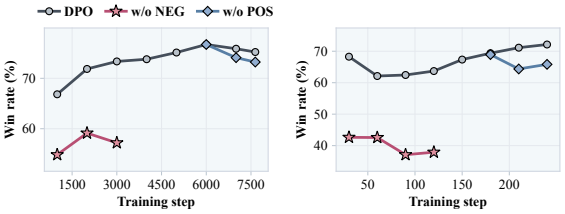


(a) ID, Pythia (b) OOD, Pythia (c) OOD, Qwen

Figure 3: **DPO Probing.** We report endpoint-probing scores for in-distribution (ID) and out-of-distribution (OOD) endpoints using Pythia-2.8B (Pythia) and Qwen3-1.7B (Qwen).

distribution endpoints but shapes out-of-distribution endpoints later in training. This contrast does not by itself indicate better or worse performance. Instead, it suggests that negative learning may contribute to the generalization advantage of DPO over supervised fine-tuning. More specifically, its target-shaping effect on unseen prompts offers a potential mechanism for this generalization benefit.

What does a negative endpoint-probing score imply? By definition, it predicts that the component moves the model away from the endpoints. However, its effect on task performance, typically evaluated by win rate, remains ambiguous. It may harm performance by interfering with fine-tuning (Yu et al., 2020) or benefit it by regularizing training (Han et al., 2020). To distinguish these possibilities, we separately removed negative learning early and positive learning later, when their respective scores are negative. Figure 4 shows that both interventions reduce win rates, consistent with a beneficial regularization role rather than merely harmful interference. Specifically, negative learning constrains positive-driven target shaping early, while positive learning constrains negative-driven shaping later.



(a) Pythia-2.8B (b) Qwen3-1.7B

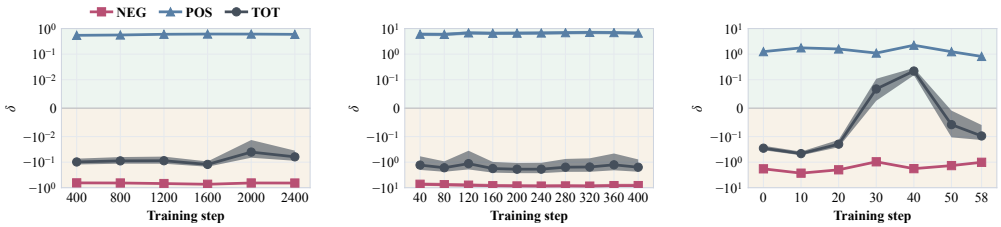
Figure 4: **DPO Win-Rate Ablation.** We report win rates after separately removing negative learning early and positive learning later, when each has a negative endpoint-probing score. These ablations suggest that components with negative δ can remain beneficial.

Take-Home Message

- The full DPO objective remains aligned with the final endpoints, indicating stable, supervised-like optimization with implicit learning targets.
- The target-shaping effect of negative learning is observed only for out-of-distribution endpoints, while in-distribution target shaping is primarily driven by positive learning.
- For out-of-distribution endpoints, target shaping is phase-dependent, with positive learning dominating early and negative learning dominating later.
- On out-of-distribution endpoints, positive and negative learning play complementary roles, alternating between target shaping and beneficial regularization-like effects.

4.3 On-Policy Reward Optimization

We next consider PPO (Ziegler et al., 2019) and GRPO (Shao et al., 2024), two on-policy reinforcement learning methods. Both seek to maximize expected reward using responses sampled and evaluated during training. In practice, this reward-maximization problem is optimized through a clipped surrogate, which encourages higher likelihood for positive-advantage outputs and lower likelihood for negative-advantage outputs. The surrogate can therefore be decomposed into positive and negative learning. We begin with PPO, which we study in the context of preference alignment.



(a) PPO, Pythia (b) PPO, Qwen (c) GRPO, Gemma

Figure 5: **PPO and GRPO Probing.** We report probing scores using out-of-distribution endpoints for PPO with Pythia-2.8B (Pythia) and Qwen3-1.7B (Qwen), and held-out endpoints for GRPO with Gemma4-E4B (Gemma).

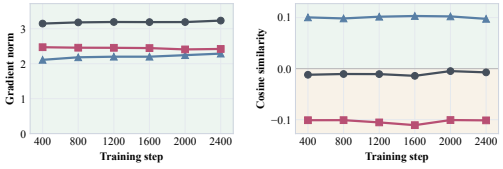
For our component-wise analysis, its clipped policy surrogate admits the following decomposition:

$$\mathcal{L}_{\text{ppo}}(\boldsymbol{\theta}) = \underbrace{\hat{\mathbb{E}}_{\mathcal{D}_{\text{ppo}}}\big[-\text{clip}_{A(z)}\big(r(z;\boldsymbol{\theta})\big)A(z)\mathbb{1}_{\{A(z)>0\}}\big]}_{\mathcal{L}^{+}_{\text{ppo}}(\boldsymbol{\theta})} + \underbrace{\hat{\mathbb{E}}_{\mathcal{D}_{\text{ppo}}}\big[-\text{clip}_{A(z)}\big(r(z;\boldsymbol{\theta})\big)A(z)\mathbb{1}_{\{A(z)<0\}}\big]}_{\mathcal{L}^{-}_{\text{ppo}}(\boldsymbol{\theta})}. \tag{6}$$

PPO assigns token-level advantages to the rollout dataset $\mathcal{D}_{\text{ppo}} = \{z = ((x, y_{<j}), y_j)\}$, where (x, y) is a sampled trajectory, j indexes the tokens in y, and $A(z)$ is the advantage assigned to token y_j. The density ratio is $r(z;\boldsymbol{\theta}) = \frac{\pi(y_j|x,y_{<j};\boldsymbol{\theta})}{\pi(y_j|x,y_{<j};\boldsymbol{\theta}_{\text{old}})}$, where $\boldsymbol{\theta}_{\text{old}}$ parameterizes the rollout policy from which $\mathcal{D}_{\text{ppo}}$ is sampled. Although PPO uses a clipped importance-weighted surrogate rather than the direct log-likelihood weighting in Eqs. (4) and (5), $\mathcal{L}^{+}_{\text{ppo}}$ and $\mathcal{L}^{-}_{\text{ppo}}$ still implement positive and negative learning. Please refer to Appendix A.3 for a more detailed discussion.

As in our DPO analysis, we used Pythia-2.8B and Qwen3-1.7B as base models and first performed supervised fine-tuning. We then trained the models with PPO on UltraFeedback, using the well tuned hyperparameters detailed in Appendix B. Online rollouts were scored by ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024). We constructed the endpoint datasets by pairing held-out, out-of-distribution HH-RLHF-helpfulness prompts with responses greedily decoded from the final PPO checkpoints.

The probing results can be found in Figures 5a–5b. Unlike DPO, PPO exhibits slightly negative, near-zero probing scores for both models, showing no stable endpoint-directed dynamics. Such scores may reflect either weak updates or nearly orthogonal gradients. To distinguish these cases, we report the gradient norms of the PPO policy surrogate and its components, together with their cosine similarities with the endpoint gradient, using Pythia-2.8B. Figure 6 shows that the full-surrogate gradient has a non-negligible norm, while its cosine similarity with the endpoint gradient remains close to zero and slightly negative. Thus, PPO gradients are mostly orthogonal to, but weakly opposed to, the endpoint gradients. This pattern is consistent with exploration beyond the final responses, although endpoint alignment alone does not establish whether these updates discover better alternatives.



(a) Gradient Norm (b) Cosine Similarity

Figure 6: **Gradient Norms and Cosine Similarity.** Using Pythia-2.8B under the same setup as Figure 5a, we report gradient norms for the full PPO policy surrogate and its components, together with their cosine similarities with the endpoint gradient.

We next examine the roles of positive and negative learning. Across both models, the probing score of $\mathcal{L}^{+}_{\text{ppo}}$ remains positive, indicating that updates from positive-advantage tokens move the model toward the final endpoints despite the continually evolving rollout dataset $\mathcal{D}_{\text{ppo}}$. By contrast, the score of $\mathcal{L}^{-}_{\text{ppo}}$ remains negative and slightly outweighs the positive component, resulting in a slightly negative score for the full PPO surrogate. Unlike in DPO, where negative learning eventually helps shape the endpoints, negative learning in PPO shows no stable target shaping at the trajectory scale and instead appears to sustain exploration over alternative responses. This finding complements the

local account of target shaping through probability redistribution under individual negative updates, showing that the same local mechanism can play a different role across fine-tuning methods.

To examine whether this behavior also arises in other on-policy settings, we studied GRPO on reasoning and tool use. Unlike token-level PPO, GRPO assigns each sampled response a group-relative advantage obtained by normalizing rewards across responses to the same prompt. We trained Gemma4-E4B with GRPO on Hermes Function-Calling v1 (Teknium et al., 2024), with details provided in Appendix B.4. Figure 5c shows a pattern similar to PPO. The full-surrogate score fluctuates around zero and is occasionally positive, but does not remain aligned with the final endpoints. The positive component is endpoint-aligned, whereas the negative component primarily offsets it. This suggests that the observed behavior extends beyond a single method or task.

Take-Home Message

- Across the tasks studied, PPO and GRPO exhibit stable but opposing component-wise endpoint alignments, while the full-surrogate alignment may vary across settings.
- Positive learning on positive-advantage tokens or responses consistently moves the model toward the endpoints despite evolving rollout data.
- Negative learning counteracts this endpoint-directed tendency rather than consistently shaping the endpoints, consistent with a role in sustaining exploration over alternative responses.

5 Controlled Negative Learning

Our analyses reveal that the benefits of negative learning depend on both the involved method and the training phase. Negative learning shifts from a regularization-like role to target shaping for out-of-distribution endpoints during DPO, whereas, in PPO and GRPO, it consistently counteracts endpoint-aligned positive learning, suggesting a role in sustaining exploration. These distinct dynamics suggest that negative learning should not be treated as uniformly beneficial or harmful. Instead, its influence could be coordinated with positive learning according to its method- and phase-specific role. Motivated by this finding, we introduce a family of *controlled optimization* methods that adjust the relative strength of negative learning throughout training. The methods share a common principle but instantiate it differently for off-policy and on-policy optimization.

5.1 Controlled DPO

For the out-of-distribution endpoints studied, positive and negative learning exchange their target-shaping and regularization-like roles during DPO. This role exchange suggests placing greater relative emphasis on the component driving target shaping at each stage, while adjusting the regularization-like influence of the other component. We therefore introduce controlled DPO (cDPO):

$$\mathcal{L}_{\mathrm{cdpo}}(\boldsymbol{\theta}) = 2(1-\lambda_t)\mathcal{L}^{+}_{\mathrm{dpo}}(\boldsymbol{\theta}) + 2\lambda_t\mathcal{L}^{-}_{\mathrm{dpo}}(\boldsymbol{\theta}). \tag{7}$$

λ_t balances positive and negative learning, with $\lambda_t = 0$, $1/2$, and 1 recovering positive-only learning, standard DPO, and negative-only learning, respectively. We set $\lambda_t = \max\left(\min(\frac{t-t_1}{t_2-t_1}, 1), 0\right)$, where $t_1 < t_2$ define a linear transition from positive to negative learning.

5.2 Controlled PPO and GRPO

Unlike in DPO, positive and negative learning exhibit stable but opposing endpoint alignments in the on-policy PPO and GRPO settings studied. Positive learning moves the model toward the final endpoints, suggesting that positive-advantage responses provide useful exploitation targets, while negative learning counteracts this direction. We examine whether controlling the strength of this counteracting effect can improve fine-tuning performance and therefore introduce controlled PPO (cPPO) and controlled GRPO (cGRPO), which share the following general form:

$$\mathcal{L}_{\mathrm{cA}}(\boldsymbol{\theta}) = \mathcal{L}^{+}_{\mathsf{A}}(\boldsymbol{\theta}) + \lambda_t\mathcal{L}^{-}_{\mathsf{A}}(\boldsymbol{\theta}), \qquad \mathsf{A} \in \{\mathrm{ppo}, \mathrm{grpo}\}. \tag{8}$$

λ_t controls the strength of negative learning, with $\lambda_t = 1$ recovering the standard policy surrogate and $\lambda_t < 1$ and $\lambda_t > 1$ attenuating and strengthening negative learning, respectively. Other terms



(a) DPO vs. cDPO (b) PPO vs. cPPO (c) GRPO vs. cGRPO

Figure 7: **Controlled Variants.** Comparison of DPO, PPO, GRPO, and their controlled variants across models and tasks. Markers indicate mean test scores across three independent training seeds, and error bars show $\pm$ one standard deviation across seeds. Metrics are AlpacaEval win rate against the corresponding supervised fine-tuning baseline for DPO and PPO, and mean programmatic reward (multiplied by 100), task-success rate (%), and pass@1 (%) for GRPO on function calling, τ^2 Retail, and MATH500, respectively.

remain unchanged. We design three distinct scheduling rules for λ_t to illustrate possible ways to coordinate positive and negative learning. Periodic attenuation periodically weakens negative learning, allowing positive updates to more strongly reinforce currently high-reward responses. Constant scaling maintains negative learning at a fixed reduced or enhanced strength, aiming to favor exploitation or exploration, respectively. Adaptive strengthening, designed specifically for GRPO, increases the strength of negative learning as the fraction of fully successful rollout groups rises, aiming to encourage further exploration. Please refer to Appendix B.8 for more details.

5.3 Experimental Results

We evaluated cDPO, cPPO, and cGRPO against their standard counterparts across preference alignment, reasoning, and tool use. For preference alignment, we performed supervised fine-tuning on UltraChat-200k (Ding et al., 2023), trained DPO and PPO on UltraFeedback, and evaluated win rates on AlpacaEval (Li et al., 2023). In addition to the models used in our probing analysis, we further included Llama3-8B to examine whether the gains extend to a larger model. For reasoning and tool use, we compared GRPO and cGRPO on Hermes Function-Calling v1 using Gemma4-E4B, τ^2 Retail (Barres et al., 2025) using Qwen3.5-9B, and MATH500 (Lightman et al., 2024) using Qwen3-0.6B. Hyperparameters and checkpoints were selected on held-out validation data.

Figure 7 compares the controlled methods with their corresponding baselines using AlpacaEval win rate for DPO and PPO, and mean programmatic reward, task success rate, and pass@1 for GRPO on function calling, τ^2 Retail, and MATH500, respectively. All experiments were repeated across three independent trials, and we report the mean and standard deviation. The controlled methods achieve higher test scores across the evaluated models and tasks, suggesting that coordinating positive and negative learning can improve fine-tuning performance. This may motivate further research on more dedicated and sophisticated coordination rules. Detailed experimental settings are provided in Appendix B, and additional results can be found in Appendix C.4.

6 Conclusion

Negative learning in LLM fine-tuning is often understood as implicit target shaping, but this account is incomplete. Our findings confirm stable target shaping under negative-only learning, yet reveal method- and phase-dependent roles within composite objectives. In DPO, negative learning shapes out-of-distribution endpoints only later in training and exhibits a regularization-like role early on. In PPO and GRPO, negative learning does not exhibit sustained target shaping toward the final-model endpoints. Instead, it consistently counteracts endpoint-aligned positive learning, consistent with a role in sustaining exploration. Guided by this understanding, our controlled variants improved performance across the evaluated methods and tasks, highlighting the value of coordinating positive and negative learning according to their evolving roles and suggesting a direction for future research.

AI USE STATEMENT

We used LLM-based tools to polish portions of the manuscript, search relevant literature, and assist with mathematical derivations and proof checking. The authors take full responsibility for the accuracy, originality, and integrity of the paper, including its claims, citations, and conclusions.

ETHICS STATEMENT

This work studies negative learning in LLM fine-tuning using established datasets and benchmarks. Potential risks include unintended changes to model behavior and degradation of capabilities outside the intended training scope. However, we do not identify particular ethical concerns specific to this study beyond those generally associated with LLM research and deployment.

REPRODUCIBILITY STATEMENT

We used publicly available datasets and code libraries and provided our implementation in an anonymous repository, which will be released upon acceptance. We provided the hardware environments, experimental configurations, model choices, and hyperparameter settings in Appendix B. Together, these resources and details reflect our best efforts to facilitate reproduction of our results.

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.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Emmanuel Bengio, Joelle Pineau, and Doina Precup. Interference and generalization in temporal difference learning. In *International Conference on Machine Learning*, 2020.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Tyler Chang, Dheeraj Rajagopal, Tolga Bolukbasi, Lucas Dixon, and Ian Tenney. Scalable influence and fact tracing for large language model pretraining. In *International Conference on Learning Representations*, 2025.

Liang Chen, Xueting Han, Qizhou Wang, Bo Han, Jing Bai, Hinrich Schuetze, and Kam-Fai Wong. Eepo: Exploration-enhanced policy optimization via sample-then-forget. In *International Conference on Learning Representations*, 2026.

Hyeong Kyu Choi, Xuefeng Du, and Yixuan Li. Safety-aware fine-tuning of large language models. *arXiv preprint arXiv:2410.10014*, 2024.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled AI feedback. In *International Conference on Machine Learning*, 2024.

Wenlong Deng, Yi Ren, Muchen Li, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. On the effect of negative gradient in group relative deep reinforcement optimization. In *Advances in Neural Information Processing Systems*, 2025.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*, 2023.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, Zico Kolter, Zachary Lipton, and Pratyush Maini. Openunlearning: Accelerating llm unlearning via unified benchmarking of methods and metrics. In *Advances in Neural Information Processing Systems*, 2025.

Yann Dubois, Balazs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. In *Advances in Neural Information Processing Systems*, 2025.

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 *Conference on Empirical Methods in Natural Language Processing*, 2024.

Gemma Team. Gemma 4, 2026.

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

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Sigua: Forgetting may make learning with noisy labels more robust. In *International Conference on Machine Learning*, 2020.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Shawn Im and Yixuan Li. On the generalization of preference learning with dpo. *arXiv preprint arXiv:2408.03459*, 2024.

Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. In *Advances in Neural Information Processing Systems*, 2017.

Kemou Li, Qizhou Wang, Yue Wang, Fengpeng Li, Jun Liu, Bo Han, and Jiantao Zhou. Llm unlearning with llm beliefs. In *International Conference on Learning Representations*, 2026.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, 7(2):181–194, 2025.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*, 2022.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, 2020.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary Chase Lipton, and J Zico Kolter. TOFU: A task of fictitious unlearning for LLMs. In *Conference on Language Modeling*, 2024.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press Cambridge, 2012.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. Kl for a kl: On-policy distillation with control variate baseline. *arXiv preprint arXiv:2605.07865*, 2026.

Jinlong Pang, Na Di, Zhaowei Zhu, Jiaheng Wei, Hao Cheng, Chen Qian, and Yang Liu. Token cleaning: Fine-grained data selection for llm supervised fine-tuning. *arXiv preprint arXiv:2502.01968*, 2025.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, 2020.

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*, 2023.

Leonardo Ranaldi and Andre Freitas. Self-refine instruction-tuning for aligning reasoning in language models. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations*, 2025.

Yi Ren and Danica Sutherland. Learning dynamics of llm finetuning. In *International Conference on Learning Representations*, 2025.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

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

Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. Hermes 3 technical report. *arXiv preprint arXiv:2408.11857*, 2024.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Bo Wang, Qinyuan Cheng, Runyu Peng, Rong Bao, Peiji Li, Qipeng Guo, Linyang Li, Zhiyuan Zeng, Yunhua Zhou, and Xipeng Qiu. Implicit reward as the bridge: A unified view of sft and dpo connections. In *Advances in Neural Information Processing Systems*, 2025a.

Changsheng Wang, Xin Chen, Sijia Liu, and Ke Ding. Breaking memorization barriers in llm code fine-tuning via information bottleneck for improved generalization. *arXiv preprint arXiv:2510.16022*, 2025b.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. Direct judgement preference optimization. In *Conference on Empirical Methods in Natural Language Processing*, 2025c.

Qizhou Wang, Bo Han, Puning Yang, Jianing Zhu, Tongliang Liu, and Masashi Sugiyama. Towards effective evaluations and comparisons for llm unlearning methods. In *International Conference on Learning Representations*, 2025d.

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q Weinberger. Rethinking llm unlearning objectives: A gradient perspective and go beyond. In *International Conference on Learning Representations*, 2025e.

Yue Wang, Qizhou Wang, Feng Liu, Wei Huang, Yali Du, Xiaojiang Du, and Bo Han. Gru: Mitigating the trade-off between unlearning and retention for llms. *arXiv preprint arXiv:2503.09117*, 2025f.

Zecheng Wang, Chunshan Li, Yupeng Zhang, Han Liu, Bingning Wang, Dianhui Chu, and Dianbo Sui. Vpo: Reasoning preferences optimization based on v-usable information. In *Advances in Neural Information Processing Systems*, 2025g.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*, 2024.

Yuzi Yan, Yibo Miao, Jialian Li, Yipin Zhang, Jian Xie, Zhijie Deng, and Dong Yan. 3d-properties: Identifying challenges in dpo and charting a path forward. In *International Conference on Learning Representations*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Advances in Neural Information Processing Systems*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, 2025.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in neural information processing systems*, 2020.

Hui Yuan, Yifan Zeng, Yue Wu, Huazheng Wang, Mengdi Wang, and Liu Leqi. A common pitfall of margin-based language model alignment: Gradient entanglement. In *International Conference on Learning Representations*, 2025.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, 2023.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. In *Advances in Neural Information Processing Systems*, 2025.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A LLM FINE-TUNING METHODS

In this section, we review NPO, DPO, PPO, GRPO, and on-policy distillation (OPD) (Agarwal et al., 2024), and derive the objective forms used in our component-wise analysis.

A.1 NPO

Given a forget set $\mathcal{D}_{\text{npo}} = \{z^- = (x, y^-)\}$, a simple approach to LLM unlearning minimizes the log-likelihood of the forget responses, which is given by $\mathcal{L}^-_{\text{ga}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}_{\text{npo}}}[\log \pi(y^- \mid x; \boldsymbol{\theta})]$. This is equivalent to gradient ascent on the negative log-likelihood. However, this objective is unbounded below and continues to encourage suppression even when the response likelihood is already very small, which can lead to unstable training and catastrophic model collapse (Zhang et al., 2024; Wang et al., 2025e;d).

To mitigate this issue, NPO adopts a DPO-inspired objective using only negative examples, without requiring preferred alternatives. Here, we consider the basic NPO objective without additional retention regularization, which is given by

$$\mathcal{L}^-_{\text{npo}}(\boldsymbol{\theta}) = -\frac{2}{\beta}\hat{\mathbb{E}}_{\mathcal{D}_{\text{npo}}}\Big[\log \sigma\big(-\beta \log \frac{\pi(y^- \mid x; \boldsymbol{\theta})}{\pi(y^- \mid x; \boldsymbol{\theta}^{\text{ref}})}\big)\Big], \tag{9}$$

where $\boldsymbol{\theta}^{\text{ref}}$ denotes the fixed reference-model parameters, $\beta > 0$ controls the weighting behavior, and σ is the sigmoid function. Unlike $\mathcal{L}_{\text{ga}}$, this objective is bounded below by zero. Using $\frac{\mathrm{d}}{\mathrm{d}u}\log\sigma(u) = \sigma(-u)$, differentiation gives $\nabla_{\boldsymbol{\theta}}\mathcal{L}^-_{\text{npo}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}_{\text{npo}}}[\omega_{\text{npo}}(z^-; \boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} \log \pi(y^- \mid x; \boldsymbol{\theta})]$, where

$$\omega_{\text{npo}}(z^-; \boldsymbol{\theta}) = 2\sigma\Big(\beta \log \frac{\pi(y^- \mid x; \boldsymbol{\theta})}{\pi(y^- \mid x; \boldsymbol{\theta}^{\text{ref}})}\Big). \tag{10}$$

The weight decreases toward zero as the response likelihood is reduced relative to the reference model, attenuating further negative updates on already suppressed responses. Treating this weight as constant during differentiation yields the gradient-equivalent form used in Eq. (4):

$$\hat{\mathcal{L}}^-_{\text{npo}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}_{\text{npo}}}\left[\omega_{\text{npo}}(z^-; \boldsymbol{\theta}|_{\text{detach}}) \log \pi(y^- \mid x; \boldsymbol{\theta})\right]. \tag{11}$$

Here, $|_{\text{detach}}$ indicates that ω_{npo} is treated as constant during differentiation. Thus, $\nabla_{\boldsymbol{\theta}}\mathcal{L}^-_{\text{npo}} = \nabla_{\boldsymbol{\theta}}\hat{\mathcal{L}}^-_{\text{npo}}$, although their objective values need not coincide. For simplicity, Eq. (4) in the main text denotes this gradient-equivalent surrogate by $\mathcal{L}^-_{\text{npo}}$.

A.2 DPO

DPO starts from KL-regularized reward maximization, whose optimal policy takes the form of $\pi^*(y \mid x) = \frac{1}{Z(x)}\pi(y \mid x; \boldsymbol{\theta}^{\text{ref}}) \exp\left(\frac{R(x,y)}{\beta}\right)$, where $Z(x)$ is the partition function. Under the Bradley–Terry preference model (Bradley & Terry, 1952), the associated preference probability is given by $p(y^+ \succ y^- \mid x) = \sigma\Big(\beta \log \frac{\pi^*(y^+|x)}{\pi(y^+|x;\boldsymbol{\theta}^{\text{ref}})} - \beta \log \frac{\pi^*(y^-|x)}{\pi(y^-|x;\boldsymbol{\theta}^{\text{ref}})}\Big)$. DPO then minimizes the negative log-likelihood of the observed preferences, yielding its canonical objective:

$$\mathcal{L}_{\text{dpo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\text{dpo}}}\Big[\log \sigma\Big(\beta\Big[\log \frac{\pi(y^+ \mid x; \boldsymbol{\theta})}{\pi(y^+ \mid x; \boldsymbol{\theta}^{\text{ref}})} - \log \frac{\pi(y^- \mid x; \boldsymbol{\theta})}{\pi(y^- \mid x; \boldsymbol{\theta}^{\text{ref}})}\Big]\Big)\Big]. \tag{12}$$

To make the component roles explicit, we present a gradient-equivalent form:

$$\hat{\mathcal{L}}_{\text{dpo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\text{dpo}}}\left[\omega_{\text{dpo}}(z; \boldsymbol{\theta}|_{\text{detach}})\left(\log \pi(y^+ \mid x; \boldsymbol{\theta}) - \log \pi(y^- \mid x; \boldsymbol{\theta})\right)\right], \tag{13}$$

where the preference weight is

$$\omega_{\text{dpo}}(z; \boldsymbol{\theta}) = \beta\sigma\Big(-\beta \log \frac{\pi(y^+ \mid x; \boldsymbol{\theta})}{\pi(y^- \mid x; \boldsymbol{\theta})} + \beta \log \frac{\pi(y^+ \mid x; \boldsymbol{\theta}^{\text{ref}})}{\pi(y^- \mid x; \boldsymbol{\theta}^{\text{ref}})}\Big). \tag{14}$$

Treating the weight as constant during differentiation yields $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{dpo}} = \nabla_{\boldsymbol{\theta}}\hat{\mathcal{L}}_{\text{dpo}}$. Also, Eq. (5) in the main text denotes this gradient-equivalent surrogate by $\mathcal{L}_{\text{dpo}}$. Accordingly, the positive- and negative-learning components are

$$\mathcal{L}^{+}_{\text{dpo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\text{dpo}}}\left[\omega_{\text{dpo}}(z;\boldsymbol{\theta}|_{\text{detach}})\log\pi(y^{+} \mid x;\boldsymbol{\theta})\right], \tag{15}$$

$$\mathcal{L}^{-}_{\text{dpo}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}_{\text{dpo}}}\left[\omega_{\text{dpo}}(z;\boldsymbol{\theta}|_{\text{detach}})\log\pi(y^{-} \mid x;\boldsymbol{\theta})\right]. \tag{16}$$

Minimizing these components encourages increasing the likelihood of y^{+} and decreasing that of y^{-}, respectively. Moreover, we examined online DPO (Guo et al., 2024) in Section C.2. Unlike the fixed preference dataset used in standard DPO, online DPO generates new response pairs from the current policy during training and assigns their preference order using a reward model. For each collected batch, the DPO objective and its gradient-equivalent decomposition above apply in the same way, with the sampled pairs held fixed during the update.

A.3 PPO

PPO seeks to directly maximize the expected reward $\mathcal{J}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}}\mathbb{E}_{\pi_{\boldsymbol{\theta}}}[R(x,y)]$, where the expectations average over training prompts and responses sampled from the current policy, respectively. Applying the likelihood-ratio identity (Williams, 1992) gives $\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}}\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[R(x,y)\nabla_{\boldsymbol{\theta}}\log\pi(y \mid x;\boldsymbol{\theta})\right]$. Expanding the response log-likelihood across tokens and expressing expected returns relative to a state-value baseline gives the equivalent token-level expression

$$\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}}\mathbb{E}_{\pi_{\boldsymbol{\theta}}}\Big[\sum_{j=1}^{|y|} A^{\pi_{\boldsymbol{\theta}}}(x, y_{<j}, y_j)\nabla_{\boldsymbol{\theta}}\log\pi(y_j \mid x, y_{<j};\boldsymbol{\theta})\Big], \tag{17}$$

where $A^{\pi_{\boldsymbol{\theta}}}$ measures the expected return of a token relative to the state-value baseline.

PPO reuses responses sampled from an old policy parameterized by $\boldsymbol{\theta}_{\text{old}}$. Following the main text, we denote the tokenized rollout dataset by $\mathcal{D}_{\text{ppo}} = \{z = ((x, y_{<j}), y_j)\}$ and the estimated advantage of token y_j by $A(z)$. The advantages are held fixed during each policy-optimization phase. Using the token-level importance ratio $r(z;\boldsymbol{\theta}) = \frac{\pi(y_j|x,y_{<j};\boldsymbol{\theta})}{\pi(y_j|x,y_{<j};\boldsymbol{\theta}_{\text{old}})}$, PPO constructs the local surrogate $\mathcal{J}_{\text{sur}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}_{\text{ppo}}}\left[r(z;\boldsymbol{\theta})A(z)\right]$. Its gradient is $\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\text{sur}}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\mathcal{D}_{\text{ppo}}}\left[r(z;\boldsymbol{\theta})A(z)\nabla_{\boldsymbol{\theta}}\log\pi(y_j \mid x, y_{<j};\boldsymbol{\theta})\right]$, which provides a local policy-gradient approximation using the fixed rollout data.

To discourage excessively large policy changes, PPO clips the surrogate in the direction favored by the advantage. Written as a loss to minimize, its clipped policy surrogate is

$$\mathcal{L}_{\text{ppo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\text{ppo}}}\left[\text{clip}_{A(z)}\big(r(z;\boldsymbol{\theta})\big)A(z)\right], \tag{18}$$

where

$$\text{clip}_{a}(r) = \begin{cases}\min(r, 1+\epsilon), & a \geq 0,\\ \max(r, 1-\epsilon), & a < 0,\end{cases} \tag{19}$$

and $\epsilon > 0$ is the clipping threshold. The sign of the advantage therefore yields the exact decomposition

$$\mathcal{L}^{+}_{\text{ppo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\text{ppo}}}\left[\text{clip}_{A(z)}\big(r(z;\boldsymbol{\theta})\big)A(z)\mathbb{1}_{\{A(z)>0\}}\right], \tag{20}$$

$$\mathcal{L}^{-}_{\text{ppo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\text{ppo}}}\left[\text{clip}_{A(z)}\big(r(z;\boldsymbol{\theta})\big)A(z)\mathbb{1}_{\{A(z)<0\}}\right]. \tag{21}$$

Outside the clipped regions, minimizing the two components encourages increasing the likelihood of positive-advantage tokens and decreasing that of negative-advantage tokens, respectively. Throughout this decomposition, $\mathcal{L}_{\text{ppo}}$ denotes the clipped policy-surrogate term rather than the complete training loss, which may additionally include explicit KL terms outside the scope of our analysis.

A.4 GRPO

GRPO follows the policy-surrogate approach of PPO but estimates advantages from group-relative rewards without a learned value model. We consider its outcome-supervised form, where each

sampled response receives a scalar reward. For each prompt x, the old policy generates G responses $\{y_i\}_{i=1}^{G}$. Their group-relative advantages are

$$A(x, y_i) = \frac{R(x, y_i) - \frac{1}{G}\sum_{k=1}^{G} R(x, y_k)}{\mathrm{std}\big(\{R(x, y_k)\}_{k=1}^{G}\big)}, \tag{22}$$

where std denotes the standard deviation of the rewards within each group. For groups with zero reward variance, we define all advantages as zero. Each response shares its advantage across all optimized tokens, and the advantages remain fixed during policy optimization.

We denote the response-level rollout dataset by $\mathcal{D}_{\mathrm{grpo}} = \{z = (x, y)\}$, where each response retains the advantage computed within its sampling group. The empirical expectation averages over prompts and then equally over the G responses generated for each prompt. Although advantages are response-level, the importance ratio remains token-wise as in PPO, which is given by $r_j(z; \boldsymbol{\theta}) = \frac{\pi(y_j|x, y_{<j}; \boldsymbol{\theta})}{\pi(y_j|x, y_{<j}; \boldsymbol{\theta}_{\mathrm{old}})}$. Using the clipping function defined in Appendix A.3, the clipped policy surrogate is

$$\mathcal{L}_{\mathrm{grpo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\mathrm{grpo}}}\Big[\frac{1}{|y|}\sum_{j=1}^{|y|} \mathrm{clip}_{A(z)}\big(r_j(z; \boldsymbol{\theta})\big) A(z)\Big]. \tag{23}$$

The within-response average preserves the response-level weighting rather than assigning greater weight to longer responses. As in PPO, the sign of the advantage yields an exact decomposition:

$$\mathcal{L}^{+}_{\mathrm{grpo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\mathrm{grpo}}}\Big[\frac{1}{|y|}\sum_{j=1}^{|y|} \mathrm{clip}_{A(z)}\big(r_j(z; \boldsymbol{\theta})\big) A(z) \mathbb{1}_{\{A(z)>0\}}\Big], \tag{24}$$

$$\mathcal{L}^{-}_{\mathrm{grpo}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\mathrm{grpo}}}\Big[\frac{1}{|y|}\sum_{j=1}^{|y|} \mathrm{clip}_{A(z)}\big(r_j(z; \boldsymbol{\theta})\big) A(z) \mathbb{1}_{\{A(z)<0\}}\Big]. \tag{25}$$

Positive and negative learning operate on responses with rewards above and below the group mean, respectively. As in PPO, $\mathcal{L}_{\mathrm{grpo}}$ denotes the clipped policy-surrogate term. The additional reference-policy KL penalty lies outside the scope of this component-wise analysis.

A.5 OPD

OPD aims to transfer the capabilities of a stronger teacher to a student model by training on student-generated responses with supervision from the frozen teacher. Unlike PPO and GRPO, its learning signal comes from the teacher distribution rather than task rewards. Here, we consider the sampled-token formulation. Following PPO, we denote the tokenized rollout dataset by $\mathcal{D}_{\mathrm{opd}} = \{z = ((x, y_{<j}), y_j)\}$. The teacher–student log-probability gap defines the token-level learning signal:

$$A(z; \boldsymbol{\theta}) = \log \pi_{\mathrm{T}}(y_j \mid x, y_{<j}) - \log \pi(y_j \mid x, y_{<j}; \boldsymbol{\theta}), \tag{26}$$

where π_{T} denotes the frozen teacher. Positive and negative values indicate that the teacher assigns higher and lower probability to the sampled token than the student, respectively. The learning signal is held fixed during differentiation. Using the token-level importance ratio $r(z; \boldsymbol{\theta})$ and clipping function defined for PPO, the OPD policy surrogate is given by

$$\mathcal{L}_{\mathrm{opd}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\mathrm{opd}}}\Big[\mathrm{clip}_{A(z;\boldsymbol{\theta})}\big(r(z; \boldsymbol{\theta})\big) A(z; \boldsymbol{\theta}|_{\mathrm{detach}})\Big]. \tag{27}$$

The empirical expectation averages over all valid completion tokens in the rollout batch, preserving token-level rather than response-level weighting.

As in PPO and GRPO, the sign of the learning signal yields an exact decomposition:

$$\mathcal{L}^{+}_{\mathrm{opd}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\mathrm{opd}}}\Big[\mathrm{clip}_{A(z;\boldsymbol{\theta})}\big(r(z; \boldsymbol{\theta})\big) A(z; \boldsymbol{\theta}|_{\mathrm{detach}}) \mathbf{1}_{\{A(z;\boldsymbol{\theta})>0\}}\Big], \tag{28}$$

$$\mathcal{L}^{-}_{\mathrm{opd}}(\boldsymbol{\theta}) = -\hat{\mathbb{E}}_{\mathcal{D}_{\mathrm{opd}}}\Big[\mathrm{clip}_{A(z;\boldsymbol{\theta})}\big(r(z; \boldsymbol{\theta})\big) A(z; \boldsymbol{\theta}|_{\mathrm{detach}}) \mathbf{1}_{\{A(z;\boldsymbol{\theta})<0\}}\Big]. \tag{29}$$

Since zero-gap tokens contribute nothing, $\mathcal{L}_{\mathrm{opd}} = \mathcal{L}^{+}_{\mathrm{opd}} + \mathcal{L}^{-}_{\mathrm{opd}}$. Outside the clipped regions, minimizing these components encourages increasing the likelihood of tokens assigned higher probability by the teacher and decreasing that of tokens assigned lower probability, respectively.

B EXPERIMENTAL CONFIGURATIONS

In this section, we describe the experimental configurations for NPO, DPO, PPO, GRPO, and OPD, including training, endpoint construction, and performance evaluation. We then detail the shared endpoint-probing procedures and protocols for comparing controlled variants with their baselines. All experiments were conducted on servers equipped with NVIDIA H100 80GB HBM3 and A100 80GB SXM4 GPUs and AMD EPYC 7V13 CPUs. Our source code is available at `https://anonymous.4open.science/r/Negative-learning-C5A6` and will be fully open-sourced upon acceptance.

B.1 NPO

Data and initialization. Following the NPO setup (Zhang et al., 2024) and the OpenUnlearning implementation (Dorna et al., 2025), we performed unlearning on the TOFU `forget05` split, which contains 200 forget examples. We initialized Llama3.2-1B and Llama3.2-3B from the TOFU-full checkpoints provided by OpenUnlearning.

Training. We used NPO with $\beta = 0.1$ without retention regularization, leaving only the negative-learning term. We trained for 10 epochs with a batch size of 32, corresponding to 70 optimizer updates, using AdamW with a learning rate of 10^{-5}, weight decay of 0.01, one epoch of warmup followed by linear decay, and gradient clipping at a maximum norm of 1. The maximum sequence length was 512 tokens. These settings follow the default NPO configuration provided by OpenUnlearning.

Endpoint construction and probing. For each model, we constructed $\mathcal{D}_{\text{end}}$ by pairing the rephrased forget prompts provided by TOFU with responses greedily decoded from the final step-70 checkpoint, using a maximum generation length of 128 tokens. We probed checkpoints every 10 steps.

B.2 DPO

Data and initialization. Following Tunstall et al. (2023), we first performed supervised fine-tuning on UltraChat-200k for all base models, followed by DPO training on UltraFeedback. We used Pythia-2.8B and Qwen3-1.7B for endpoint probing and additionally evaluated Llama3-8B in the controlled-method comparisons.

Training. All three models used $\beta = 0.1$ and were trained for two epochs with AdamW and weight decay 0.01. The model-specific configurations were:

Model	Global batch size	Initial learning rate
Pythia-2.8B	32	5×10^{-7}
Qwen3-1.7B	512	5×10^{-5}
Llama3-8B	256	1×10^{-5}

For Pythia-2.8B, we used a linear learning-rate schedule with a 5% warmup ratio and gradient clipping at a maximum norm of 1.

Endpoint construction and probing. For in-distribution probing, we used held-out UltraFeedback prompts. For out-of-distribution probing, we used HH-RLHF-helpfulness prompts. By default, each prompt was paired with a response greedily decoded from the final DPO checkpoint. We evaluated the full DPO objective and its positive- and negative-learning components at the same jointly trained checkpoints.

Performance evaluation. We evaluated preference alignment on AlpacaEval using GPT-5.5 as the judge in an *LLM-as-a-judge* framework (Dubois et al., 2024). Candidate responses were greedily decoded with a maximum generation length of 512 tokens. We compared each candidate model with its corresponding supervised fine-tuning baseline and reported the raw win rate without length correction, measuring how often the judge preferred the candidate response. These evaluations test whether the probing analysis motivates useful interventions.

B.3 PPO

Data and initialization. As in DPO, we initialized PPO from models supervised fine-tuned on UltraChat-200k and used UltraFeedback prompts for online rollouts. Pythia-2.8B and Qwen3-1.7B were used for probing, with Llama3-8B additionally included in the controlled-method comparisons.

Training and rollout generation. We used AdamW and gradient clipping at a maximum norm of 1. We sampled one response per prompt with top-$p = 0.9$. The model-specific configurations were:

Model	Global batch	Learning rate	Temperature	Max. new tokens
Pythia-2.8B	128	2×10^{-6}	1.0	256
Qwen3-1.7B	128	2×10^{-6}	0.7	400
Llama3-8B	32	5×10^{-7}	0.7	400

For Qwen3-1.7B and Llama3-8B, the maximum prompt length was 256 tokens. Pythia-2.8B and Qwen3-1.7B rollouts were scored by ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024), whereas Llama3-8B rollouts were scored by Skywork-Reward-Llama3.1-8B (Liu et al., 2024). For Pythia-2.8B, the reported training trajectory extended to step 2400, with endpoint probing performed every 400 steps.

Endpoint construction and probing. We constructed endpoint datasets from held-out HH-RLHF-helpfulness prompts paired with responses greedily decoded from the final PPO checkpoints. At each probed checkpoint, we computed endpoint-probing scores for the clipped policy surrogate and its positive- and negative-advantage components.

Performance evaluation. We used the same AlpacaEval protocol as for DPO, with greedy decoding capped at 512 generated tokens and comparison against the corresponding supervised fine-tuning baseline.

B.4 GRPO

We evaluated GRPO in three settings: single-turn function calling, multi-turn retail interaction, and mathematical reasoning.

Single-Turn Function Calling. We used Gemma4-E4B (the instruction-tuned checkpoint) on Hermes Function-Calling v1. The experiment used approximately 2000 training prompts and 300 held-out evaluation prompts. We applied no additional supervised fine-tuning before GRPO. We trained for 58 updates, using 32 prompts per update and a group size of $G = 5$, yielding 160 sampled responses per update. We applied *low-rank adaptation* (LoRA) (Hu et al., 2021) to all linear layers with rank 32 and $\alpha = 64$. We used AdamW with a learning rate of 10^{-6} and a KL coefficient of 10^{-3}. Responses were sampled at temperature 1.0 with a maximum generation length of 1024 tokens and scored by a programmatic function-calling reward. Following the shared endpoint-construction protocol, we paired held-out prompts with responses greedily decoded from the final model. We greedily decoded responses to 300 held-out prompts, scored them using the training reward function, and reported the mean programmatic reward.

Multi-Turn Retail Interaction. We initialized Qwen3.5-9B from a retail-specific supervised fine-tuning checkpoint and trained on the retail domain of τ^2-Bench. Customer responses were generated by a frozen Qwen3.5-4B simulator. Each update used eight prompts and $G = 8$. We trained a text-tower LoRA adapter with rank 32 and $\alpha = 64$ using AdamW, a learning rate of 10^{-4}, and a KL coefficient of 10^{-3}. Policy responses were sampled at temperature 1.0, with at most 512 generated tokens per assistant turn, 16 assistant turns, and 12 simulated-user replies. Binary rewards were determined from the final environment state. We evaluated one fixed-seed trajectory on retail evaluation tasks and reported the task-success rate.

Mathematical Reasoning. We trained Qwen3-0.6B on MATH problems, balanced between Levels 4 and 5. We applied no additional supervised fine-tuning before GRPO. Each update used eight prompts and $G = 8$. We performed full-parameter training with AdamW, a learning rate of 10^{-6}, and a KL coefficient of 10^{-3}. Responses were sampled at temperature 0.6, top-$p = 0.95$, and top-$k = 20$, with a maximum generation length about 16000 tokens. Rewards were assigned by exact-answer verification. We evaluated checkpoints on MATH500, using one fixed-seed response per problem, and reported pass@1.

B.5 OPD

Data and initialization. We use Qwen3-1.7B-Base as the student and a frozen Qwen3-1.7B as the teacher, and train on `guanning-ai/dapo14k` (Yu et al., 2025). We adapt the sampled-token OPD baseline released with vOPD (Oh et al., 2026).

Training. We train with LoRA using rank 64 and scaling factor 128, a learning rate of 10^{-5} with cosine decay, and a global batch size of 64 prompts. Training uses distributed data parallelism and runs for one epoch, comprising 211 updates over $13{,}504$ examples. We conduct one training run with seed 42.

Endpoint construction and probing. We constructed two endpoint datasets, each containing 64 prompts, drawn from held-out data and examples consumed during training, respectively. Each prompt was paired with a response sampled from the final step-211 student checkpoint, using temperature 1.0, top-$p = 1.0$, and a maximum generation length of 2048 tokens. At each probed checkpoint, we computed endpoint-probing scores for the OPD surrogate and its positive- and negative-learning components over the trainable LoRA parameters.

B.6 Endpoint-Probing Implementation

Endpoint choice. Unless otherwise specified, endpoint datasets paired held-out prompts with responses greedily decoded from the final checkpoint. For these endpoints, greedy decoding removes response-sampling randomness for a fixed prompt and checkpoint. The probe is retrospective and measures whether intermediate updates support the realized final responses, not whether these responses are optimal. We compare component effects under matched endpoints and assess task performance separately using the metrics specified above.

Gradient estimation. Component gradients were evaluated at the same jointly trained model state, using the decompositions in Appendix A. To reduce computation in the DPO and PPO analyses, we sampled 500 examples from the objective data and 500 examples from $\mathcal{D}_{\mathrm{end}}$. We partitioned each sample into mini-batches of four examples for DPO and six for PPO to limit GPU memory usage. We averaged the endpoint gradients across mini-batches before computing the probing scores.

Approximation validation. To validate the first-order approximation in Figure 1, we compared the endpoint-probing score δ with the exact one-step endpoint-loss reduction Δ for DPO on Qwen3-1.7B and PPO on Pythia-2.8B. At each probed checkpoint, we computed δ and applied one gradient update to a copy of the model to measure Δ, using the same objective mini-batch and endpoint data for both quantities. Endpoint prompts were drawn from HH-RLHF-helpfulness, with responses greedily decoded from the final checkpoint. We kept the original checkpoint unchanged across comparisons.

B.7 DPO Component Ablations

We detail the ablations in Figure 4, which examine whether components with negative endpoint-probing scores remain beneficial to fine-tuning performance.

Removing negative learning early. We initialized from the same supervised fine-tuning checkpoints as standard DPO and trained with only $\mathcal{L}^{+}_{\mathrm{dpo}}$. We evaluated checkpoints through step 3000 for Pythia-2.8B and step 120 for Qwen3-1.7B, corresponding to the early phase when negative learning opposes endpoint fitting. These results are labeled *w/o NEG*.

Removing positive learning later. We initialized from the standard DPO checkpoints at step 6000 for Pythia-2.8B and step 180 for Qwen3-1.7B, when positive learning opposes endpoint fitting. We then continued training with only $\mathcal{L}^{-}_{\mathrm{dpo}}$. These results are labeled *w/o POS*. Training steps retain the numbering of the original DPO runs.

For both ablations, we removed the specified component without rescaling the remaining component. All other training settings and the AlpacaEval evaluation protocol remained the same as for standard DPO.

B.8 Controlled Negative Learning

We detail the scheduling rules used to coordinate positive and negative learning in cDPO, cPPO, and cGRPO, followed by the hyperparameters selected on held-out validation data.

cDPO. We gradually shift the relative emphasis from positive to negative learning to reflect their changing roles during training. Our DPO analysis shows that positive learning drives target shaping early, whereas negative learning becomes endpoint-aligned later for the out-of-distribution endpoints studied. We therefore use

$$\mathcal{L}_{\mathrm{cdpo}}(\boldsymbol{\theta}) = 2(1-\lambda_t)\mathcal{L}^{+}_{\mathrm{dpo}}(\boldsymbol{\theta}) + 2\lambda_t\mathcal{L}^{-}_{\mathrm{dpo}}(\boldsymbol{\theta}), \qquad \lambda_t = \max\big(\min\big(\frac{t-t_1}{t_2-t_1}, 1\big), 0\big). \tag{30}$$

Here, $t_1 < t_2$ specify the transition interval, during which λ_t increases linearly from 0 to 1. For the special case $t_1 = t_2$, we use an abrupt transition, with $\lambda_t = 0$ before the transition step and $\lambda_t = 1$ from that step onward. The factors of two recover standard DPO at $\lambda_t = 1/2$. The transition interval may extend beyond the training horizon $[0, T]$: setting $t_1 < 0$ and $t_2 > T$ keeps both components active throughout training.

cPPO and cGRPO. In the PPO and GRPO settings studied, negative learning counteracts endpoint-aligned positive learning. We control the strength of negative learning to examine whether coordinating the two components can improve fine-tuning performance. The controlled policy surrogates are

$$\begin{aligned} \mathcal{L}_{\mathrm{cppo}}(\boldsymbol{\theta}) &= \mathcal{L}^{+}_{\mathrm{ppo}}(\boldsymbol{\theta}) + \lambda_t\mathcal{L}^{-}_{\mathrm{ppo}}(\boldsymbol{\theta}), \\ \mathcal{L}_{\mathrm{cgrpo}}(\boldsymbol{\theta}) &= \mathcal{L}^{+}_{\mathrm{grpo}}(\boldsymbol{\theta}) + \lambda_t\mathcal{L}^{-}_{\mathrm{grpo}}(\boldsymbol{\theta}). \end{aligned} \tag{31}$$

Here, $\lambda_t \geq 0$ controls the strength of negative learning, with $\lambda_t = 1$ recovering the standard policy surrogate and $\lambda_t < 1$ and $\lambda_t > 1$ attenuating and strengthening negative learning, respectively. Other objective terms remain unchanged. We consider three scheduling rules:

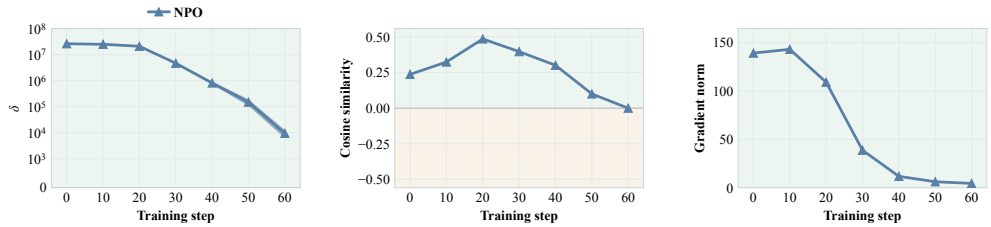
- **Periodic attenuation.** We periodically weaken negative learning, allowing positive updates to more strongly reinforce currently high-reward responses. We set

$$\lambda_t = 1 + \max\big(\min\big(\sin(\pi t/t_3), 0\big), -\tau\big). \tag{32}$$

  Here, $t_3 > 0$ determines the period $2t_3$, and $\tau \in (0, 1]$ controls the maximum attenuation.
- **Constant scaling.** We maintain negative learning at a fixed reduced or enhanced strength, aiming to favor exploitation or exploration, respectively. We set $\lambda_t = c$, where $c \geq 0$. Here, $c < 1$ reduces the strength of negative learning, whereas $c > 1$ enhances it.
- **Adaptive strengthening.** For GRPO, we increase the strength of negative learning as the fraction of fully successful rollout groups rises, aiming to encourage further exploration. We set $\lambda_t = 1 + \kappa s_t$, where $\kappa \geq 0$. Here, s_t is the fraction of rollout groups in the batch in which all responses are successful, and κ controls the maximum increase in negative-learning strength.

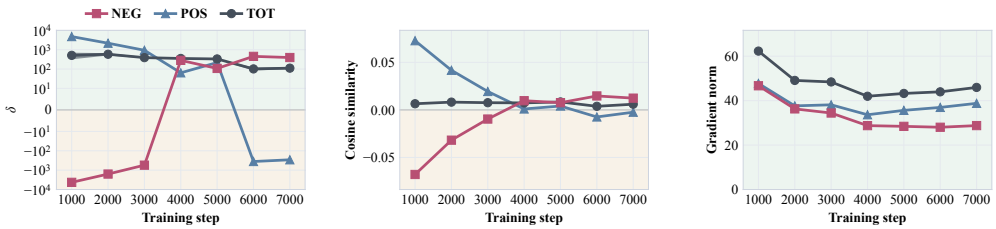
Hyperparameter selection. We tuned the scheduling hyperparameters on held-out validation data and selected the configurations below for the main comparisons. In the table, Periodic, Constant, and Adaptive refer to periodic attenuation, constant scaling, and adaptive strengthening, respectively.

Method	Model	Control hyperparameters
cDPO	Pythia-2.8B	$t_1 = 2500,\ t_2 = 6500$
cDPO	Qwen3-1.7B	$t_1 = 100,\ t_2 = 200$
cDPO	Llama3-8B	$t_1 = 20,\ t_2 = 60$
cPPO	Pythia-2.8B	Periodic: $t_3 = 2,\ \tau = 0.08$
cPPO	Qwen3-1.7B	Periodic: $t_3 = 100,\ \tau = 0.01$
cPPO	Llama3-8B	Periodic: $t_3 = 5,\ \tau = 0.99$
cGRPO	Gemma4-E4B	Periodic: $t_3 = 5,\ \tau = 0.001$
cGRPO	Qwen3.5-9B	Adaptive: $\kappa = 0.8$
cGRPO	Qwen3-0.6B	Constant: $c = 0.9$



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 8: **NPO Probing on Llama3.2-3B.** We report endpoint-probing scores, cosine similarities between the NPO and endpoint gradients, and their gradient norms.



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 9: **DPO Probing on Pythia-2.8B.** We report endpoint-probing scores, cosine similarities with the endpoint gradient, and gradient norms for the full DPO objective and its positive and negative components.

C Additional Experiments

This section provides additional probing results and presents further controlled-method comparisons.

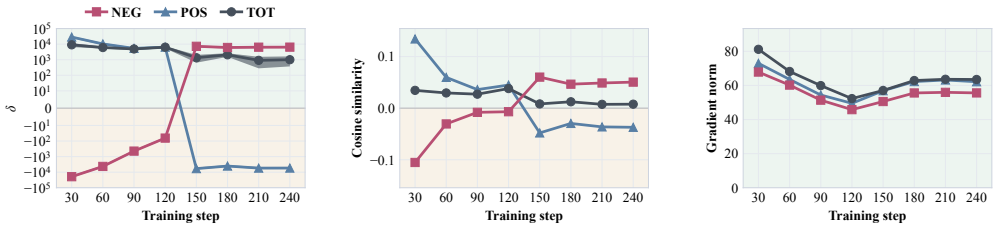
C.1 Endpoint Probing on Main Settings

Figure 6 complements the endpoint-probing scores with cosine similarities and gradient norms, based on the decomposition $\delta(\mathcal{L};\boldsymbol{\theta}) = \|\nabla\mathcal{L}_{\text{end}}(\boldsymbol{\theta})\|\|\nabla\mathcal{L}(\boldsymbol{\theta})\|\cos(\nabla\mathcal{L}_{\text{end}}(\boldsymbol{\theta}), \nabla\mathcal{L}(\boldsymbol{\theta}))$. These quantities distinguish gradient direction from scale, as a small probing score may reflect either small gradient norms or weak directional alignment. We extend this analysis to the experiments below. All quantities within each experiment were evaluated at the same checkpoints using the same endpoint set.

NPO. Figure 8 provides the complete NPO results on Llama3.2-3B. The positive probing scores across the evaluated checkpoints confirm that negative learning exhibits target shaping within this setting, consistent with previous findings (Zhang et al., 2024; Zhu et al., 2025). However, the cosine similarity gradually approaches zero, indicating that endpoint alignment weakens during training. This trend raises the possibility that target shaping may not persist under longer training, echoing previous reports of instability in negative-only learning (Wang et al., 2025e; Li et al., 2026).

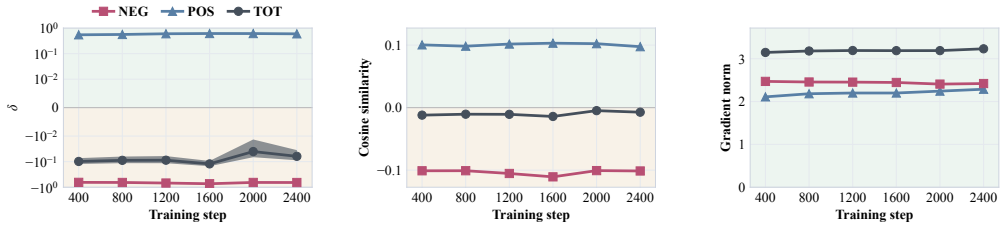
DPO and PPO. Figures 9–12 report the complete results for Pythia-2.8B and Qwen3-1.7B. For DPO, the cosine similarities show the same phase-dependent transition as the probing scores. Positive learning is aligned with the endpoints early, while negative learning becomes aligned later. The gradient norms provide complementary information but do not alone explain this transition. For PPO, positive and negative learning remain oppositely aligned with the endpoints across checkpoints. Their contributions largely cancel, leaving the full-surrogate score small despite non-negligible component gradients. Thus, the small overall score reflects opposing contributions from positive and negative learning rather than weak gradients.

GRPO. The main text examines GRPO on function calling using a single model. We further vary both the model and task to assess whether the observed behavior extends to multi-turn tool use and mathematical reasoning. We probed Gemma4-E4B on Hermes Function-Calling v1 at checkpoints $\{0, 10, 20, 30, 40, 50, 58\}$, Qwen3.5-9B on τ^2 Retail at $\{0, 3, 5, 10\}$, and Qwen3-0.6B on MATH at



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 10: **DPO Probing on Qwen3-1.7B.** We report endpoint-probing scores, cosine similarities with the endpoint gradient, and gradient norms for the full DPO objective and its positive and negative components.



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 11: **PPO Probing on Pythia-2.8B.** We report endpoint-probing scores, cosine similarities with the endpoint gradient, and gradient norms for the full PPO policy surrogate and its positive and negative components.

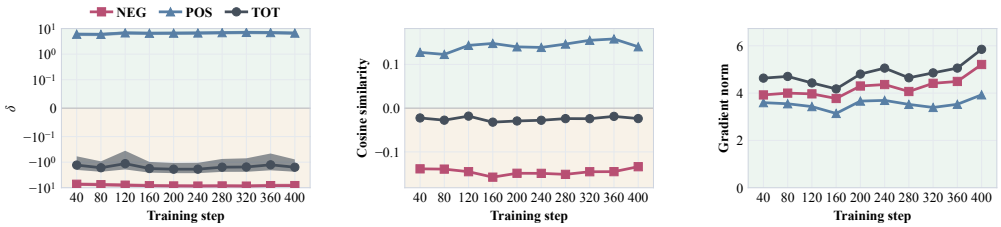
$\{0, 10, 20, 30\}$. Their endpoint sets contained 128, 16, and 64 final-policy responses or trajectories, respectively, and remained fixed across checkpoints. Figures 13–15 report the probing scores, cosine similarities, and gradient norms.

Across all settings, positive learning remains directed toward the endpoints, while negative learning moves in the opposite direction, consistent with our observations in Section 4.3. Their cosine similarities show the same directional separation. The full-surrogate score fluctuates across checkpoints, although its magnitude remains small relative to the component scores. On MATH, it even remains positive throughout. This does not contradict our observations about negative learning, which consistently opposes endpoint fitting across all three settings. The sign of the total score instead depends on the relative strengths of the positive and negative contributions, which can be further controlled.

Sensitivity to sampling. To test whether our findings depend on greedy endpoint decoding, we use the same DPO training trajectory as in Figure 10, but randomly sample one response per prompt from its final checkpoint at temperatures 0.2 and 0.7, both with top-$p = 1.0$. Each endpoint set remains fixed across the eight probed checkpoints, with all other probing settings unchanged. Figure 16 shows that both sampling conditions preserve the POS and NEG alignment signs observed with greedy endpoints, including their reversal between steps 120 and 150. Although the magnitudes change, the main pattern persists, supporting the robustness of our findings to various decoding rules.

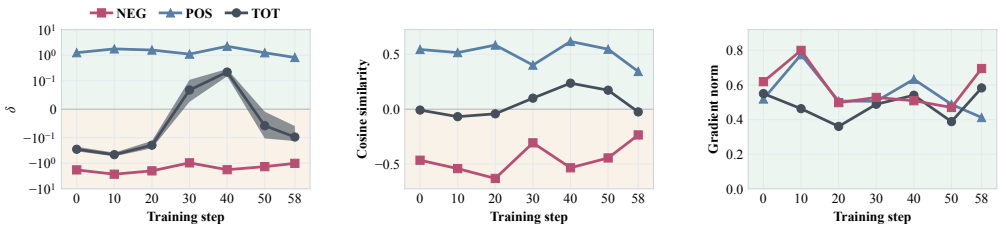
C.2 Endpoint Probing on Online DPO

We also examine on-policy DPO, which collects fresh preference pairs during training rather than using a fixed dataset as in off-policy DPO. The policy and frozen reference are initialized from the same Qwen3-1.7B checkpoint used in our off-policy experiment. Before each update, we sample two responses for each of 128 prompts from the current policy and rank them with a frozen Skywork-Reward-V2-Llama-3.1-8B reward model. We train for 240 updates with $\beta = 0.1$ and a peak learning rate of 5×10^{-7}. The final checkpoint achieves a raw win rate of 62.11% against the corresponding supervised fine-tuning baseline on 805 AlpacaEval instructions. For probing, we pair 500 HH-RLHF-helpfulness prompts with greedy responses from the final checkpoint, capped at 400 tokens. We estimate objective gradients using a fixed set of 500 preference pairs sampled from the completed



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 12: **PPO Probing on Qwen3-1.7B.** We report endpoint-probing scores, cosine similarities with the endpoint gradient, and gradient norms for the full PPO policy surrogate and its positive and negative components.



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 13: **GRPO Probing on Hermes Function-Calling v1.** We report endpoint-probing scores for the full GRPO policy surrogate and its positive and negative components, together with component cosine similarities with the endpoint gradient and gradient norms.

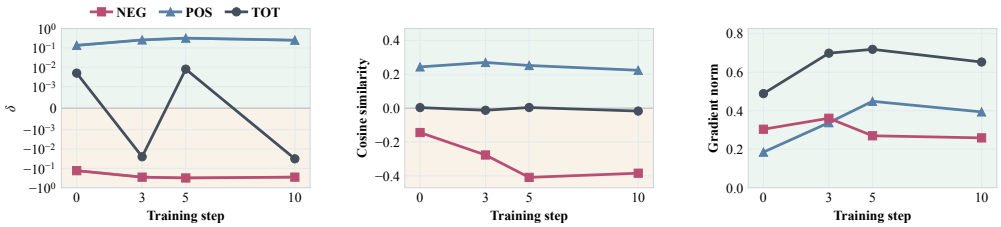
training run. At steps $\{30, 60, 90, 120, 150, 180, 210, 240\}$, we recompute the component gradients on these same pairs.

Figure 17 shows a transition similar to off-policy DPO. Positive learning initially aligns with the final endpoints, while negative learning opposes them. The probing score for negative learning becomes positive by step 120, and that for positive learning becomes negative at step 150. The score for the full DPO objective remains positive but small relative to the component scores. These results suggest that on- versus off-policy data collection alone does not determine component alignment. The objective and its supervision signal may be more relevant to the observed pattern.

C.3 Endpoint Probing on OPD

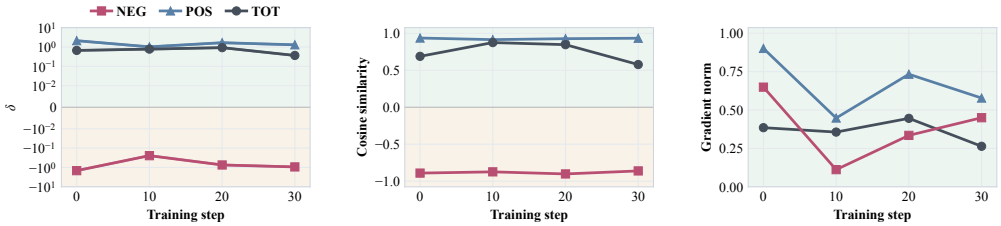
We further examine whether the opposing component alignments observed in PPO and GRPO also arise in other on-policy settings. OPD provides a complementary case: it also trains on student-generated responses, but uses teacher-provided supervision rather than task rewards. We probe Qwen3-1.7B-Base trained with a frozen Qwen3-1.7B teacher, using the training configuration described in Appendix B. We consider two endpoint settings: 64 held-out prompts with a distinct set of 64 prompts for objective-gradient estimation, and 64 training prompts shared between the two. The training-prompt setting extends the held-out endpoint construction used in our main analysis. The latter prompts are drawn from examples consumed during the full training run and need not have been encountered by every checkpoint. Thus, this comparison varies both training exposure and prompt overlap. We probe checkpoints at steps $\{0, 10, 30, 50, 70, 90, 110, 150, 180, 210\}$, using one sampled response per prompt for objective-gradient estimation.

Figure 18 reports probing scores, cosine similarities, and gradient norms. In both settings, negative learning remains positively endpoint-aligned across the evaluated checkpoints. Positive learning is initially aligned with the endpoints but becomes negatively aligned at steps 180 and 210. The cosine similarities exhibit the same directional change. The full-surrogate score remains positive on held-out prompts but becomes slightly negative at the final two checkpoints in the training-prompt setting. These dynamics differ from the stable opposing component alignments observed in PPO and GRPO. A possible explanation is that teacher supervision gives negative learning a more consistent



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 14: **GRPO Probing on τ^2 Retail.** We report endpoint-probing scores for the full GRPO policy surrogate and its positive and negative components, together with component cosine similarities with the endpoint gradient and gradient norms. We used Qwen3.5-9B for multi-turn tool use in the τ^2 Retail environment.



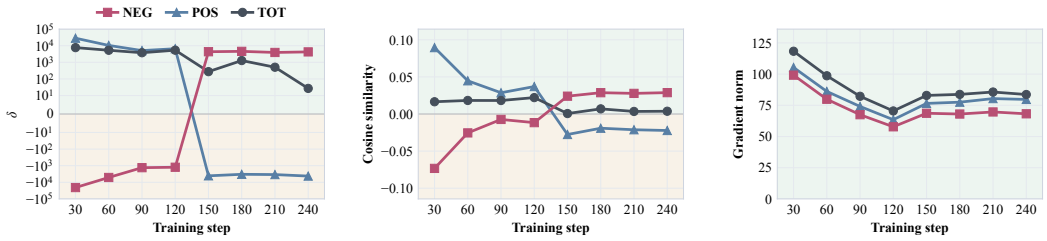
(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

Figure 15: **GRPO Probing on MATH.** We report endpoint-probing scores for the full GRPO policy surrogate and its positive and negative components, together with component cosine similarities with the endpoint gradient and gradient norms. We used Qwen3-0.6B for mathematical reasoning on MATH.

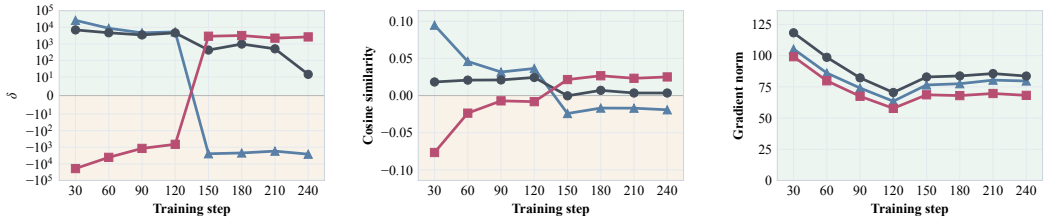
target-shaping direction, while positive updates need not remain aligned with the final responses. This echoes our earlier argument that component roles depend on the objective and training phase, suggesting that on-policy sampling alone does not imply the PPO/GRPO pattern.

C.4 Controlled Methods

For DPO and PPO, hyperparameters and checkpoints are selected by win rate against the corresponding supervised fine-tuning baseline on held-out UltraFeedback prompts, without using AlpacaEval test scores. For GRPO, hyperparameters and checkpoints are selected on validation data disjoint from both the training and test sets, using the same metrics as in the reported evaluations: mean programmatic reward for function calling, task-success rate for τ^2 Retail, and pass@1 for mathematical reasoning. Figures 19–21 report additional AlpacaEval results for different cDPO and cPPO schedules on Pythia-2.8B, Qwen3-1.7B, and Llama3-8B. Importantly, the improvements are not confined to the selected schedules. Multiple alternative configurations also outperform the corresponding baselines across all three models. For example, alternative cDPO schedules achieve win rates of 83.56–87.91% on Qwen3-1.7B and 69.54–77.52% on Llama3-8B, compared with baseline scores of 72.14% and 67.10%, respectively. Several alternative cPPO schedules likewise improve over their baselines across the three models.

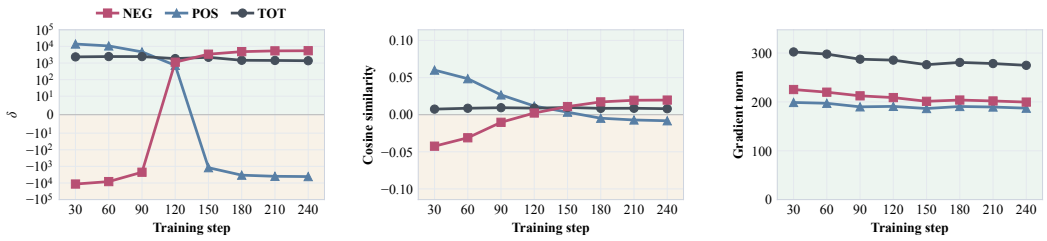


(a) temperature 0.2



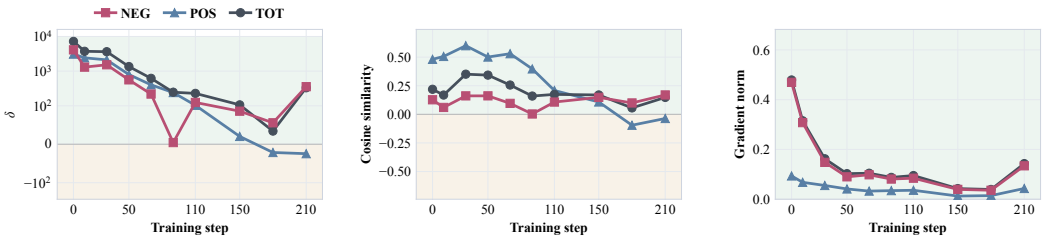
(b) temperature 0.7

Figure 16: **DPO Probing with Sampled Endpoints.** We report endpoint-probing scores, cosine similarities with the endpoint gradient, and gradient norms for Qwen3-1.7B using responses randomly sampled from the same final checkpoint at the indicated temperatures, rather than greedily decoded responses.



(a) Endpoint-probing score (b) Cosine similarity (c) Gradient norm

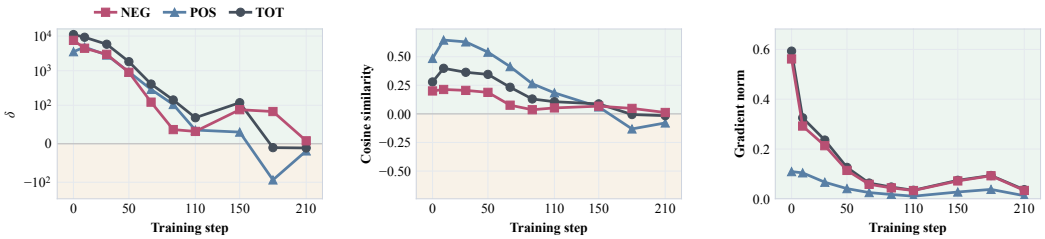
Figure 17: **Online DPO Probing on Qwen3-1.7B.** We report endpoint-probing scores, cosine similarities with the endpoint gradient, and gradient norms for the DPO objective and its positive and negative components. Probing uses fixed recorded preference pairs and greedy final-policy endpoints. Results are from one training run.



(a) Endpoint-probing score

(b) Cosine similarity

(c) Gradient norm

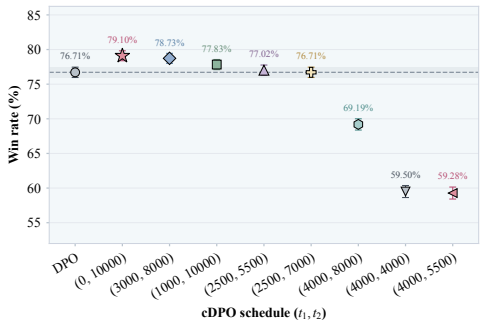


(d) Endpoint-probing score

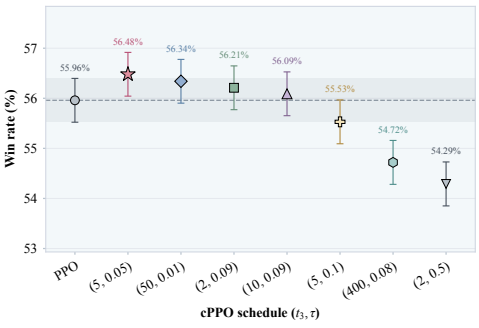
(e) Cosine similarity

(f) Gradient norm

Figure 18: **OPD Probing on Qwen3-1.7B-Base.** We report endpoint-probing scores, cosine similarities with the endpoint gradient, and gradient norms for the full OPD surrogate and its positive and negative components. The top row uses distinct held-out prompts for endpoint construction and objective-gradient estimation, while the bottom row uses the same training prompts for both. All results are from one training run.

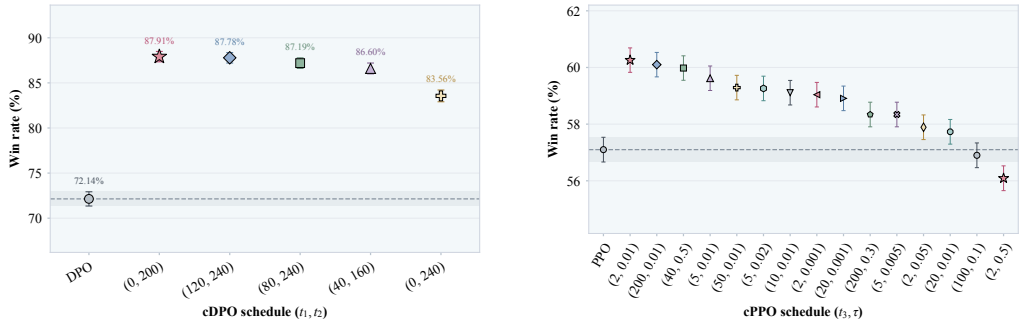


(a) cDPO schedules (t_1, t_2)



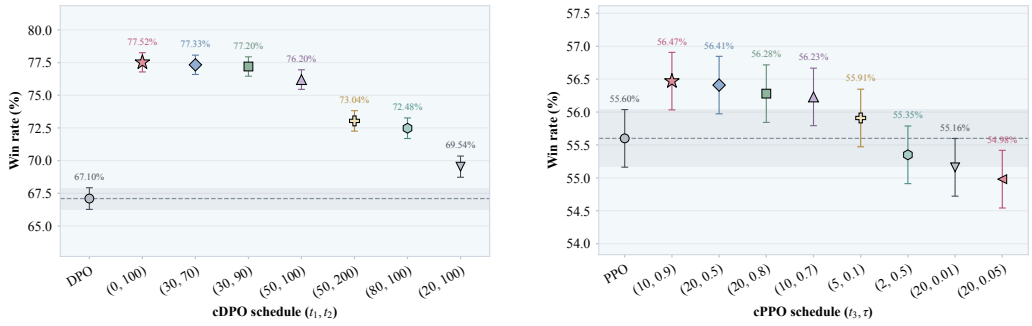
(b) cPPO schedules (t_3, τ)

Figure 19: **Controlled Variants on Pythia-2.8B.** We compare AlpacaEval win rates across different cDPO and cPPO schedules. Markers indicate mean test win rates across three independent training seeds, and error bars show $\pm$ one standard deviation across seeds. Dashed horizontal lines indicate the corresponding DPO and PPO baseline means, with shaded bands showing $\pm$ one standard deviation.



(a) cDPO schedules (t_1, t_2) (b) cPPO schedules (t_3, τ)

Figure 20: **Controlled Variants on Qwen3-1.7B.** We compare AlpacaEval win rates across different cDPO and cPPO schedules. Markers indicate mean test win rates across three independent training seeds, and error bars show $\pm$ one standard deviation across seeds. Dashed horizontal lines indicate the corresponding DPO and PPO baseline means, with shaded bands showing $\pm$ one standard deviation.



(a) cDPO schedules (t_1, t_2) (b) cPPO schedules (t_3, τ)

Figure 21: **Controlled Variants on Llama3-8B.** We compare AlpacaEval win rates across different cDPO and cPPO schedules. Markers indicate mean test win rates across three independent training seeds, and error bars show $\pm$ one standard deviation across seeds. Dashed horizontal lines indicate the corresponding DPO and PPO baseline means, with shaded bands showing $\pm$ one standard deviation.

D Mathematical Analysis of Endpoint Probing

This appendix establishes the local approximation underlying endpoint probing and gives sufficient conditions under which sparse checkpoint scores remain informative between observations.

Throughout, we fix the endpoint dataset $\mathcal{D}_{\text{end}}$ and its loss $\mathcal{L}_{\text{end}}$, and compute all gradients with respect to the same trainable parameters. For brevity, we write

$$e_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)}), \qquad g_t = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}), \qquad \delta_t = e_t^\top g_t,$$

where e_t and g_t are the endpoint and objective gradients at checkpoint t, respectively, and δ_t is the corresponding endpoint-probing score.

D.1 Local Error of the First-Order Approximation

We derive the first-order approximation underlying endpoint probing and its local error bound, mathematically clarifying how accurately $\eta\delta_t$ approximates the exact one-step change Δ_t. We make the standard local smoothness assumption that $\mathcal{L}_{\text{end}}$ has a Lipschitz-continuous gradient on a neighborhood containing the update segment, i.e.,

$$\|\nabla \mathcal{L}_{\text{end}}(u) - \nabla \mathcal{L}_{\text{end}}(v)\| \le H_{\text{end}} \|u - v\|, \tag{33}$$

where $H_{\text{end}} > 0$ is the local smoothness constant and $\|\cdot\|$ denotes the Euclidean norm.

Proposition D.1. *For a gradient step $\boldsymbol{\theta}^{(t)} - \eta g_t$ with $\eta > 0$, define the endpoint-loss reduction as*

$$\Delta_t = \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)}) - \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)} - \eta g_t).$$

Under the local smoothness assumption in Eq. (33), we have

$$|\Delta_t - \eta\delta_t| \le \frac{H_{\text{end}}\eta^2}{2}\|g_t\|^2.$$

Hence, δ_t correctly predicts the direction of the loss change whenever

$$|\delta_t| > \frac{H_{\text{end}}\eta}{2}\|g_t\|^2.$$

Proof. Let $\boldsymbol{\theta}(s) = \boldsymbol{\theta}^{(t)} - s\eta g_t$ for $s \in [0, 1]$. Since $\frac{\mathrm{d}}{\mathrm{d}s}\mathcal{L}_{\text{end}}(\boldsymbol{\theta}(s)) = -\eta \nabla \mathcal{L}_{\text{end}}(\boldsymbol{\theta}(s))^\top g_t$, integrating along the update segment and subtracting $\eta\delta_t$ gives

$$\Delta_t - \eta\delta_t = \eta \int_0^1 \left[\nabla \mathcal{L}_{\text{end}}(\boldsymbol{\theta}(s)) - \nabla \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)})\right]^\top g_t \, \mathrm{d}s.$$

The distance from $\boldsymbol{\theta}(s)$ to the initial checkpoint is $s\eta\|g_t\|$. Local smoothness therefore implies

$$\left\|\nabla \mathcal{L}_{\text{end}}(\boldsymbol{\theta}(s)) - \nabla \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)})\right\| \le H_{\text{end}} s\eta \|g_t\|.$$

Applying the triangle inequality and Cauchy–Schwarz, we obtain

$$\begin{aligned}|\Delta_t - \eta\delta_t| \le \eta \int_0^1 \left\|\nabla \mathcal{L}_{\text{end}}(\boldsymbol{\theta}(s)) - \nabla \mathcal{L}_{\text{end}}(\boldsymbol{\theta}^{(t)})\right\| \|g_t\| \, \mathrm{d}s &\le H_{\text{end}}\eta^2 \|g_t\|^2 \int_0^1 s \, \mathrm{d}s \\ &= \frac{H_{\text{end}}\eta^2}{2}\|g_t\|^2.\end{aligned}$$

Moreover, if $|\delta_t| > H_{\text{end}}\eta\|g_t\|^2/2$, then $|\Delta_t - \eta\delta_t| < |\eta\delta_t|$. Thus, the approximation error cannot reverse the sign of $\eta\delta_t$, and Δ_t and δ_t have the same sign. □

Learning-rate dependence. At a fixed checkpoint and for a fixed update direction, the absolute approximation error is bounded by a term quadratic in η. If $\delta_t \neq 0$, the relative error satisfies

$$\frac{|\Delta_t - \eta\delta_t|}{\eta|\delta_t|} \le \frac{H_{\text{end}}\eta\|g_t\|^2}{2|\delta_t|}. \tag{34}$$

When the right-hand side is below 1, the error is smaller than the magnitude of the first-order prediction, so Δ_t and δ_t have the same sign. Thus, a sufficiently small learning rate ensures sign agreement for a fixed nonzero score. Predictions near zero require more caution because even a small approximation error may reverse their sign.

Gradient-norm clipping. For a clipping threshold $C > 0$ for the gradient norm, define

$$\alpha_t = \min\Big(1, \frac{C}{\|g_t\|}\Big), \qquad \bar{g}_t = \alpha_t g_t,$$

with $\alpha_t = 1$ when $g_t = 0$. Let Δ_t^{clip} be the endpoint-loss reduction after $\boldsymbol{\theta}^{(t)} - \eta\bar{g}_t$. Proposition D.1 implies

$$\big|\Delta_t^{\mathrm{clip}} - \eta\alpha_t\delta_t\big| \le \frac{H_{\mathrm{end}}\eta^2}{2}\min(\|g_t\|^2, C^2). \tag{35}$$

Gradient clipping scales the update by α_t, so the first-order prediction becomes $\eta\alpha_t\delta_t$ without changing its sign. It also reduces the approximation-error bound by a factor of α_t^2, acting as a smaller effective learning rate (Zhang et al., 2019).

Optimizer-aware interpretation. Endpoint probing isolates the gradient contribution of each objective component at a given checkpoint, regardless of the optimizer used during training. Its additive decomposition therefore remains valid for models trained with AdamW. Specifically, δ_t measures whether the current objective gradient supports the endpoints, as formalized by the gradient step in Eq. (2). The same analysis also accommodates optimizer-specific updates. For an actual update $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta_t d_t$, local smoothness gives

$$\big|\mathcal{L}_{\mathrm{end}}(\boldsymbol{\theta}^{(t)}) - \mathcal{L}_{\mathrm{end}}(\boldsymbol{\theta}^{(t+1)}) - \eta_t e_t^\top d_t\big| \le \frac{H_{\mathrm{end}}\eta_t^2}{2}\|d_t\|^2. \tag{36}$$

For AdamW, d_t incorporates momentum, adaptive scaling, and weight decay, and may therefore differ from g_t. Our analysis focuses on the objective gradients to separate positive and negative learning, while predicting an actual optimizer step requires using d_t instead. A smaller learning rate and a smaller update-direction norm $\|d_t\|$ likewise tighten the approximation-error bound in this setting.

D.2 Impact of Sparse Checkpoint Sampling on Probing

We establish sufficient conditions under which a checkpoint score represents the direction of endpoint alignment between observations. With the endpoint dataset fixed, we consider an interval over which the probed objective is also fixed and has a locally Lipschitz-continuous gradient:

$$\|g_{k+1} - g_k\| \le H_g\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|, \tag{37}$$

where $H_g \ge 0$ is a local smoothness constant. This assumption controls gradient changes caused by parameter movement. It does not automatically hold across changes in rollout data or advantages in PPO and GRPO; applying the result across such changes requires the same bound to remain valid.

Proposition D.2. *Suppose that Eqs. (33) and (37) hold throughout the interval from t to s, with $\|e_k\| \le B_{\mathrm{end}}$ and $\|g_k\| \le B_g$. Then*

$$|\delta_s - \delta_t| \le \big(H_{\mathrm{end}}B_g + B_{\mathrm{end}}H_g\big)\sum_{k=t}^{s-1}\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|. \tag{38}$$

If the right-hand side is smaller than $|\delta_t|$, the score retains the sign of δ_t at every step from t through s.

Proof. Using $\delta_k = e_k^\top g_k$, we write

$$\delta_{k+1} - \delta_k = (e_{k+1} - e_k)^\top g_{k+1} + e_k^\top(g_{k+1} - g_k).$$

The triangle inequality and Cauchy–Schwarz give

$$|\delta_{k+1} - \delta_k| \le \|e_{k+1} - e_k\|\|g_{k+1}\| + \|e_k\|\|g_{k+1} - g_k\|.$$

Endpoint-loss smoothness implies

$$\|e_{k+1} - e_k\| \le H_{\text{end}} \|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|.$$

Combining this with the objective-gradient assumption and the norm bounds yields

$$|\delta_{k+1} - \delta_k| \le \big(H_{\text{end}} B_g + B_{\text{end}} H_g\big) \|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|.$$

Summing over the interval, we obtain

$$|\delta_s - \delta_t| = \Big|\sum_{k=t}^{s-1} (\delta_{k+1} - \delta_k)\Big| \le \sum_{k=t}^{s-1} |\delta_{k+1} - \delta_k| \le \big(H_{\text{end}} B_g + B_{\text{end}} H_g\big) \sum_{k=t}^{s-1} \|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|.$$

For any intermediate step $j \in \{t, \dots, s\}$, the same argument bounds $|\delta_j - \delta_t|$ by the corresponding sum up to $j-1$. Because all summands are nonnegative, this partial sum cannot exceed the bound for the full interval. If the full bound is smaller than $|\delta_t|$, then $|\delta_j - \delta_t| < |\delta_t|$, so δ_j cannot reach or cross zero. □

Learning-rate dependence. When the trajectory follows gradient descent on the probed objective, $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \eta_k g_k$. Using $\|g_k\| \le B_g$, we obtain

$$|\delta_s - \delta_t| \le \big(H_{\text{end}} B_g + B_{\text{end}} H_g\big) B_g \sum_{k=t}^{s-1} \eta_k. \tag{39}$$

For a constant learning rate η, the bound is proportional to $(s-t)\eta$. With the smoothness and norm bounds held fixed, smaller learning rates and shorter checkpoint intervals tighten this upper bound.

Gradient-norm clipping. For gradient descent on the probed objective with clipping threshold $C > 0$,

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\| = \eta_k \min(\|g_k\|, C) \le \eta_k \min(B_g, C).$$

Consequently,

$$|\delta_s - \delta_t| \le \big(H_{\text{end}} B_g + B_{\text{end}} H_g\big) \min(B_g, C) \sum_{k=t}^{s-1} \eta_k. \tag{40}$$

Clipping tightens this upper bound when $C < B_g$. If the bound is below $|\delta_t|$, the score retains its sign throughout the interval. For AdamW, or when probing an individual component along a jointly trained trajectory, Eq. (38) instead uses the actual parameter displacements. These results provide sufficient conditions for interpreting sparse scores as interval-level trends; they do not establish that the checkpoint intervals used in our experiments satisfy these conditions.

E Limitations and Further Discussion

Endpoint probing examines whether locally induced target shaping persists as a stable supervisory signal, using final-model responses as empirical surrogates for implicit targets. These responses need not be optimal to provide a common reference for comparing components across checkpoints. Our findings characterize stable and phase-dependent target shaping relative to the evaluated in-distribution and out-of-distribution endpoints. Endpoint alignment alone does not determine task performance. In DPO, removing early negative learning or late positive learning reduced win rates despite their opposition to endpoint fitting, supporting a beneficial regularization-like role consistent with related work (Han et al., 2020). In PPO and GRPO, negative learning counteracts endpoint-aligned positive learning, consistent with an exploration-related role and prior uses of response suppression to encourage alternative generations (Chen et al., 2026). To the best of our knowledge, these interpretations offer the most plausible account of the observed dynamics given the available evidence, although they may be revised as the understanding of negative learning develops. The controlled-method results demonstrate the practical value of adjusting component strengths within the evaluated settings, without requiring these interpretations to be definitive.