Prompt Interference in LLM Post-Training: Why Pass@k Optimization Can Degrade Pass@1

Anas Barakat
Singapore University of Technology and Design
anas_barakat@sutd.edu.sg

Souradip Chakraborty
Massachusetts Institute of Technology
soura24@mit.edu

Khushbu Pahwa*
Amazon
pahwa16khu@gmail.com

Amrit Singh Bedi
University of Central Florida
amritbedi@ucf.edu

Abstract

Pass@k is a widely used performance metric for verifiable large language model tasks, including mathematical reasoning, code generation, and short-answer reasoning. It defines success if any of k independently sampled solutions passes a verifier. This multi-sample inference metric has motivated inference-aware fine-tuning methods that directly optimize pass@k. However, prior work reports a recurring trade-off: pass@k improves while pass@1 degrades under such methods. This trade-off is practically important because pass@1 often remains a hard operational constraint due to latency and cost budgets, imperfect verifier coverage, and the need for a reliable single-shot fallback. We study the origin of this trade-off and provide a theoretical characterization of when pass@k policy optimization can reduce pass@1 through gradient conflict induced by prompt interference. We show that pass@k policy gradients can conflict with pass@1 gradients because pass@k optimization implicitly reweights prompts toward low-success prompts; when these prompts are what we term *negatively interfering*, their upweighting can rotate the pass@k update direction away from the pass@1 direction. We illustrate our theoretical findings with large language model experiments on verifiable mathematical reasoning and coding tasks.

1 Introduction

Pass@k optimization. In many verifiable tasks, such as code generation and short-answer math, a system can afford multiple response attempts for the same prompt and check the correctness of each response attempt with an automatic verifier (e.g., unit tests, exact-match) that can check the correctness of generated answers. This naturally leads to *pass@k inference*: sample k independent candidate solutions and declare success if *any* candidate is correct. The corresponding metric, pass@k, measures the probability that at least one of k i.i.d. samples solves the prompt, and has become a standard performance measure when multiple attempts are available [6, 18, 13]. This "k attempts with verification" setting suggests that the training objective should align with deployment: if inference uses k attempts, training should directly optimize the same pass@k objective rather than optimizing only the single-sample expected reward. Recent work develops policy-gradient estimators and practical training procedures that optimize pass@k, and demonstrates large gains in pass@k performance on reasoning and code benchmarks [29, 32].

Trade-off: optimizing pass@k can reduce pass@1. A key complication is that pass@k optimization does not necessarily improve single-shot pass@1 performance. Empirically, some recent works

*Work is not related to position at Amazon.

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

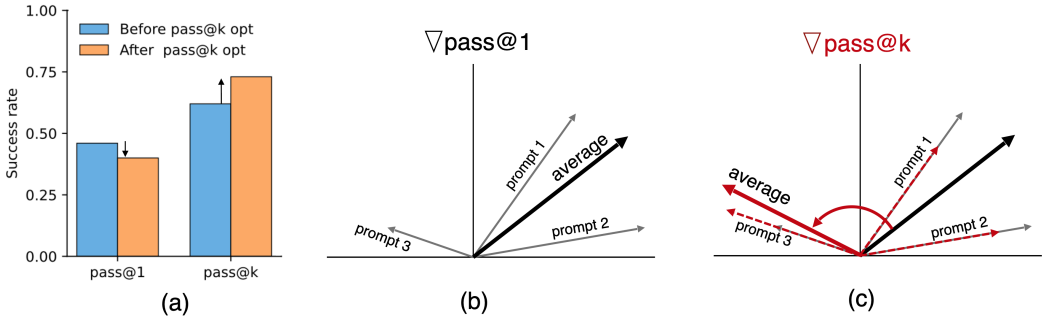


Figure 1: **(a) Empirical trade-off.** Under pass@k policy optimization, pass@k increases while pass@1 may decrease. We explain this empirically observed trade-off in **(b)** and **(c)**, which schematically illustrate the pass@1 and pass@k ($k > 1$) gradients for three prompts and their expectations in policy-parameter space. **(b) Pass@1 gradients with negatively interfering prompt.** This panel shows a setting in which prompt 3 is *negatively interfering* with prompts 1 and 2, i.e., the per-prompt pass@1 gradient for prompt 3 has negative inner product with the per-prompt pass@1 gradients for prompts 1 and 2 (see Section 3). Here, ∇pass@1 denotes the population pass@1 gradient, given by the average (expectation over prompts; here under a uniform distribution) of the per-prompt pass@1 gradients. **(c) Pass@k vs pass@1 gradient conflict.** Per-prompt pass@k gradients are scaled versions of the corresponding per-prompt pass@1 gradients (Eq. (2)). This reweighting amplifies the magnitude of the pass@k gradient for the negatively interfering prompt 3, causing the resulting population pass@k gradient to conflict with the population pass@1 gradient (their inner product becomes negative, corresponding to an obtuse angle, shown here as approximately $120°$). Hence, a policy update in the pass@k gradient direction can increase pass@k while decreasing pass@1.

report objective-dependent trade-offs where pass@k improves while pass@1 drops [29, 32]. For example, Walder and Karkhanis [32] explicitly highlight a trade-off between pass@1 and pass@k during code-generation fine-tuning (e.g. Tables 1 to 4 therein), and their results show that optimizing a pass@k objective can lower pass@1 even as higher-k metrics increase substantially (Table 1). This makes pass@k optimization risky in settings where single-shot accuracy is a hard requirement.

Why this degradation matters. Pass@1 remains operationally important: many deployments have limited latency or cost budgets, may not have reliable verifiers for every prompt, and often need a strong single-shot fallback even when retries are possible. In these regimes, a training procedure that improves pass@k at the cost of pass@1 can reduce overall system reliability and complicate model selection and deployment.

A per-prompt collinearity paradox. At a high level, one might expect pass@k and pass@1 to trade off simply because they are different objectives. However, this intuition is incomplete in the pass@k setting. For every fixed prompt, the pass@k gradient is exactly a nonnegative scalar multiple of the pass@1 gradient. Thus, there is no per-prompt directional conflict between the two objectives. The empirical degradation of pass@1 must therefore arise only after aggregating across prompts.

Open question. Despite a growing adoption of pass@k objectives, it is still not well understood *why* pass@k optimization can hurt pass@1, and *when* we should expect this trade-off to appear. Without a principled explanation, it is difficult to design reliable inference-aware fine-tuning methods that deliver multi-attempt gains while preserving strong single-shot performance. This leads to our central research question:

If pass@k and pass@1 are aligned on every individual prompt, when and why can pass@k policy optimization degrade population pass@1 performance?

Our key insight. We show that the conflict is instead a population-level effect. Compared to pass@1, optimizing the pass@k objective induces an implicit *prompt reweighting* toward prompts with lower success probability (i.e., prompts the current policy rarely solves). When these prompts contribute gradients that conflict with the population pass@1 gradient, upweighting them increases their influence on the pass@k policy gradient update. Consequently, the pass@k gradient can conflict with the pass@1 gradient direction. In this regime, pass@k improves under pass@k updates while pass@1 decreases (see Figure 1 for a schematic explanation). We identify the prompts that can lead to gradient conflict as negatively interfering using the concept of *prompt interference*.

Our main contributions are as follows:

• **Prompt interference.** We formalize prompt interference through a gradient-similarity kernel over prompts: two prompts interfere negatively when an update improving one tends to reduce the other's success probability. The induced distribution over negatively interfering prompts is crucial to understand the pass@k and pass@1 trade-off.

• **Pass@k-specific gradient conflict characterization.** Although per-prompt pass@k and pass@1 gradients are always positively collinear, we show that their population gradients can be conflicting in the sense that they can form an obtuse angle. This implies that a policy update following pass@k's policy gradient tends to increase pass@k while decreasing pass@1. We provide a characterization of this gradient conflict by establishing an interpretable expression for the inner product between pass@k and pass@1. This characterization crucially uses the fact that per-prompt pass@k gradients are *weighted* pass@1 per-prompt policy gradients. The gradient conflict expression shows that gradient conflict occurs when negatively interfering prompt regions are upweighted by the implicit pass@k reweighting and hence dominate the contribution of positively interfering prompts.

• **Sufficient conditions and influence of k.** Using our gradient conflict characterization, we provide sufficient conditions under which gradient conflict occurs. We further study the influence of the parameter k and show that increasing k encourages gradient conflict under some conditions in the relative probability of success in negatively versus positively interfering prompts.

• **Pass@1 degradation under pass@k updates.** We prove that pass@1 decreases while pass@k increases (simultaneously) under one-step pass@k policy updates satisfying an explicit stepsize condition. To show this result, we establish and use smoothness of the pass@k objective under a standard policy parameter regularity assumption.

• **Empirical validation.** We illustrate our main findings on large language models for solving math reasoning questions and code generation tasks.

Relation to multi-task gradient conflict. Our notion of prompt interference is related to ray interference and negative transfer in multi-task learning [26], as well as gradient-conflict methods such as gradient surgery [34]. However, the phenomenon studied here is not a direct instance of generic task-gradient conflict. In standard multi-task optimization, one typically studies arbitrary task objectives whose gradients may already disagree. In contrast, pass@1 and pass@k are two structured inference-time objectives induced by the same prompt-wise success probabilities, and for every fixed prompt their gradients are positively collinear. Thus the central question is different: how can two per-prompt aligned objectives become conflicting after averaging over prompts? Standard per-task gradient conflict alone cannot explain pass@1 degradation. We show that the answer is the pass@k-specific weight $w_{k,\theta}(x) = k(1 - p_\theta(x))^{k-1}$, which shifts the effective prompt distribution toward low-success prompts. Gradient conflict arises only when this structured reweighting amplifies prompts that negatively interfere with population pass@1 improvement. Our contribution is therefore to characterize when this pass@k-induced distribution shift turns prompt interference into pass@1 degradation. See Appendix A for an extended related work discussion.

2 Preliminaries

In this section, we formally define the pass@k objective and recall its policy gradients as used in pass@k policy optimization. We consider a setting where a language model is represented as a stochastic policy π_θ parameterized by weights $\theta \in \mathbb{R}^d$ over a set of responses $\mathcal{Y}$ given a set of prompts denoted by $\mathcal{X}$ (e.g., math questions). The conditional probability distribution of the model's output responses when prompted with x is $\pi_\theta(\cdot|x)$. We denote by $\mathcal{D}$ the distribution over prompts.

Pass@k objective. Given a prompt x sampled from a distribution $\mathcal{D}$, a single 'attempt' is a response $y \sim \pi_\theta(\cdot|x)$. We suppose we have access to a binary reward verifier $r(x, y) \in \{0, 1\}$ which encodes correctness of response y for prompt x. The per-prompt success probability for any $x \in \mathcal{X}$ and any policy parameter $\theta \in \mathbb{R}^d$ will be denoted:

$$p_\theta(x) := \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x, y)] = \mathbb{P}_{y\sim\pi_\theta(\cdot|x)}(r(x, y) = 1)\,, \tag{1}$$

where the last identity follows from the fact that rewards are binary ($r(x, y) \in \{0, 1\}$). The pass@k objective is defined as the probability that at least one response is correct among k responses, i.e.,

$$J_k(\theta) := \mathbb{P}\left(\max_{1 \le i \le k} r(x, y_i) = 1\right) = \mathbb{E}_{x \sim \mathcal{D}}[1 - (1 - p_\theta(x))^k], \tag{pass@k}$$

where $x \sim \mathcal{D}$, $y_i \sim \pi_\theta(\cdot|x)$ for $i = 1, \cdots k$ are k independent response samples. Hence pass@k can be written using the nonlinear transformation $f_k(p) := 1 - (1 - p)^k$ as $J_k(\theta) = \mathbb{E}_{x \sim \mathcal{D}}[f_k(p_\theta(x))]$. It immediately follows that pass@k is an increasing function in k: $\forall m \ge k \ge 1, \forall \theta \in \mathbb{R}^d, J_m(\theta) \ge J_k(\theta)$. Moreover, pass@k can be bounded by pass@1 using Jensen's inequality and concavity of f_k w.r.t. its argument: for all $k \ge 1, \theta \in \mathbb{R}^d$, $J_1(\theta) \le J_k(\theta) \le \min(1, f_k(J_1(\theta)))$. Note that under our notation $J_1(\theta) = \mathbb{E}_{x \sim \mathcal{D}}[p_\theta(x)]$.

Pass@k policy gradients. Pass@k optimization can be performed using pass@k policy gradients. It follows from the chain rule that the gradient of pass@k w.r.t. the policy parameters is given by:

$$\nabla J_k(\theta) = \mathbb{E}_{x \sim \mathcal{D}}[w_k(p_\theta(x)) \nabla p_\theta(x)], \, w_k(p) := k(1 - p)^{k-1}, \tag{2}$$

as noted in a few recent works [20, 35]. Pass@k gradients reweight prompts by

$$w_{k,\theta}(x) := w_k(p_\theta(x)) = k(1 - p_\theta(x))^{k-1}. \tag{3}$$

This weighting emphasizes low-probability of success prompts and downweights prompts with high probability of success. To compute pass@k gradients in (2), note that the gradient $\nabla p_\theta(x)$ of the prompt-wise probability of success for any prompt $x \in \mathcal{X}$ and any policy parameter $\theta \in \mathbb{R}^d$ immediately follows from (1) and is given by:

$$\nabla p_\theta(x) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[r(x, y) s_\theta(x, y)], \tag{4}$$

where $s_\theta(x, y) := \nabla \log \pi_\theta(y|x)$ denotes the score function.

This gradient expression has been recently used in [29, 32] for instance. Per-prompt pass@k and pass@1 are positively collinear as $\nabla J_k(x; \theta) := w_{k,\theta}(x) \nabla p_\theta(x) = w_{k,\theta}(x) \nabla J_1(x; \theta)$ and $w_{k,\theta}(x) \ge 0$. However we will crucially see that pass@k and pass@1 as *expectation over prompts* are not necessarily positively collinear. This discrepancy, which was overlooked in the literature, highlights the importance of the expectation over prompts and the structure of per-prompt pass@1 gradients. This motivates our next main section in which we formalize the concept of *prompt interference* capturing the potential conflict between the diversity of prompts in the underlying prompt distribution and the shared policy parameters across prompts.

3 Prompt Interference

In this section, we formalize the phenomenon of *prompt interference* which will play an important role in explaining later in the paper why pass@1 can degrade under pass@k policy updates. This section is focused on understanding the structure of pass@1 policy gradients over the set of prompts. The role of the pass@k objective and its influence on optimization will be discussed in section 4.

The pass@1 objective is defined as the expectation of the per-prompt probability of success with respect to the distribution of prompts (see pass@k with $k = 1$). As a function of policy parameters, this pass@1 objective induces a shared-parameter coupling across prompts: the same policy parameters are shared across the entire distribution of prompts which might be diverse. As a consequence, a (shared) policy parameter update can improve the probability of success ($p_\theta(x)$) on some prompts while it can degrade it on other prompts. When this negative transfer occurs, we say that there is negative prompt interference. We provide a formal definition of this phenomenon in this section.

To define prompt interference, we investigate the structure of the pass@1 policy gradient, looking at the *per-prompt* pass@1 policy gradients and comparing them across prompts. We define the following natural prompt gradient similarity measure $\kappa_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined for any policy parameter $\theta \in \mathbb{R}^d$ and any prompts $x, x' \in \mathcal{X}$ by:

$$\kappa_\theta(x, x') := \langle \nabla p_\theta(x), \nabla p_\theta(x') \rangle . \tag{5}$$

This kernel captures similarities between prompts in terms of their pass@1 gradients which encode their ability to improve the probability of success. Using the policy gradient expressions of the gradients of pass@1 in (4), we obtain:

$$\kappa_\theta(x, x') = \mathbb{E}_{y \sim \pi_\theta(\cdot|x) y' \sim \pi_\theta(\cdot|x')}\left[\delta(x, y, x', y') \langle s_\theta(x, y), s_\theta(x', y') \rangle\right], \tag{6}$$

where $\delta(x, y, x', y') := r(x, y)\, r(x', y')$ which is equal to 1 if and only if both y, y' are correct completions for x, x' respectively (as rewards are binary). The kernel informs on whether improving pass@1 on one prompt tends to also improve pass@1 on another prompt under the shared policy parameters. To see this, consider a policy update $\theta' = \theta + \eta \nabla p_\theta(x)$ with a small stepsize $\eta > 0$ in the direction $\nabla p_\theta(x)$ improving the probability of success on x. A first-order Taylor expansion of $\theta \mapsto p_\theta(x')$ gives $p_{\theta^+}(x') \simeq p_\theta(x') + \eta \langle \nabla p_\theta(x'), \nabla p_\theta(x) \rangle = p_\theta(x') + \eta \kappa_\theta(x', x)$. This leads us to the following prompt interference definition.

Definition 3.1 (Prompt interference)**.** Two prompts x and x' *interfere positively* if $\kappa_\theta(x', x) > 0$: a small update that increases $p_\theta(x)$ also tends to increase $p_\theta(x')$. They *interfere negatively* if $\kappa_\theta(x', x) < 0$, in which case increasing $p_\theta(x)$ tends to decrease $p_\theta(x')$.

This notion parallels ray interference in multi-task learning [26, Definition 1], but here the gradients are prompt-wise pass@1 gradients in LLM post-training. See Appendix A for further discussion on related work. We next illustrate prompt interference on a toy example.

Prompt interference in a toy example. In Appendix C, we consider a minimal contextual-bandit abstraction of verifier-based multi-sample LLM evaluation as a running example to illustrate (negative) prompt interference and the pass@k-pass@1 trade-off. The detailed description of this example and a visualization of the cosine kernel heatmap are given in Appendix C (see Figure 5). In this example, we demonstrate that pass@1 may decrease while pass@k increases under pass@k optimization. We show that negative prompt interference can induce a conflict between pass@1 and pass@k gradients when it is amplified by the induced pass@k prompt reweighting. In the next section, we go beyond our toy example to characterize this conflict between pass@k and pass@1 gradients in general.

4 Pass@k and Pass@1 Gradients Conflict

In this section, we exploit the observation that pass@k policy optimization induces an implicit prompt reweighting, biased toward upweighting low-success prompts. To connect with the previous section, if this reweighting coincides with upweighting negatively interfering prompts, then pass@k policy updates may result in conflicting pass@k and pass@1 gradients, leading to pass@1 degradation under pass@k policy optimization updates. We make this chain of arguments precise in this section.

4.1 Gradient conflict: Reweighting meets interference

To explain why optimizing pass@k can hurt pass@1, we study a key quantity: the angle between the policy gradients of pass@k and pass@1 respectively. We show that this angle can be obtuse and we refer to this setting as a case of *conflicting gradients*.

This observation might seem a little surprising from the theory viewpoint at first glance. Indeed, the per-prompt gradient $\nabla J_k(x; \theta)$ of pass@k is always a positive multiple of the per-prompt policy gradient $\nabla J_1(x; \theta)$ of pass@1 since $\nabla J_k(x; \theta) = w_{k,\theta}(x) \nabla J_1(x; \theta)$ and $w_{k,\theta}(x) \geq 0$ for any prompt $x \in \mathcal{X}$. Nevertheless, the population gradients $\nabla J_k(\theta) = \mathbb{E}[\nabla J_k(x; \theta)]$ and $\nabla J_1(\theta) = \mathbb{E}[\nabla J_1(x; \theta)]$ need not form an acute angle as their inner product can be negative as we will see.

The following proposition provides an explicit expression of this inner product to characterize the gradients conflict. To state this result, we introduce the prompt-wise gradient agreement score for pass@1 defined for any policy parameter $\theta \in \mathbb{R}^d$ and any prompt $x \in \mathcal{X}$ as follows:

$$a_\theta(x) := \langle \nabla J_1(x; \theta), \nabla J_1(\theta) \rangle = \mathbb{E}_{x' \sim \mathcal{D}}[\kappa_\theta(x, x')] \,. \tag{7}$$

This agreement score quantifies the prompt-level gradient interference with the average pass@1 objective. If $a_\theta(x) > 0$, that prompt's pass@1 gradient points in a direction that also improves average pass@1. If $a_\theta(x) < 0$, it is anti-aligned and improving that prompt's pass@1 tends to decrease population pass@1 under the shared policy parameters θ . Using this agreement score, our characterization is as follows.

Proposition 4.1 (Gradients conflict characterization). *For any $k \geq 1$ and any $\theta \in \mathbb{R}^d$,*

$$\langle \nabla J_k(\theta), \nabla J_1(\theta) \rangle = \mathbb{E}_{x\sim\mathcal{D}}[w_{k,\theta}(x) a_\theta(x)] = \mathbb{E}[w_{k,\theta}(x)] \cdot \|\nabla J_1(\theta)\|^2 + \mathrm{cov}(w_{k,\theta}(x), a_\theta(x)) \,. \tag{8}$$

In particular, $\langle \nabla J_k(\theta), \nabla J_1(\theta) \rangle < 0$ is equivalent to each one of the following conditions: (i) $\mathbb{E}_{x\sim\mathcal{D}}[w_{k,\theta}(x) a_\theta(x)] < 0$, and (ii) $\mathrm{cov}(w_{k,\theta}(x), a_\theta(x)) < -\mathbb{E}[w_{k,\theta}(x)] \cdot \|\nabla J_1(\theta)\|^2$.

Discussion of conditions. We provide two interpretations explaining gradients conflict and we refer the reader to Figure 2 below for illustrations on the toy example of section C, see also Figure 6 in Appendix C for a contour plot of pass@1 and pass@k objectives in the policy parameter space showing the gradient conflict region in 2D.

(i) Prompt distribution shift: pass@k reweights prompts. Recalling that the weights $w_{k,\theta}(x)$ are nonnegative and assuming $\mathbb{E}_{\mathcal{D}}[w_{k,\theta}(x)] > 0$, define the reweighted prompt distribution $\tilde{\mathcal{D}}_{k,\theta}(dx) \propto w_{k,\theta}(x)\mathcal{D}(dx)$ inducing the expectation $\mathbb{E}_{\tilde{\mathcal{D}}_{k,\theta}}[f(x)] = \frac{\mathbb{E}_{\mathcal{D}}[w_{k,\theta}(x) f(x)]}{\mathbb{E}_{\mathcal{D}}[w_{k,\theta}(x)]}$ for any measurable function f of the prompt random variable x. Note that the denominator is nonzero as the weights are not equal to zero a.e. The distribution $\tilde{\mathcal{D}}_{k,\theta}$ places higher mass on prompts with smaller success probability $p_\theta(x)$ (hard prompts) since $w_{k,\theta}(x)$ is decreasing in $p_\theta(x)$. Therefore, we can rewrite (8) as follows:

$$\langle \nabla J_k(\theta), \nabla J_1(\theta) \rangle = \mathbb{E}_{\mathcal{D}}[w_{k,\theta}(x)] \cdot \mathbb{E}_{\tilde{\mathcal{D}}_{k,\theta}}[a_\theta(x)] \,, \tag{9}$$

and since $\mathbb{E}_{\mathcal{D}}[w_{k,\theta}(x)] > 0$ ($w_{k,\theta}(x) \geq 0$ and not equal to zero a.e.*), we obtain the following equivalence between $\langle \nabla J_k(\theta), \nabla J_1(\theta) \rangle < 0$ and $\mathbb{E}_{\tilde{\mathcal{D}}_{k,\theta}}[a_\theta(x)] < 0$. This characterization shows that conflict of gradients of pass@1 and pass@k occurs if and only if the average of the agreement score $a_\theta(x)$ under the pass@k induced prompt distribution $\tilde{\mathcal{D}}_{k,\theta}$ is negative. In contrast note that $\mathbb{E}_{\mathcal{D}}[a_\theta(x)] = \|\nabla J_1(\theta)\|^2 \geq 0$. Pass@k changes the prompt weighting from $\mathcal{D}$ to $\tilde{\mathcal{D}}_{k,\theta}$ and the average agreement under this shifted distribution can become negative.

In summary, we obtain the following causal chain: pass@k induces a prompt reweighting toward harder prompts (i.e. lower $p_\theta(x)$ since $w_{k,\theta}(x)$ decreases in $p_\theta(x)$). If hard prompts have negative agreement score $a_\theta(x)$ (i.e. per-prompt pass@1 gradients are conflicting with the pass@1 gradient) then the reweighted average agreement becomes negative and the gradient of pass@k and pass@1 form an obtuse angle as a consequence, i.e. they are conflicting.

(ii) Covariance between weights and agreement scores. The covariance $\mathrm{cov}(w_{k,\theta}(x), a_\theta(x))$ captures the coupling between 'how much pass@k emphasizes a prompt' (via weight $w_{k,\theta}(x)$) and 'whether that prompt helps pass@1' (via alignment score $a_\theta(x)$). The covariance term quantifies whether pass@k's hardness reweighting concentrates gradient mass on prompts whose success-probability gradients are anti-aligned with the pass@1 gradient. Conflict happens when this anti-aligned hard-prompt mass overwhelms the positive term. See Appendix D for more details.

Degenerate weights: For $k \geq 2, \mathbb{E}[w_{k,\theta}(x)] = 0$ occurs exactly when $p_\theta(x) = 1$ for almost every prompt x. This means that the model is perfect for almost every prompt and there is no need to perform policy optimization. In this case, $\nabla J_k(\theta) = 0$ and the normalized reweighted distribution is undefined. The unnormalized identity in Proposition 4.1 remains valid without this condition.

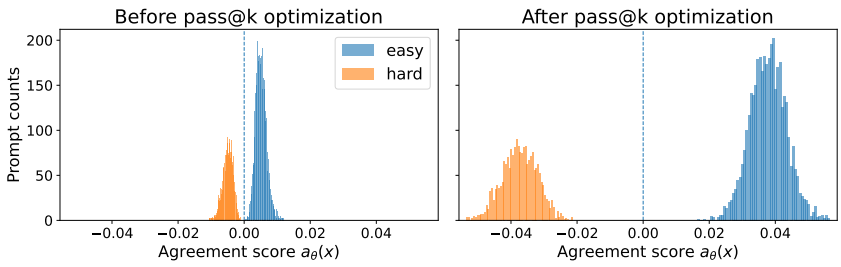


Figure 2: The prompt distribution has prompts with negative and positive agreement scores $a_\theta(x)$ (7) on the toy example (Appendix C). Prompts with negative scores are responsible for gradient conflict and hence pass@1 decrease. After pass@k optimization, negative scores get even more negative.

Remark 4.2. For a more general discussion of the conflict between pass@k and pass@m gradients for any $k, m \geq 1$ using our kernel viewpoint, see Appendix E.

4.2 Sufficient conditions and threshold in k

We now provide sufficient conditions for pass@k and pass@1 gradients conflict, as a corollary of Proposition 4.1. First, we make a standard smoothness assumption on the policy parameterization.

Assumption 4.3 (Lipschitz and smooth policy)**.** There exist $G, F > 0$ s.t. for all prompts $x \in \mathcal{X}$ and all $\theta \in \mathbb{R}^d$, we have the expected score bounds: $\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[\|\nabla \log \pi_\theta(y|x)\|^2] \leq G^2$ and $\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[\|\nabla^2 \log \pi_\theta(y|x)\|] \leq F$.

This is a standard assumption used in the convergence analysis of policy gradient methods, see, e.g., [36, 22], and satisfied by Gaussian policies with bounded feature vectors and softmax tabular policies.

To quantify the magnitude of negative interference and its prevalence under the distribution $\mathcal{D}$, we define the m-strongly negatively interfering prompt set with margin $m > 0$:

$$\mathcal{X}_-(\theta, m) := \{x \in \mathcal{X} : a_\theta(x) \leq -m\}\,, \tag{10}$$

and denote by $q_\theta := \mathcal{D}(\mathcal{X}_-(\theta, m)) \in [0, 1]$ its probability mass under the distribution $\mathcal{D}$ of prompts. Then, the expected pass@k weight contributions inside and outside the m-strongly interfering set $\mathcal{X}_-(\theta, m)$ are given by:

$$W_-(k, \theta) := \mathbb{E}[w_{k,\theta}(x)\mathbb{1}_{\{x\in\mathcal{X}_-(\theta,m)\}}]\,, \quad W_+(k, \theta) := \mathbb{E}[w_{k,\theta}(x)\mathbb{1}_{\{x\notin\mathcal{X}_-(\theta,m)\}}]\,. \tag{11}$$

The following corollary of Proposition 4.1 shows that gradients of pass@1 and pass@k conflict when the contribution of pass@k weights in negatively interfering prompt regions dominates its counterpart for positively interfering prompts.

Corollary 4.4 (Dominating negatively interfering prompts)**.** *Let Assumption 4.3 hold. Suppose that $q_\theta > 0$ and let $k \geq 2$. Then for any $\theta \in \mathbb{R}^d$, $\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle \leq -\delta(\theta)$ where $\delta(\theta) := mW_-(k, \theta) - G^2 W_+(k, \theta)$. If in addition $\delta(\theta) > 0$, then $\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle \leq -\delta(\theta) < 0$.*

Discussion of assumption. Some negative agreement is necessary for gradient conflict. Indeed, if instead we have for instance for almost every $x \in \mathcal{X}, a_\theta(x) \geq 0$, then it follows from (8) that $\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle \geq 0$ as the weights $w_{k,\theta}(x)$ are also always nonnegative. Negative agreement score is a necessary condition for gradient conflict. The gradient conflict condition of Corollary 4.4 depends on the relative weights ratio $W_-(k, \theta)/W_+(k, \theta)$ in negative vs positive prompt agreement score regions. This ratio is a function of both k and the probability of success $p_\theta(x)$ (1) under policy π_θ . We now exploit the closed form of the pass@k weights $w_{k,\theta}(x) := k(1 - p_\theta(x))^{k-1}$ to evaluate the ratio under assumptions on the prompt-wise probabilities of success. The next result shows that beyond a threshold $k^\star$, the pass@k and pass@1 gradients form an obtuse angle. We make a separation assumption on the probability of success over prompts: the probability of success for m-strongly interfering prompts is supposed to be uniformly smaller than that of the remaining prompts. Intuitively, this boils down to assume that negatively interfering prompts are relatively harder to solve than the rest of the prompts. This assumption may not always hold.

Proposition 4.5 (Influence of k)**.** *In the setting of corollary 4.4, assume there exist $\varepsilon, \delta, q \in (0, 1)$ s.t. $\sup_{x\in\mathcal{X}_-(m,\theta)} p_\theta(x) \leq \varepsilon < \delta \leq \inf_{x\notin\mathcal{X}_-(m,\theta)} p_\theta(x)$, and $q_\theta \geq q$ for any θ, then $\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle \leq -\Delta$ where $\Delta = k[(1 - \varepsilon)^{k-1} mq - (1 - \delta)^{k-1} G^2(1 - q)]$. If in addition $k > k^\star := 1 + \frac{\log(\frac{(1-q)G^2}{qm})}{\log(\frac{1-\varepsilon}{1-\delta})}$, then $\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle < 0$.*

If k is large enough then gradients of pass@k and pass@1 will be conflicting and pass@1 will degrade while pass@k will improve under a small step size policy update following pass@k's gradient. The threshold $k^\star$ becomes smaller when the probability mass of the negatively interfering region is larger. Hence, under larger negative prompt interference, the pass@k-pass@1 conflict becomes more severe.

4.3 Pass@1 degradation under pass@k updates

In this section, we show and quantify pass@1 decrease and pass@k increase under pass@k policy optimization, exploiting the conflict between pass@k and pass@1 gradients shown in the previous sections. To quantify improvement and degradation under policy updates, we prove smoothness of the pass@k objective for any $k \geq 1$ under Assumption 4.3 (see Lemma F.1 in appendix F.4). Using this smoothness property, we show that gradient conflict results in provable pass@1 decrease under pass@k optimization.

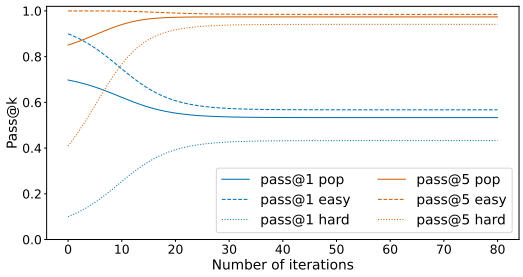


Figure 3: Pass@5 increases while pass@1 decreases under pass@5 optimization in example of section C, 'pop' refers to 'population' i.e. pass@k as defined in (pass@k), 'easy' (resp. 'hard') means expectation is only taken over prompts labeled as easy (resp. hard).

Proposition 4.6 (Pass@1 decrease under pass@k updates)**.** *In the setting of corollary 4.4, set* $\theta^+ = \theta + \eta \nabla J_k(\theta)$ *where* $k \geq 2$ *and* $\eta > 0$*. If* $0 < \eta < \min(\frac{\delta(\theta)}{C_2}, \frac{1}{L_k})$ *where* $C_2 := k^2 G^2 L_1/2$*,*

$$J_1(\theta^+) \leq J_1(\theta) - \eta\delta(\theta) + C_2\eta^2 < J_1(\theta)\,, \quad J_k(\theta^+) \geq J_k(\theta) + \frac{\eta}{2}\|\nabla J_k(\theta)\|^2 \geq J_k(\theta)\,. \tag{12}$$

We conclude this section with additional practical insights derived from our theoretical investigation.

Practical implications. Our theory suggests three diagnostics:

(i) *Check interference:* pass@k optimization is risky when low-success prompts have negative agreement with the pass@1 gradient.

(ii) *Tune k cautiously:* larger k strengthens the reweighting and can amplify conflict *when* such prompts are negatively interfering.

(iii) *Inspect prompt subsets:* two datasets with the same average pass@1 can behave differently; what matters is whether the low-success prompts upweighted by pass@k are aligned or anti-aligned with population pass@1 improvement. This suggests clipped/interference-aware reweighting or multi-objective updates as mitigation.

5 Experiments

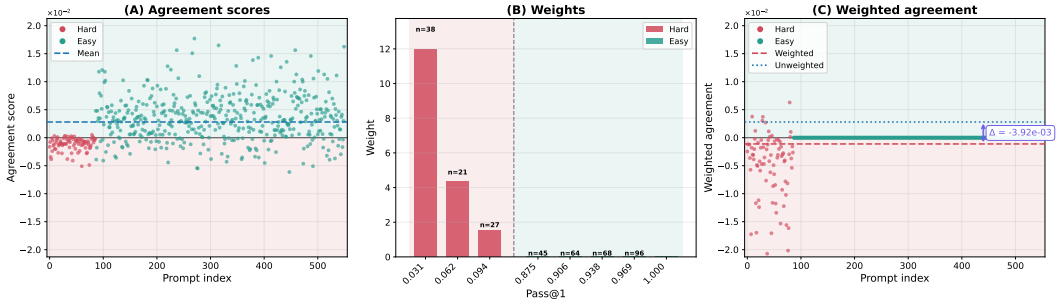
We empirically test whether the pass@k objective can induce gradient conflict with pass@1 on math reasoning, as predicted by Proposition 4.1. We provide further evidence on code generation (HumanEval [6] and MBPP [3]) in Appendix G.4.

Dataset and models. We use the MATH dataset [14], which contains competition-style high school math problems spanning seven subjects (Algebra, Counting & Probability, Geometry, Intermediate Algebra, Number Theory, Precalculus, and Prealgebra). We randomly sample 2,000 problems for evaluation. We run experiments with two reasoning models: DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Qwen-7B.

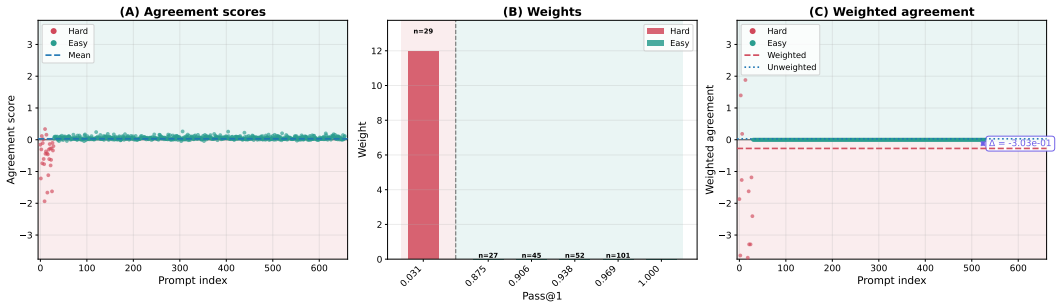
Pass@k gradient computation. For computational efficiency, we compute pass@1 gradients with respect to policy parameters in the language model's final hidden layer (dimension $d = 4096$ for Llama-8B, $d = 3584$ for Qwen-7B). We compute pass@k gradients using Monte Carlo estimates based on (2) and pass@k estimates developed in prior work [32, 6].

Setup. We create filtered data sets $\mathcal{D}_{\delta_1,\delta_2}$ of prompts and responses with varying difficulty thresholds (δ_1, δ_2) consisting of (i) hard prompts ($p_\theta(x_i) < \delta_2$) and (ii) easy prompts ($p_\theta(x_i) > \delta_1$). We test different combinations with $\delta_1 \in \{0.80, 0.85, 0.90\}$ and $\delta_2 \in \{0.05, 0.10, 0.15\}$. For each combina-

tion, we compute estimates of the agreement scores $a_\theta(x)$ defined in (7), pass@k weights $w_{k,\theta}(x)$ (see (3)) and estimated average weighted agreement scores over prompts, which corresponds to pass@k-pass@1 gradient inner product by Proposition 4.1. See Appendix G.2 for all threshold configurations.



(a) DeepSeek-R1-Distill-Llama-8B: $\delta_1 = 0.85$, $\delta_2 = 0.10$ (550 prompts: 86 hard, 464 easy, ratio 5.4:1)



(b) DeepSeek-R1-Distill-Qwen-7B: $\delta_1 = 0.85$, $\delta_2 = 0.05$ (656 prompts: 29 hard, 627 easy, ratio 21.6:1)

Figure 4: **Pass@k-induced reweighting produces population-gradient conflict in filtered MATH subsets.** Both models demonstrate how pass@k reweighting causes gradient conflict with pass@1, despite different threshold settings. Agreement scores are computed relative to the pass@1 gradient of filtered prompts. **Left column (A):** Agreement scores show separation between hard negatives (red) and easy positives (green), with unweighted means shown as blue dashed lines. **Middle column (B):** Pass@k weights grouped by pass@1 probability, showing extreme weight disparity ($\sim 10^{28}$:1) between hard and easy prompts. **Right column (C):** Weighted contributions (estimates of pass@k vs pass@1 inner products) demonstrate how pass@k reweighting induces negative pass@k/pass@1 gradient alignment. *Purple arrows highlight the critical downward shift* from the unweighted mean (blue dotted, same as plot A) to the weighted mean (red dashed). **(a) DeepSeek-Llama:** The *downward shift* from $+2.80 \times 10^{-3}$ to -1.12×10^{-3} ($\Delta = -3.92 \times 10^{-3}$) flips the gradient from positive to negative, resulting in inner product -0.613. **(b) DeepSeek-Qwen:** The *dramatic downward shift* from $+2.97 \times 10^{-2}$ to -2.74×10^{-1} ($\Delta = -3.04 \times 10^{-1}$) demonstrates even stronger gradient conflict. Despite only 29 hard prompts vs 627 easy (21.6:1 ratio), the extreme weight disparity causes hard negatives to completely dominate, resulting in inner product -181.

Discussion. Our experiments validate the theoretical predictions in several ways. First, the agreement scores (column A in Figure 4) show clear separation between hard prompts (red points, negative agreement scores clustered below zero) and easy prompts (green points, positive agreement scores above zero), confirming that prompt interference exists in practice. The unweighted mean agreement (blue dashed line) is positive, indicating that under uniform weighting, the population gradient would improve pass@1. Second, the pass@k weights (column B in Figure 4) reveal extreme disparity in how pass@k values different prompts. The histogram shows that hard prompts (red bars, low pass@1 values $\lesssim 0.1$) receive weights of $n = 38$ at scale 12, $n = 21$ at scale 4, and $n = 27$ at scale 1, while easy prompts (green bars, high pass@1 values $\gtrsim 0.85$) receive negligible weights near 10^{-28}. This creates weight ratios exceeding $10^{28} : 1$, demonstrating the extreme reweighting mechanism our theory identifies: pass@k dramatically amplifies the influence of low-success prompts while effectively ignoring high-success prompts. Third, column C demonstrates the consequence of this reweighting through weighted agreement scores. The purple arrows in panel C highlight the critical

downward shift from the unweighted mean (blue dotted line, matching column A) to the weighted mean (red dashed line). For Llama-8B (Figure 4a), this shift is $\Delta = -3.92 \times 10^{-3}$, moving from $+2.80 \times 10^{-3}$ to -1.12×10^{-3} and flipping the gradient alignment from positive to negative, resulting in an inner product of -0.613. For Qwen-7B (Figure 4b), despite having only 29 hard prompts versus 627 easy prompts (a 21.6:1 ratio heavily favoring easy prompts), the extreme weight disparity causes an even more dramatic shift of $\Delta = -3.04 \times 10^{-1}$, moving from $+2.97 \times 10^{-2}$ to -2.74×10^{-1}, yielding a strongly negative inner product of -181.

This validates Proposition 4.1: when the negative covariance between pass@k weights $w_{k,\theta}(x)$ and agreement scores $a_\theta(x)$ dominates the positive term $\mathbb{E}[w_{k,\theta}(x)] \cdot \|\nabla J_1(\theta)\|^2$, gradient conflict occurs. The visualization supports the proposed mechanism: pass@k upweights hard prompts (column B) $\rightarrow$ these hard prompts exhibit negative interference $a_\theta(x) < 0$ (column A) $\rightarrow$ the reweighted gradient conflicts with pass@1's gradient (column C). The robustness across threshold configurations (see Appendix G.2, Figure 7) demonstrates this is not an artifact of threshold selection.

Additional pass@k policy optimization experiment. We performed end-to-end pass@8 policy optimization on Qwen2.5-Math-1.5B-Instruct using on-policy RL with verifiable rewards. We construct the training pool from the train split of ricdomolm/MATH-500, which contains approximately 12,000 MATH problems after excluding the 500-problem MATH-500 evaluation set. For each of the five independent training runs, we randomly sample 2,000 training problems using the run seed, while evaluation is performed on all 500 problems from the held-out HuggingFaceH4/MATH-500 benchmark. We additionally verify that no normalized problem text is shared between the sampled training pool and the held-out evaluation set.

We optimize rank-16 LoRA adapters inserted into the `q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj` and `down_proj` modules of transformer layers, resulting in 2,637,824 trainable parameters. The model is trained for 500 optimization steps using a pass@8 objective with binary correctness rewards. At checkpoints $\{0, 125, 250, 375, 500\}$, we evaluate using 32 sampled responses for each of 500 fixed held-out MATH problems. Pass@1 is computed as the mean correctness, while pass@8 is estimated using the standard unbiased combinatorial estimator. To reduce Monte Carlo variability, every checkpoint within a run is evaluated using the same held-out prompts, sampling configuration, and random seed, allowing paired comparisons relative to the step 0 checkpoint.

At every checkpoint, we recompute the gradient interference diagnostic using the same responses to estimate success probabilities and reward-weighted score-function gradients, all with respect to the trainable LoRA parameters. We report the results of the experiment in Table 1.

Table 1: **Pass@k optimization can hurt pass@1 performance**: under pass@8 policy optimization, pass@8 improves by $+1.65 \pm 0.49$ percentage points, while pass@1 decreases by 2.40 ± 0.49 percentage points relative to initialization. Consistent with our theory, the population-gradient cosine becomes negative early during optimization and remains negative as pass@1 degrades. Mean $\pm 95\%$ confidence interval reported during training across 5 independent runs.

Training step	pass@1	pass@8	Cosine similarity(∇pass@1, ∇pass@8)
0	0.7007 ± 0.0027	0.8380 ± 0.0016	$+0.032 \pm 0.028$
125	0.6979 ± 0.0023	0.8448 ± 0.0021	-0.066 ± 0.039
250	0.6919 ± 0.0024	0.8501 ± 0.0019	-0.154 ± 0.044
375	0.6846 ± 0.0021	0.8503 ± 0.0017	-0.133 ± 0.053
500	$\mathbf{0.6767 \pm 0.0039}$	$\mathbf{0.8545 \pm 0.0025}$	$\mathbf{-0.246 \pm 0.044}$

6 Conclusion and Future Work

We studied why pass@k policy optimization can degrade pass@1 and showed that the conflict arises when pass@k reweighting amplifies negatively interfering prompts. Central to our analysis is the concept of *prompt interference*: because a language model shares parameters across a heterogeneous prompt distribution, improving some prompts can adversely affect others. More broadly, prompt interference provides a useful lens for understanding post-training objectives that redistribute optimization effort across prompts, suggesting both practical diagnostics and interference-aware mitigation methods. Understanding how broadly this phenomenon appears across models, tasks, and objectives is an important direction for future work.

Acknowledgments and Disclosure of Funding

We thank Aviral Kumar for pointing us to related work on negative interference between easy and hard problems, pass@k policy optimization [24, 25], and prior work on ray interference in deep reinforcement learning [26].

We used ChatGPT during the development of this project to assist with coding, the development of the toy example, and brainstorming and refining theoretical ideas. The manuscript, including all mathematical claims and proofs, was written and verified by the authors, with ChatGPT used for suggestions on selected technical points, wording, and presentation. The authors take full responsibility for the content of this manuscript.

References

[1] Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. Variational best-of-n alignment. In *International Conference on Learning Representations*, 2025. (Cited on p. 15)

[2] Gholamali Aminian, Idan Shenfeld, Amir R Asadi, Ahmad Beirami, and Youssef Mroueh. Best-of-n through the smoothing lens: Kl divergence and regret analysis. *arXiv preprint arXiv:2507.05913*, 2025. (Cited on p. 15)

[3] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. (Cited on p. 8)

[4] Ananth Balashankar, Ziteng Sun, Jonathan Berant, Jacob Eisenstein, Michael Collins, Adrian Hutter, Jong Lee, Chirag Nagpal, Flavien Prost, Aradhana Sinha, Ananda Theertha Suresh, and Ahmad Beirami. Infalign: Inference-aware language model alignment. In *International Conference on Machine Learning*, 2025. (Cited on p. 15)

[5] Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander Nicholas D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *International Conference on Machine Learning*, 2025. (Cited on p. 15)

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. (Cited on pp. 1, 8, and 15)

[7] Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025. (Cited on p. 15)

[8] Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Aviral Kumar, Rishabh Agarwal, Sridhar Thiagarajan, Craig Boutilier, and Aleksandra Faust. Inference-aware fine-tuning for best-of-n sampling in large language models. In *International Conference on Learning Representations*, 2025. (Cited on p. 15)

[9] Damek Davis and Benjamin Recht. What is the objective of reasoning with reinforcement learning? *arXiv preprint arXiv:2510.13651*, 2025. (Cited on p. 15)

[10] 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 *Annual Conference on Neural Information Processing Systems*, 2025. (Cited on p. 15)

[11] Qiwei Di, Kaixuan Ji, Xuheng Li, Heyang Zhao, and Quanquan Gu. Best-of-majority: Minimax-optimal strategy for pass@k inference scaling. *arXiv preprint arXiv:2510.03199*, 2025. (Cited on p. 15)

[12] Weihua Du, Yiming Yang, and Sean Welleck. Optimizing temperature for language models with multi-sample inference. *arXiv preprint arXiv:2502.05234*, 2025. (Cited on p. 15)

[13] Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Taco Cohen, and Gabriel Synnaeve. RLEF: Grounding code LLMs in execution feedback with reinforcement learning. In *International Conference on Machine Learning*, 2025. (Cited on p. 1)

[14] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *Advances in Neural Information Processing Systems*, 2021. (Cited on p. 8)

[15] Yuhua Jiang, Jiawei Huang, Yufeng Yuan, Xin Mao, Yu Yue, Qianchuan Zhao, and Lin Yan. Risk-sensitive rl for alleviating exploration dilemmas in large language models. *arXiv preprint arXiv:2509.24261*, 2025. (Cited on p. 15)

[16] Joshua Kazdan, Rylan Schaeffer, Youssef Allouah, Colin Sullivan, Kyssen Yu, Noam Levi, and Sanmi Koyejo. Efficient prediction of pass@k scaling in large language models. *arXiv preprint arXiv:2510.05197*, 2025. (Cited on p. 15)

[17] Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy S Liang. Spoc: Search-based pseudocode to code. *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on p. 15)

[18] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022. (Cited on p. 1)

[19] 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. (Cited on p. 15)

[20] Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Renjie Liao, and Christos Thrampoulidis. Beyond accuracy: A policy gradient reweighting approach for pass@k maximization in LLMs. In *2nd AI for Math Workshop @ ICML 2025*, 2025. (Cited on pp. 4 and 15)

[21] Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025. (Cited on p. 15)

[22] Matteo Papini, Matteo Pirotta, and Marcello Restelli. Smoothing policies and safe policy gradients. *Machine Learning*, 111(11):4081–4137, 2022. (Cited on p. 7)

[23] Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Simko: Simple pass@k policy optimization. *arXiv preprint arXiv:2510.14807*, 2025. (Cited on p. 15)

[24] Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. How to explore to scale rl training of llms on hard problems? https://blog.ml.cmu.edu/2025/11/26/how-to-explore-to-scale-rl-training-of-llms-on-hard-problems, 2025. CMU MLD Blog. (Cited on p. 11)

[25] Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration. *arXiv preprint arXiv:2601.18779*, 2026. (Cited on pp. 11, 15, and 16)

[26] Tom Schaul, Diana Borsa, Joseph Modayil, and Razvan Pascanu. Ray interference: a source of plateaus in deep reinforcement learning. *arXiv preprint arXiv:1904.11455*, 2019. (Cited on pp. 3, 5, 11, and 15)

[27] Pier Giuseppe Sessa, Robert Dadashi-Tazehozi, Leonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Rame, Bobak Shahriari, Sarah Perrin, Abram L. Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos Garea, Amélie Héliou, Aliaksei Severyn, Matthew Hoffman, Nikola Momchev, and Olivier Bachem. BOND: Aligning LLMs with best-of-n distillation. In *International Conference on Learning Representations*, 2025. (Cited on p. 15)

[28] 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. (Cited on p. 15)

[29] Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Remi Munos. Optimizing language models for inference time objectives using reinforcement learning. In *International Conference on Machine Learning*, 2025. (Cited on pp. 1, 2, 4, 15, and 16)

[30] Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage shaping as surrogate reward maximization: Unifying pass@k policy gradients. *arXiv preprint arXiv:2510.23049*, 2025. (Cited on p. 15)

[31] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. (Cited on p. 15)

[32] Christian Walder and Deep Tejas Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. In *Annual Conference on Neural Information Processing Systems*, 2025. (Cited on pp. 1, 2, 4, 8, 15, and 16)

[33] Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025. (Cited on p. 15)

[34] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020. (Cited on pp. 3, 15, and 16)

[35] Yang Yu. Pass@k metric for rlvr: A diagnostic tool of exploration, but not an objective. *arXiv preprint arXiv:2511.16231*, 2025. (Cited on pp. 4 and 15)

[36] Rui Yuan, Robert M Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pages 3332–3380. PMLR, 2022. (Cited on p. 7)

Contents

1 Introduction **1**

2 Preliminaries **3**

3 Prompt Interference **4**

4 Pass@k and Pass@1 Gradients Conflict **5**

4.1 Gradient conflict: Reweighting meets interference 5

4.2 Sufficient conditions and threshold in k 7

4.3 Pass@1 degradation under pass@k updates 8

5 Experiments **8**

6 Conclusion and Future Work **10**

A Related Work **15**

B Limitations **16**

C Toy example **16**

C.1 Setting and illustration of negative prompt interference 16

C.2 Tradeoff: pass@k ascent can decrease pass@1 17

D Discussion of the covariance condition in Proposition 4.1 **18**

E A kernel view of gradient conflict beyond $k = 1$ **18**

F Proofs **19**

F.1 Proof of Proposition 4.1: Gradients conflict characterization 19

F.2 Proof of Corollary 4.4: Dominating negatively interfering prompts 19

F.3 Proof of Proposition 4.5: Influence of k 20

F.4 Lemma F.1: Smoothness of pass@k 20

F.5 Proof of Proposition 4.6: Pass@1 degradation under pass@k updates 22

G Additional Experiments **23**

G.1 More details regarding experimental setting 23

G.2 Threshold robustness analysis 23

G.3 Additional ablations: Which prompts are amplified by pass@k? 23

G.4 Additional experiments on code verification 25

A Related Work

Inference-time fine-tuning. Inference-time objective optimization has recently witnessed a surge of interest for optimizing language model performance for specific inference-time procedures such as win rate [4], Best-of-N [27, 1, 8, 5, 2], majority voting [31, 19, 12] or more recently their combination [11]. Davis and Recht [9] showed that several existing algorithms for post-training large language models (such as rejection sampling fine-tuning (see, e.g., [33]) and GRPO [28]) can be seen as instances of stochastic gradient ascent for maximizing a monotonically increasing function of the probability of giving a correct answer given a prompt, in the binary reward setting.

Pass@k optimization. In this work, we focus on the special case of the pass@k metric which has been introduced as an evaluation metric in [6] for code generation tasks, similarly to a metric proposed earlier in [17]. Given access to verifiable rewards, Chen et al. [6] proposed an unbiased estimator of the pass@k metric. Inspired by inference-aware optimization gains, Tang et al. [29] recently proposed to directly perform pass@k policy optimization with variance-reduced pass@k policy gradients using the control-variates method. Concurrently, Walder and Karkhanis [32] introduced new lower variance unbiased estimators for pass@k and its gradient using more general U-statistics and demonstrated its performance for solving hard problems.

Mahdavi et al. [20] derived expressions for the pass@k objective and pass@k policy gradients as presented in (2) and proposed a new pass@k gradient reweighting method with a tunable parameter γ different from k. Yu [35] also derives per-prompt pass@k policy gradients and shows that per-prompt pass@k and pass@1 gradients are positively collinear vectors, questioning the soundness of considering the pass@k objective and advocating for different exploration mechanisms while keeping pass@k as an inference-time evaluation metric. We show in our work that the *expected* pass@k and pass@1 gradients over the prompt distribution may not necessarily be positively collinear.

Chen et al. [7] investigated how pass@k model training balances the exploration and exploitation capabilities of large language models. Jiang et al. [15] further considered a different risk-sensitive objective than pass@k based on a soft maximum (log-sum-exp) objective depending on a temperature parameter, providing a knob to trade-off between pass@k and pass@1 performance. Peng et al. [23] proposed Simple Pass@K Optimization (SimKO) which redistributes gradient updates across the top-K token-level probability candidates to promote exploration and improve the balance between exploitation and exploration.

Thrampoulidis et al. [30] connect policy gradient algorithms for pass@k optimization [20, 29, 32] to advantage shaped GRPO methods [7, 28, 10, 21].

Kazdan et al. [16] show how pass@k can be estimated even in cases where the number of samples per problem is smaller than the number of attempts k and use dynamic sampling to allocate sampling budget according to prompt difficulty.

To the best of our knowledge, prior work on pass@k optimization does not theoretically characterize when and why optimizing pass@k can degrade pass@1 under pass@k policy optimization; we provide such a characterization.

Interference. The concept of *ray interference* has been previously introduced and investigated in multi-task reinforcement learning [26]. As an analogy with multi-task learning, tasks can be viewed as prompts (or problems to solve in the special case of mathematical reasoning) in our LLM post-training setting. Instead of composite optimization objectives in the form of a sum over tasks, pass@k objectives are expected probabilities of success over a prompt distribution, which may be discrete or continuous (see (pass@k)). Our definition of prompt interference (Definition 3.1) parallels [26, Definition 1], using gradients of pass@1 in our LLM post-training setting rather than gradients of task loss functions in multi-task learning. Related notions of gradient conflict/interference have also been studied in multi-task learning (e.g., [34]). To extend the multi-task learning analogy discussed in [25], tasks can also be thought of as clusters of prompts labeled by degree of difficulty.

Inspired by [26], Qu et al. [25] recently investigated learning to reason on hard problems when no correct rollouts are sampled and argue that pass@k optimization does not address this challenge. They emphasize the role of ray interference in hindering learning on heterogeneous prompt mixtures; in particular, Section 3.2 discusses negative interference between easy and hard problems, and how pass@k policy optimization does not mitigate its consequences. Conceptually, Qu et al. [25] study interference as an exploration/solvability bottleneck on heterogeneous mixtures, while we analyze

the pass@k objective itself and show how prompt-level gradient conflict can induce an explicit pass@k–pass@1 trade-off.

Our findings are consistent with the empirical insights and intuitions in [25, Section 3.2]. We offer a complementary theoretical view by formalizing prompt interference for LLM post-training and providing conditions under which pass@1 degradation can occur under pass@k policy optimization via prompt-level gradient conflict (see Proposition 4.1, Corollary 4.4, and Proposition 4.6). Our results also connect to several open questions highlighted by [25].

While our paper focuses on the theoretical understanding of the pass@k vs pass@1 trade-off and limitations of pass@k policy optimization, Qu et al. [25] propose a framework (Privileged On-Policy Exploration, POPE) to mitigate ray interference and improve the solvability of hard problems by guiding on-policy exploration. Our theoretical insights may suggest alternative mitigation mechanisms and clarify when such interventions are necessary and effective.

B Limitations

In this section, we discuss limitations of our theoretical and empirical results.

First, the gradient conflict characterization in Proposition 4.1 is local to a fixed policy and does not imply monotonic degradation of pass@1 throughout stochastic, multi-step training. Both the pass@k weights and prompt agreement scores evolve with the policy, so the gradient alignment may change over training and pass@1 may eventually stabilize or recover. The success-probability separation assumption is used only to obtain the explicit threshold-in-k sufficient condition in Proposition 4.5; when success probabilities of negatively interfering and remaining prompts overlap, the exact characterization in Proposition 4.1 remains valid, but the threshold guarantee need not hold. Similarly, the smoothness assumption is required for the finite-step guarantee in Proposition 4.6 rather than for the core population-gradient identity.

Second, our original MATH experiments use hard/easy filtered subsets and a reduced-space gradient diagnostic as controlled stress tests. The additional end-to-end experiment at the end of Section 5 removes these restrictions by evaluating all 500 held-out MATH-500 prompts and computing gradient diagnostics with respect to the trainable LoRA parameters. Nevertheless, the estimates use 32 sampled responses per prompt and may therefore have substantial per-prompt Monte Carlo variance, especially for low-success prompts.

Third, although pass@1 degradation under pass@k optimization has been observed previously [29, 32], it would be interesting to conduct more simulations beyond our illustrative example at the end of Section 5, which is limited to one 1.5B model, one mathematical-reasoning benchmark and $k = 8$. Broader empirical evaluation would allow to further explore how frequently and strongly the phenomenon occurs across model families, scales, tasks and values of k.

Finally, our results suggest several possible mitigation strategies, including clipping pass@k weights, agreement-aware downweighting of negatively interfering prompts, and first-order pass@1 gradient guards inspired by the multi-task gradient-conflict literature (e.g., [34]). Testing and comparing these different strategies in full LLM post-training is an interesting avenue for future work.

C Toy example

C.1 Setting and illustration of negative prompt interference

We consider a minimal contextual-bandit abstraction of verifier-based multi-sample LLM evaluation as a running example. A prompt is a context $x \in \mathcal{X}$ drawn from a mixture distribution with a latent difficulty label $\ell(x) \in \{\text{easy}, \text{hard}\}$ with a fixed proportion of hard labels: $\mathbb{P}(\ell(x) = \text{hard}) = p_{\text{hard}}$. Each prompt is represented by a 2D feature vector $\psi(x) = [1, s(x)]^\top \in \mathbb{R}^2$ with a bias 1 and a scalar feature $s(x) \in \mathbb{R}$. Conditioned on difficulty, the scalar feature follows overlapping Gaussians whose separation is controlled by a parameter $\delta > 0$: if $\ell(x) = \text{easy}$ then $s(x) \sim \mathcal{N}(-\delta/2,\, 1)$, and if $\ell(x) = \text{hard}$ then $s(x) \sim \mathcal{N}(+\delta/2,\, 1)$. Smaller δ yields greater overlap between easy and hard feature distributions. A completion is represented as a discrete action $y \in \{0, 1\}$. Each prompt has a unique correct completion determined by its difficulty: $y^\star(x) = 0$ if $\ell(x) = \text{easy}$ and $y^\star(x) = 1$ if $\ell(x) = \text{hard}$. The verifier reward is deterministic binary: $r(x, y) = \mathbf{1}\{y = y^\star(x)\}$. We use a

two-action softmax policy parameterized by a single vector $\theta \in \mathbb{R}^2$ such that $\pi_\theta(1|x) = \sigma(\theta^\top \psi(x))$ and $\pi_\theta(0|x) = 1 - \sigma(\theta^\top \psi(x))$, where $\sigma(t) = \frac{1}{1+\exp(-t)}$ is the sigmoid function. The policy shares parameters across heterogeneous subpopulations (easy vs hard). For later use, we also note that we initialize the policy at a reference policy θ_{ref} that is globally biased toward action $y = 0$, inducing high initial success on easy prompts and low initial success on hard prompts.

Illustration of negative prompt interference. It follows that the per-prompt success probability defined in (1) is given by $p_\theta(x) = \pi_\theta(y^\star(x)|x)$ for any prompt $x \in \mathcal{X}$. A simple computation yields:

$$\nabla p_\theta(x) = \begin{cases} -z_\theta(x)\,\psi(x), & \text{if } \ell(x) = \text{easy}, \\ +\,z_\theta(x)\,\psi(x), & \text{if } \ell(x) = \text{hard}\,, \end{cases} \tag{13}$$

where $z_\theta(x) := \sigma(\theta^\top \psi(x))(1 - \sigma(\theta^\top \psi(x))) \geq 0$. As a consequence, two prompts that have a similar representation will have opposite per-prompt pass@1 gradients and will hence be negatively interfering. This will be the case in our example for prompts in the overlap between the two Gaussians: two such prompts can have different labels (hard and easy) while having a similar prompt representation (with a similar scalar feature $s(x)$). See Figure 5 for an illustration.

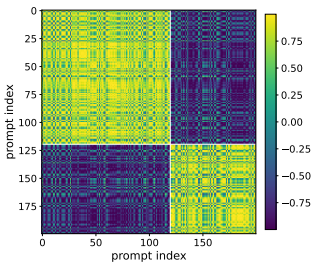


Figure 5: Cosine kernel heatmap: $\cos(\nabla p_\theta(x), \nabla p_\theta(x'))$ for subsamples of prompts: 120 easy and 80 hard among a total of 6000 samples. Blue regions show negative prompt interference.

In the next section, we show in our toy example how negative prompt interference can lead to pass@1 decrease under one-step pass@k policy gradient ascent.

C.2 Tradeoff: pass@k ascent can decrease pass@1

Using the toy example introduced in the previous section, we show in this section that pass@1 can decrease under one-step pass@k policy gradient ascent.

Fix $\delta = 0.2$ and consider two realized prompts in the overlap region between the 2 Gaussians: a prompt x_e with $\ell(x_e) = \text{easy}$ and $\psi(x_e) = [1, -0.1]^\top$, and x_h with $\ell(x_h) = \text{hard}$ and $\psi(x_h) = [1, +0.1]^\top$. These two prompts have nearly identical representations as $\cos(\psi(x_e), \psi(x_h)) \simeq 0.98$ is close to 1. Because of (13), they are therefore negative interfering as $\kappa_\theta(x_e, x_h) = \langle \nabla p_\theta(x_e), \nabla p_\theta(x_h) \rangle = -z_\theta(x_e) z_\theta(x_h) \langle \psi(x_e), \psi(x_h) \rangle < 0$ since $\langle \psi(x_e), \psi(x_h) \rangle > 0$. Numerically we have $\kappa_\theta(x_e, x_h) \simeq -0.01$ and $\cos(\nabla p_\theta(x_e), \nabla p_\theta(x_h)) \simeq -0.98$ which means that the per-prompt pass@1 gradients are almost antiparallel. On this two-point empirical distribution, the pass@1 and pass@k gradients are given by:

$$\nabla J_1(\theta) = \frac{1}{2}(\nabla p_\theta(x_e) + \nabla p_\theta(x_h))\,, \quad \nabla J_{10}(\theta) = \frac{1}{2}(w_e \nabla p_\theta(x_e) + w_h \nabla p_\theta(x_h))\,,$$

where the pass@10 weight $w_e := w_{10,\theta}(x_e) \simeq 2.33 \times 10^{-7}$ whereas $w_h := w_{10,\theta}(x_h) \simeq 3.88$. Pass@10 upweights the hard prompt ($p_\theta(x_h) \simeq 0.10$ low) and aggressively downweights the easy prompt x_e (large $p_\theta(x_h) \simeq 0.86$). Indeed, recall that $w_{k,\theta}(x) = k(1 - p_\theta(x))^{k-1}$ which is near k when $p_\theta(x) \approx 0$ but near 0 when $p_\theta(x) \approx 1$. As a consequence, the pass@10 gradient is therefore dominated by $\nabla p_\theta(x_h)$ which conflicts with $\nabla p_\theta(x_e)$. Importantly, we have $\cos(\nabla J_1(\theta), \nabla J_{10}(\theta)) = -0.77 < 0$ meaning that the pass@1 and pass@10 are conflicting and strongly anti-aligned. With one gradient-ascent step on pass@10 $\theta^+ = \theta + \eta \nabla J_{10}(\theta)$ with $\eta = 5$, we have:

$$J_1(\theta) \simeq 0.48 \;\to\; J_1(\theta^+) \simeq 0.46 \quad (\downarrow), \qquad J_{10}(\theta) \simeq 0.83 \;\to\; J_{10}(\theta^+) = 0.95 \quad (\uparrow).$$

This shows that pass@1 decreases while pass@k increases under pass@k optimization. In our toy example, we have observed that negative prompt interference can induce a conflict between pass@1 and pass@k gradients when it is amplified by the induced pass@k prompt reweighting.

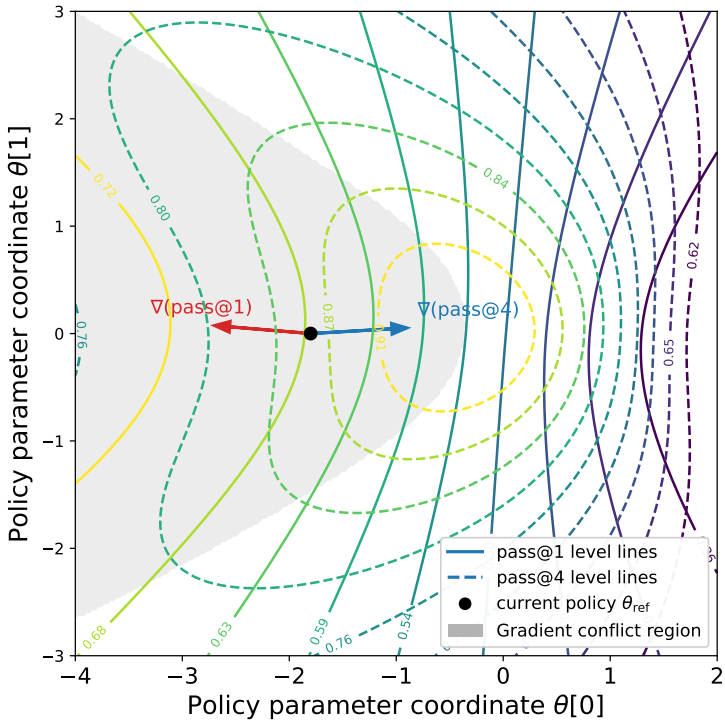


Figure 6: Contour plots of pass@1 and pass@k objectives in the policy parameter space. Gradients of pass@k and pass@1 with respect to policy parameters are conflicting in the gray area.

D Discussion of the covariance condition in Proposition 4.1

Introducing the correlation $\rho_{w,a}(\theta) := \frac{\mathrm{cov}(w_{k,\theta}(x), a_\theta(x))}{\sigma_w(\theta)\cdot\sigma_a(\theta)}$ where $\sigma_w(\theta), \sigma_a(\theta)$ are respectively the standard deviations of $w_{k,\theta}(x)$ and $a_\theta(x)$ and supposing $\sigma_w(\theta)\sigma_a(\theta) > 0$, we can then rewrite condition (8) equivalently as:

$$\rho_{w,a}(\theta) < -\frac{\mathbb{E}[w_{k,\theta}(x)] \cdot \|\nabla J_1(\theta)\|^2}{\sigma_w(\theta) \cdot \sigma_a(\theta)} \,. \tag{14}$$

This condition emphasizes that conflict requires strong negative correlation between hardness weights and agreement scores as well as sufficient variance. If the weights barely vary ($\sigma_w(\theta) \simeq 0$), pass@$k$ cannot strongly shift focus across prompts and conflict is harder. If agreement scores barely vary ($\sigma_a(\theta) \simeq 0$) there is no meaningful conflict structure to exploit that can lead to misaligned gradients.

E A kernel view of gradient conflict beyond $k = 1$

In this section, we further investigate the gradient conflict phenomenon between pass@k and pass@m for any $k, m \geq 1$. Using the kernel κ_θ defined in (5), we now relate pass@k and pass@m gradients for any $k, m \geq 1$ using (2) to obtain:

$$\langle \nabla J_k(\theta), \nabla J_m(\theta) \rangle = \mathbb{E}_{x,x' \sim \mathcal{D}}[w_{k,\theta}(x) w_{m,\theta}(x') \kappa_\theta(x, x')] \,. \tag{15}$$

This identity separates two factors: (1) where each objective focuses via the weights $w_{k,\theta}(x), w_{m,\theta}(x')$ with prompt reweighting toward 'hard' prompts as k grows; and (2) whether those regions interfere via the sign structure of $\kappa_\theta(x, x')$ which is independent of the parameter k. Conflict between pass@k and pass@m gradients happens when the objectives emphasize prompt regions whose cross-similarity (encoded by the kernel) is negative.

The following result which is an immediate consequence of (15) shows that conflict requires interference.

Proposition E.1 (No interference regime: No negative transfer $\implies$ no conflict)**.** *For any policy parameter $\theta \in \mathbb{R}^d$, if $\kappa_\theta(x, x') \geq 0$ for $\mathcal{D} \times \mathcal{D}$ a.e. (x, x') then, $\langle \nabla J_k(\theta), \nabla J_m(\theta) \rangle \geq 0$ for all $k, m \geq 1$.*

If pairwise prompt similarities are nonnegative on the support of the prompt distribution $\mathcal{D}$ then all pass@k objectives are mutually aligned. Therefore, conflict can only occur when pairwise similarities are not always nonnegative and depends on how the weights $w_{k,\theta}, w_{m,\theta}$ shift mass. In other words, pass@k objectives cannot conflict unless there is negative transfer between some prompts.

F Proofs

F.1 Proof of Proposition 4.1: Gradients conflict characterization

Proof. First recall from (2) that for any $k \geq 1$ and any $\theta \in \mathbb{R}^d$,

$$\nabla J_k(\theta) = \mathbb{E}_{x\sim\mathcal{D}}[w_k(p_\theta(x))\nabla p_\theta(x)] = \mathbb{E}_{x\sim\mathcal{D}}[w_k(p_\theta(x))\nabla J_1(x;\theta)]\,. \tag{16}$$

Then it immediately follows that:

$$\begin{aligned}\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle &= \langle \mathbb{E}_{x\sim\mathcal{D}}[w_k(p_\theta(x))\nabla J_1(x;\theta)], \nabla J_1(\theta)\rangle \\ &= \mathbb{E}_{x\sim\mathcal{D}}[\langle w_k(p_\theta(x))\nabla J_1(x;\theta), \nabla J_1(\theta)\rangle] \\ &= \mathbb{E}_{x\sim\mathcal{D}}[w_k(p_\theta(x))a_\theta(x)]\,,\end{aligned} \tag{17}$$

where we recall that $a_\theta(x) := \langle \nabla J_1(x;\theta), \nabla J_1(\theta)\rangle$. By definition of the covariance between two real-valued random variables:

$$\begin{aligned}\mathrm{cov}(w_{k,\theta}(x), a_\theta(x)) &= \mathbb{E}_{x\sim\mathcal{D}}[w_{k,\theta}(x)\cdot a_\theta(x)] - \mathbb{E}_{x\sim\mathcal{D}}[w_{k,\theta}(x)]\cdot\mathbb{E}_{x\sim\mathcal{D}}[a_\theta(x)] &(18)\\ &= \langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle - \mathbb{E}_{x\sim\mathcal{D}}[w_{k,\theta}(x)]\cdot\|\nabla J_1(\theta)\|^2\,, &(19)\end{aligned}$$

where the second equality follows from using (17) and noticing that $\mathbb{E}_{x\sim\mathcal{D}}[a_\theta(x)] = \|\nabla J_1(\theta)\|^2$ by definition of $a_\theta(x)$. Rearranging the above identity then gives:

$$\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle = \mathbb{E}_{x\sim\mathcal{D}}[w_{k,\theta}(x)]\cdot\|\nabla J_1(\theta)\|^2 + \mathrm{cov}(w_{k,\theta}(x), a_\theta(x))\,. \tag{20}$$

As a consequence of (17) and (20), we have that $\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle < 0$ if and only if: $\mathbb{E}_{x\sim\mathcal{D}}[w_k(p_\theta(x))a_\theta(x)] < 0$ which is also equivalent to the condition:

$$\mathrm{cov}(w_{k,\theta}(x), a_\theta(x)) < -\mathbb{E}[w_{k,\theta}(x)]\cdot\|\nabla J_1(\theta)\|^2\,. \tag{21}$$

□

F.2 Proof of Corollary 4.4: Dominating negatively interfering prompts

Proof. First, Proposition 4.1 yields:

$$\left\langle \nabla J_k(\theta), \nabla J_1(\theta)\right\rangle = \mathbb{E}[\omega_{k,\theta}(x)\, a_\theta(x)]\ . \tag{22}$$

We decompose this inner product into 2 regions: the m-strongly negatively interfering region $\mathcal{X}_-(\theta, m)$ and its complement,

$$\begin{aligned}\left\langle \nabla J_k(\theta),\ \nabla J_1(\theta)\right\rangle &= \mathbb{E}[\omega_{k,\theta}(x)\, a_\theta(x)] \\ &= \mathbb{E}\left[\omega_{k,\theta}(x)\, a_\theta(x)\, \mathbb{1}_{\{x\in\mathcal{X}_-(\theta,m)\}}\right] + \mathbb{E}\left[\omega_{k,\theta}(x)\, a_\theta(x)\, \mathbb{1}_{\{x\notin\mathcal{X}_-(\theta,m)\}}\right]\ .\end{aligned} \tag{23}$$

We now upperbound each one of the above terms. For the first term, by definition of the set of prompts $\mathcal{X}_-(\theta, m)$, we have $a_\theta(x) \leq -m$ for any $x \in \mathcal{X}_-(\theta, m)$. As for the second term, we show that for any $x \notin \mathcal{X}_-(\theta, m)$, $|a_\theta(x)| \leq G^2$ using Assumption 4.3 as follows.

Bounding $a_\theta(x)$. Using the definition of the agreement score $a_\theta(x)$ (see (7)) together with Cauchy-Schwarz inequality, we have

$$|a_\theta(x)| = |\langle \nabla J_1(x;\theta), \nabla J_1(\theta)\rangle| \leq \|\nabla J_1(x;\theta)\|\cdot\|\mathbb{E}[\nabla J_1(x';\theta)]\| \leq \|\nabla J_1(x;\theta)\|\cdot\mathbb{E}[\|\nabla J_1(x';\theta)\|]\,. \tag{24}$$

Then recalling the policy gradient expression of $\nabla J_1(x;\theta)$ in (4) and using Jensen's inequality, we have:

$$\|\nabla J_1(x;\theta)\|^2 = \|\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x,y)s_\theta(x,y)]\|^2 \le \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[\|s_\theta(x,y)\|^2] \le G^2\,, \tag{25}$$

where the last inequality follows from using Assumption 4.3. Using this inequality in (24) yields the desired inequality:

$$\forall x \in \mathcal{X}, \quad |a_\theta(x)| \le G^2\,. \tag{26}$$

Back to (23) and using the inequalities established above, we conclude the proof:

$$\big\langle \nabla J_k(\theta),\ \nabla J_1(\theta)\big\rangle \le -\delta(\theta)\,, \tag{27}$$

where $\delta(\theta) := mW_-(k,\theta) - G^2 W_+(k,\theta)$ and the aggregated weights $W_-(k,\theta), W_+(k,\theta)$ are defined in (11). If in addition $\delta(\theta) > 0$, then $\langle \nabla J_k(\theta), \nabla J_1(\theta)\rangle \le -\delta(\theta) < 0$. □

F.3 Proof of Proposition 4.5: Influence of k

Proof. Recall from Proposition 4.1 that:

$$\big\langle \nabla J_k(\theta), \nabla J_1(\theta)\big\rangle = \mathbb{E}[\omega_{k,\theta}(x)\, a_\theta(x)]\ . \tag{28}$$

Using again the same decomposition as in (23), we have:

$$\begin{aligned}\big\langle \nabla J_k(\theta),\ \nabla J_1(\theta)\big\rangle &= \mathbb{E}[\omega_{k,\theta}(x)\, a_\theta(x)] \\ &= \mathbb{E}\big[\omega_{k,\theta}(x)\, a_\theta(x)\, \mathbb{1}_{\{x\in\mathcal{X}_-(\theta,m)\}}\big] + \mathbb{E}\big[\omega_{k,\theta}(x)\, a_\theta(x)\, \mathbb{1}_{\{x\notin\mathcal{X}_-(\theta,m)\}}\big]\ .\end{aligned} \tag{29}$$

Depending on the prompt region, we have two distinct ways to bound the pass@k weights:

(i) For $x \in \mathcal{X}(m,\theta)$, we have by assumption, $p_\theta(x) \le \varepsilon$, hence $\omega_{k,\theta}(x) \ \ge\ k(1-\varepsilon)^{k-1}$.

(ii) For $x \notin \mathcal{X}(m,\theta)$ we have by assumption, $p_\theta(x) \ge \delta$, hence $\omega_{k,\theta}(x) \ \le\ k(1-\delta)^{k-1}$.

Observing in addition to both items above that $a_\theta(x) \le -m$ for any $x \in \mathcal{X}_-(\theta,m)$ and $a_\theta(x) \le G^2$ for any $x \in \mathcal{X}$ (as shown in (26) under Assumption 4.3) and hence for any $x \notin \mathcal{X}_-(\theta,m)$ in particular, we obtain from (29) the following inequality:

$$\big\langle \nabla J_k(\theta), \nabla J_1(\theta)\big\rangle \ \le\ -k(1-\varepsilon)^{k-1}\, m\, q \ +\ k(1-\delta)^{k-1}\, G^2\,(1-q)\,, \tag{30}$$

where we also use the assumption $q_\theta \ge q$. Thus it suffices to have

$$k(1-\delta)^{k-1}G^2(1-q) \ <\ k(1-\varepsilon)^{k-1}mq,$$

to ensure that $\big\langle \nabla J_k(\theta), \nabla J_1(\theta)\big\rangle < 0$. The above condition is equivalent to:

$$\frac{G^2(1-q)}{mq} \ <\ \left(\frac{1-\varepsilon}{1-\delta}\right)^{k-1}.$$

Taking $\log$ in the above inequality, recalling that $1-\varepsilon > 1-\delta$ and reordering the inequality yields:

$$k \ > k^\star := \ 1 \ +\ \frac{\log\Big(\dfrac{(1-q)\,G^2}{q\,m}\Big)}{\log\Big(\dfrac{1-\varepsilon}{1-\delta}\Big)}\,, \tag{31}$$

which concludes the proof. □

F.4 Lemma F.1: Smoothness of pass@k

Lemma F.1 (Smoothness of pass@k)**.** *Under Assumption 4.3, for all $k \ge 1$, the pass@k objective J_k is L_k-smooth with $L_k := k^2G^2 + kF$, i.e. for all $\theta, \theta' \in \mathbb{R}^d$,*

$$|J_k(\theta') - J_k(\theta) - \langle \nabla J_k(\theta), \theta' - \theta\rangle| \le \frac{L_k}{2}\|\theta' - \theta\|_2^2\,.$$

Proof. To show smoothness of the pass@k objective J_k, we bound its Hessian $\nabla^2 J_k(\theta)$ uniformly in the policy parameter θ using Assumption 4.3.

Since $\nabla J_k(\theta) = \mathbb{E}_{x\sim\mathcal{D}}[w_k(p_\theta(x))\nabla p_\theta(x)]$ as shown in (2), the chain rule gives the Hessian of J_k for every $\theta \in \mathbb{R}^d$,

$$\nabla^2 J_k(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\Big[\omega_k'(p_\theta(x))\,\nabla p_\theta(x)\,\nabla p_\theta(x)^\top + \omega_k(p_\theta(x))\,\nabla^2 p_\theta(x)\Big]. \tag{32}$$

Taking the operator norm and using the triangular inequality yields:

$$\big\|\nabla^2 J_k(\theta)\big\| \le \mathbb{E}_{x\sim\mathcal{D}}\Big[\big|\omega_k'(p_\theta(x))\big|\cdot\,\|\nabla p_\theta(x)\|^2 + \big|\omega_k(p_\theta(x))\big|\cdot\,\|\nabla^2 p_\theta(x)\|\Big]. \tag{33}$$

In the rest of the proof, we bound each term separately.

Step 1: Bounding $\|\nabla p_\theta(x)\|^2$. Recall from (4) that the pass@1 policy gradient is given by:

$$\nabla p_\theta(x) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[r(x,y)\,\nabla\log\pi_\theta(y|x)\Big]. \tag{34}$$

By Jensen's inequality and since $r(x,y)\in\{0,1\}$, we have

$$\|\nabla p_\theta(x)\|^2 \le \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[r(x,y)^2\,\|\nabla\log\pi_\theta(y|x)\|^2\Big] \le \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[\|\nabla\log\pi_\theta(y|x)\|^2\Big] \le G^2\,, \tag{35}$$

where the last bound follows from Assumption 4.3.

Step 2: Bounding $\|\nabla^2 p_\theta(x)\|$. Differentiating the pass@1 policy gradient, we obtain:

$$\nabla^2 p_\theta(x) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[r(x,y)\,\nabla\log\pi_\theta(y|x)\,\nabla\log\pi_\theta(y|x)^\top + r(x,y)\,\nabla^2\log\pi_\theta(y|x)\Big].$$

Taking the operator norm and using the triangle inequality, we obtain:

$$\begin{aligned}\|\nabla^2 p_\theta(x)\| &\le \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[|r(x,y)|\cdot\,\|\nabla\log\pi_\theta(y|x)\|^2\Big] + \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[|r(x,y)|\cdot\,\|\nabla^2\log\pi_\theta(y|x)\|\Big]\\ &\le \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[\|\nabla\log\pi_\theta(y|x)\|^2\Big] + \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[\|\nabla^2\log\pi_\theta(y|x)\|\Big] \le G^2 + F\,,\end{aligned} \tag{36}$$

where the last inequality again follows from using Assumption 4.3.

Step 3: Bounding weights ω_k' and ω_k. By definition of the weights $w_k(p) = f_k'(p) = k(1-p)^{k-1}$ for any $p\in[0,1]$. Therefore it follows that $\omega_k'(p) = -k(k-1)(1-p)^{k-2}$ for $k\ge 2$. Hence, for any $p\in[0,1]$ and any $k\ge 2$,

$$|\omega_k(p)| \le k\,, \quad |\omega_k'(p)| \le k(k-1)\,. \tag{37}$$

For $k=1$, $w_1(p)=1$ and $w_1'(p)=0$, so the same conclusion follows directly.

Conclusion. Plugging (35)–(36) and the bounds on ω_k, ω_k' (37) into (33) yields

$$\|\nabla^2 J_k(\theta)\| \le k(k-1)\,G^2 + k\,(G^2+F) \le k^2G^2 + kF.$$

Finally, boundedness of the Hessian norm implies that J_k is L_k-smooth with $L_k = k^2G^2 + kF$, i.e., for any $\theta,\theta'\in\mathbb{R}^d$,

$$\Big|J_k(\theta') - J_k(\theta) - \langle\nabla J_k(\theta), \theta'-\theta\rangle\Big| \le \frac{L_k}{2}\|\theta'-\theta\|^2.$$

□

F.5 Proof of Proposition 4.6: Pass@1 degradation under pass@k updates

Proof. We prove the decrease of pass@1 and increase of pass@k under pass@k policy gradient ascent separately. Recall the policy update rule with step size $\eta > 0$:

$$\theta^+ = \theta + \eta \nabla J_k(\theta). \tag{38}$$

(i) Pass@1 descent. By L_1-smoothness of J_1 (Lemma F.1) and using the policy update rule (38), we have:

$$\begin{aligned} J_1(\theta^+) &\leq J_1(\theta) + \langle \nabla J_1(\theta), \theta^+ - \theta \rangle + \frac{L_1}{2}\|\theta^+ - \theta\|^2 \\ &= J_1(\theta) + \eta \langle \nabla J_1(\theta), \nabla J_k(\theta) \rangle + \frac{L_1 \eta^2}{2}\|\nabla J_k(\theta)\|^2. \end{aligned} \tag{39}$$

We now bound each one of the last two terms in the above inequality.

Bounding $\|\nabla J_k(\theta)\|^2$. Recall the pass@k policy gradient expression $\nabla J_k(\theta) = \mathbb{E}_x[\omega_\theta(x)\, \nabla J_1(x;\theta)]$, we have by Jensen and Cauchy-Schwarz inequalities:

$$\|\nabla J_k(\theta)\|^2 = \|\mathbb{E}_x[\omega_\theta(x)\, \nabla J_1(x;\theta)]\|^2 \;\leq\; \mathbb{E}_x\left[|\omega_\theta(x)|^2\, \|\nabla J_1(x;\theta)\|^2\right] \leq k^2 G^2\,, \tag{40}$$

where the last inequality follows from using the pass@1 policy gradient bound in (25) together with immediate weight bound $|\omega_\theta(x)| \leq k$ and the fact that the reward is binary (this can be generalized to a continuous uniformly bounded reward function).

Bounding $\langle \nabla J_1(\theta), \nabla J_k(\theta) \rangle$. By Corollary 4.4, we have

$$\langle \nabla J_1(\theta), \nabla J_k(\theta) \rangle \leq -\delta(\theta)\,. \tag{41}$$

Combining the bounds (40) and (41), we immediately obtain from (39):

$$J_1(\theta^+) \leq J_1(\theta) - \eta\delta(\theta) + C_2\eta^2\,, \tag{42}$$

where $C_2 := \frac{L_1 k^2 G^2}{2}$ and $L_1 = G^2 + F$ as shown in Lemma F.1, G, F being smoothness constants defined in Assumption 4.3.

Hence, if $\delta(\theta) > 0$ and $0 < \eta < \eta_+ := \frac{\delta(\theta)}{C_2}$, then $-\eta\delta(\theta) + C_2\eta^2 < 0$ and it follows that

$$J_1(\theta^+) \leq J_1(\theta) - \eta\delta(\theta) + C_2\eta^2 < J_1(\theta)\,, \tag{43}$$

which is the first desired inequality.

(ii) Pass@k ascent. As for pass@k under pass@k policy gradient ascent, the proof is a standard application of smoothness in optimization. We provide a brief proof in our setting for completeness.

Using L_k-smoothness of the pass@k objective shown in Lemma F.1 together with the policy update rule (38), we have:

$$\begin{aligned} J_k(\theta^+) &\geq J_k(\theta) + \langle \nabla J_k(\theta), \theta^+ - \theta \rangle - \frac{L_k}{2}\|\theta^+ - \theta\|^2 \\ &= J_k(\theta) + \eta\|\nabla J_k(\theta)\|^2 - \frac{L_k\eta^2}{2}\|\nabla J_k(\theta)\|^2 \\ &= J_k(\theta) + \eta\Big(1 - \frac{L_k\eta}{2}\Big)\|\nabla J_k(\theta)\|^2. \end{aligned} \tag{44}$$

We pick then $\eta \leq \frac{1}{L_k}$ to obtain the second desired inequality:

$$J_k(\theta^+) \geq J_k(\theta) + \frac{\eta}{2}\|\nabla J_k(\theta)\|^2 \geq J_k(\theta)\,. \tag{45}$$

Overall, a sufficient step-size condition to guarantee simultaneous increase of pass@k and decrease of pass@1 (i.e., in particular $J_1(\theta^+) < J_1(\theta)$ and $J_k(\theta^+) > J_k(\theta)$ when $\nabla J_k(\theta) \neq 0$) is:

$$0 < \eta < \min\left(\frac{\delta(\theta)}{C_2}, \frac{1}{L_k}\right), \tag{46}$$

and this concludes the proof. □

G Additional Experiments

G.1 More details regarding experimental setting

Sampling configuration. For each problem, we generate $k = 32$ independent responses using temperature sampling with temperature $T = 0.7$ and nucleus sampling with $p = 0.95$. Responses are evaluated using exact match against ground truth answers and binary rewards indicate correctness.

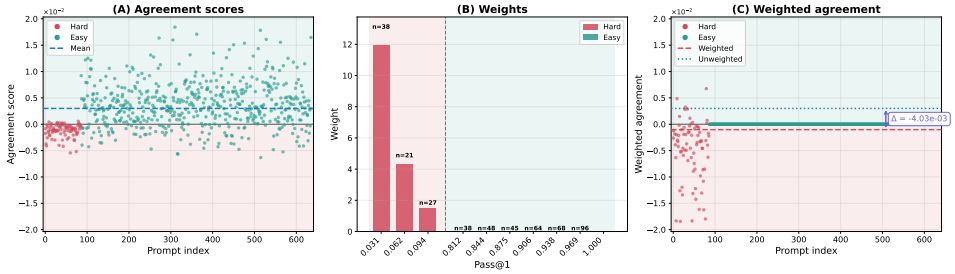
Experiments compute resources. Experiments on the toy example in Appendix C were performed on a MacBook Pro M4 and only take a few seconds to minutes. Experiments on LLMs were performed on GPUs and take several hours to run.

G.2 Threshold robustness analysis

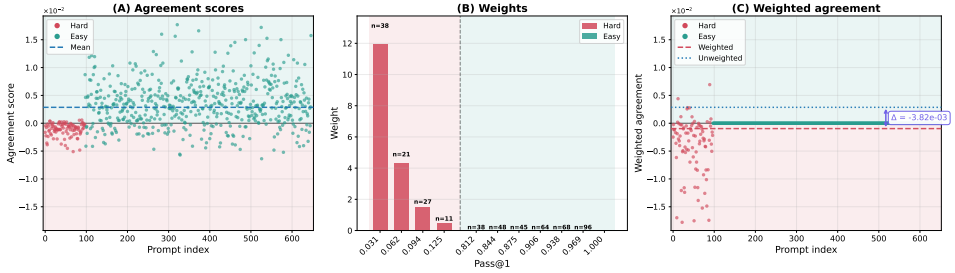
Figure 7 shows gradient misalignment across all additional threshold configurations for DeepSeek-R1-Distill-Llama-8B (beyond the main text example of $\delta_1 = 0.85$, $\delta_2 = 0.10$). All combinations exhibit negative inner products (range: -0.49 to -0.65), demonstrating that the misalignment phenomenon is robust to threshold choices.

G.3 Additional ablations: Which prompts are amplified by pass@k?

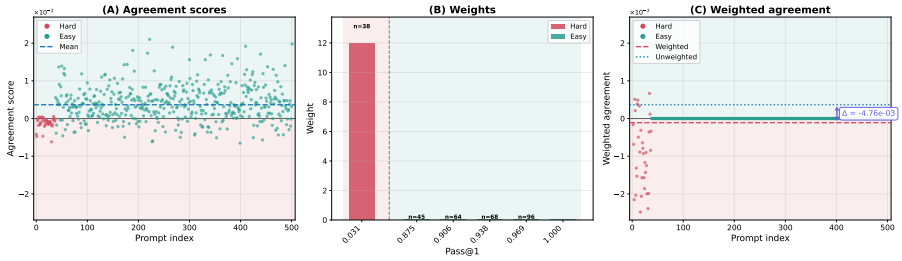
To have a complementary view of the amplification mechanism, we directly plot weights against agreement scores, with color encoding pass@1 probability. See Figure 8 which shows results for both models across representative threshold configurations. The critical observation is where these weights concentrate in agreement-score space (x-axis). High-weight prompts cluster in the negative-agreement region ($a_\theta(x) < 0$), while low-weight prompts cluster in the positive-agreement region. This means pass@k's design, which correctly prioritizes hard prompts for its own objective, systematically amplifies negatively interfering prompts that hurt pass@1.



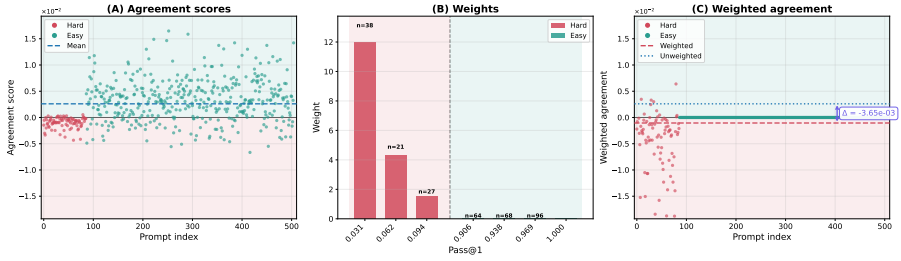
(a) $\delta_1 = 0.80$, $\delta_2 = 0.10$ (636 prompts: 86 hard, 550 easy, IP=-0.652)



(b) $\delta_1 = 0.80$, $\delta_2 = 0.15$ (647 prompts: 97 hard, 550 easy, IP=-0.621)

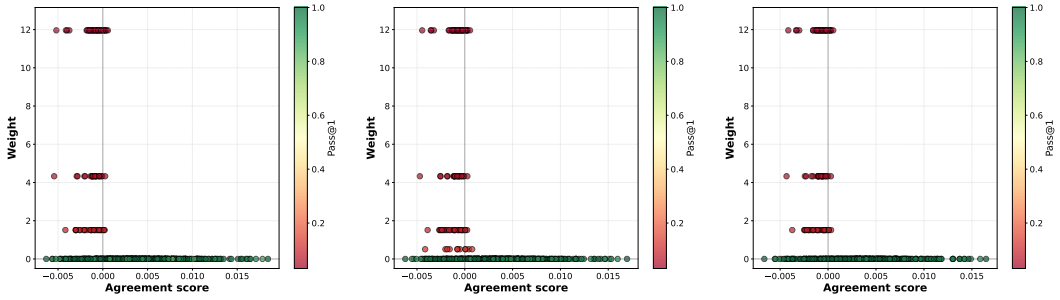


(c) $\delta_1 = 0.85$, $\delta_2 = 0.05$ (502 prompts: 38 hard, 464 easy, IP=-0.551)

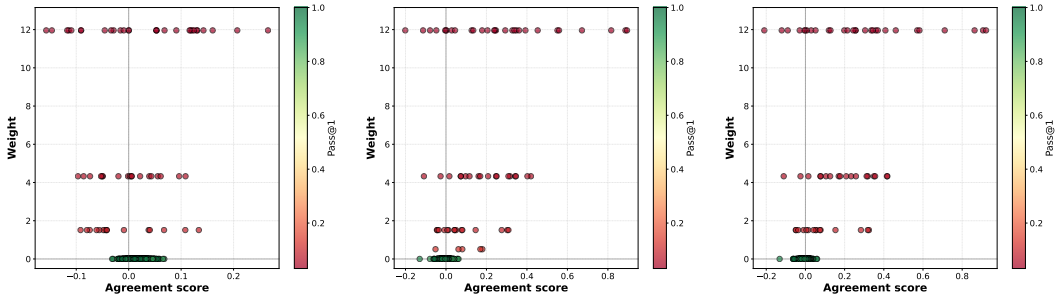


(d) $\delta_1 = 0.90$, $\delta_2 = 0.10$ (505 prompts: 86 hard, 419 easy, IP=-0.527)

Figure 7: **Gradient conflict across all threshold configurations (DeepSeek-R1-Distill-Llama-8B).** Each panel shows agreement scores, weights, and weighted contributions for a different (δ_1, δ_2) configuration. All exhibit negative inner products, confirming robust misalignment across thresholds. The consistent downward shift from unweighted to weighted means shows pass@k optimization systematically opposes pass@1 improvement.



(a) Llama: $\delta_1 = 0.80$, $\delta_2 = 0.10$ (b) Llama: $\delta_1 = 0.85$, $\delta_2 = 0.15$ (c) Llama: $\delta_1 = 0.90$, $\delta_2 = 0.10$



(d) Qwen: $\delta_1 = 0.80$, $\delta_2 = 0.10$ (e) Qwen: $\delta_1 = 0.85$, $\delta_2 = 0.15$ (f) Qwen: $\delta_1 = 0.90$, $\delta_2 = 0.10$

Figure 8: **Pass@k amplification pattern across models and thresholds.** Each point represents one prompt, with the corresponding agreement score (x-axis), pass@k weight (y-axis), and pass@1 probability shown by color (red = low, green = high). **Consistent pattern:** Prompts with negative alignment (left), low pass@1 (red), and high weights (4-12) cluster upper-left, while prompts with positive alignment (right), high pass@1 (green), and zero weights ($\sim 10^{-28}$) appear lower-right. **Top row (a-c):** DeepSeek-R1-Distill-Llama-8B across three δ_2 values (0.05, 0.10, 0.15). **Bottom row (d-f):** DeepSeek-R1-Distill-Qwen-7B with same thresholds.

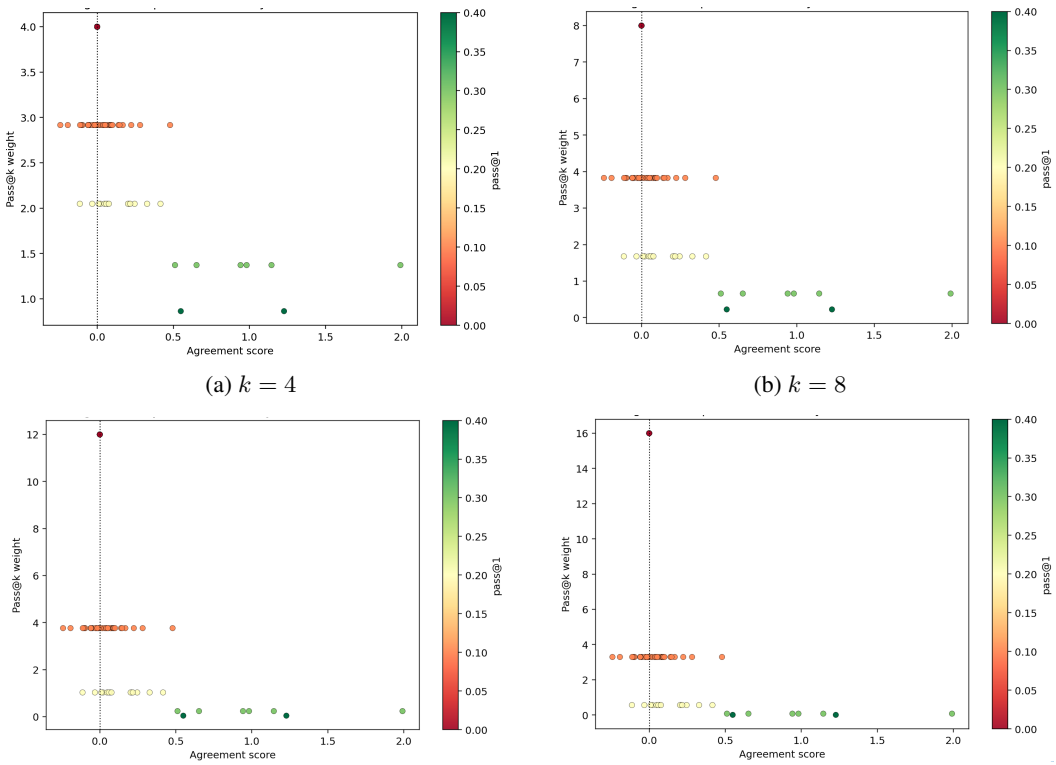
G.4 Additional experiments on code verification

Table 2: Negatively interfering prompt pairs on code generation benchmarks (Definition 3.1).

Model	Dataset	# Prompts	Neg. Pairs	Fraction
Qwen2.5-Coder-1.5B-Instruct	HumanEval	164	**744**	**5.6%**
Meta-Llama-3.1-8B	MBPP	60	**412**	**23.3%**

Table 3: Top-5 negatively interfering pairs — Qwen2.5-Coder-1.5B-Instruct on HumanEval.

Prompt i	Prompt j	$\langle g_i, g_j \rangle$	$\cos(g_i, g_j)$	$\hat{p}_1(i)$	$\hat{p}_1(j)$
HumanEval/92	HumanEval/44	-1594.8	-0.198	1.000	1.000
HumanEval/72	HumanEval/104	-1450.5	-0.255	0.938	0.688
HumanEval/72	HumanEval/133	-1290.0	-0.167	0.938	0.875
HumanEval/92	HumanEval/73	-1267.6	-0.136	1.000	1.000
HumanEval/72	HumanEval/143	-1246.8	-0.112	0.938	0.938



(a) $k = 4$ (b) $k = 8$

(c) $k = 12$ (d) $k = 16$

Figure 9: **Pass@k amplification and negative prompt interference in HumanEval dataset using a Qwen 1.5B model for different values of k.** Each point represents one prompt, with the corresponding agreement score (x-axis), pass@k weight (y-axis), and pass@1 probability shown by color (red = low, green = high). **Consistent pattern:** Prompts with negative alignment (left), low pass@1 (red), and high weights (4-12) cluster upper-left, while prompts with positive alignment (right), high pass@1 (green), and small weights appear lower-right.

Table 4: Top-5 negatively interfering pairs — Meta-Llama-3.1-8B on MBPP.

Prompt i	Prompt j	$\langle g_i, g_j \rangle$	$\cos(g_i, g_j)$	$\hat{p}_1(i)$	$\hat{p}_1(j)$
mbpp_792	mbpp_800	-3.71×10^{-5}	-0.113	0.906	0.875
mbpp_517	mbpp_800	-2.32×10^{-5}	-0.072	0.844	0.875
mbpp_366	mbpp_800	-1.46×10^{-5}	-0.055	0.875	0.875
mbpp_792	mbpp_833	-1.25×10^{-5}	-0.045	0.906	0.906
mbpp_329	mbpp_915	-9.84×10^{-6}	-0.286	0.938	0.031

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our contributions are stated in the same way in both the abstract and in the introduction (in more details).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Section B for a discussion of limitations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: All our results are stated with the precise assumptions we need to prove them.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer:[Yes]

Justification: Our main contributions are to produce a theoretical understanding of the pass@k-pass@1 tradeoff. Experiments mainly serve an illustrative purpose to show that the phenomenon we explain does show up in practice. See the 'Experiments' section in the main part of the paper and Appendix G for additional details regarding all the hyperparameters we choose and the precise experimental setting.

Guidelines:

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

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

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

   Answer: [No]

   Justification: Our main contributions are theoretical. Experiments are not the central part of our contributions. We can release the code later if required.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See experiments section and Appendix G for more details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We use five independent runs and 95% confidence intervals in Table 1. See also appendix for other prompt statistics and different hyperparameter configurations.

   Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Appendix G.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have checked the Code of Ethics and our research does not involve any such violation.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite relevant works regarding the datasets we used for experiments on LLMs for math reasoning and code verification. See Section 5 and Appendix G.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: See Acknowledgements section.

    Guidelines:

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