Learning Robust Reasoning through Guided Adversarial Self-Play

Shuozhe Li[1] **Vaishnav Tadiparthi**[2] **Kwonjoon Lee**[2]
Nakul Agarwal[2] **Hossein Nourkhiz Mahjoub**[2] **Ehsan Moradi Pari**[2]
Lizhang Chen[1] **Du Cheng** **Amy Zhang**[1] **Leqi Liu**[1]

[1]University of Texas at Austin, Austin, TX, USA
[2]Honda Research Institute USA, San Jose, CA, USA
shuozhe.li@utexas.edu

Abstract

Reinforcement learning from verifiable rewards (RLVR) produces strong reasoning models, yet these models can fail catastrophically when the conditioning context is fallible (e.g., corrupted chain-of-thought, misleading partial solutions, or mild input perturbations), because standard RLVR optimizes final-answer correctness only under clean conditioning. We introduce **GASP** (**G**uided **A**dversarial **S**elf-**P**lay), a robustification method that explicitly trains error detection and repair capabilities using only outcome verification. Without human labels or external teachers, **GASP** instantiates an adversarial self-play game within a single model: a **polluter** learns to induce failure through locally coherent corruptions, while an **agent** learns to diagnose and recover under the same corrupted conditioning. To address the scarcity of successful recoveries early in training, we propose *in-distribution repair guidance*, an auxiliary imitation objective on self-generated repairs that increases recovery probability while preserving previously acquired capabilities. Across four open-weight models (1.5B–8B), **GASP** converts strong-but-brittle reasoners into substantially more robust ones that better withstand misleading and perturbed context, while often improving clean accuracy. Further analysis shows that adversarial corruptions create an effective curriculum, and that in-distribution guidance enables rapid recovery learning with minimal representational drift.

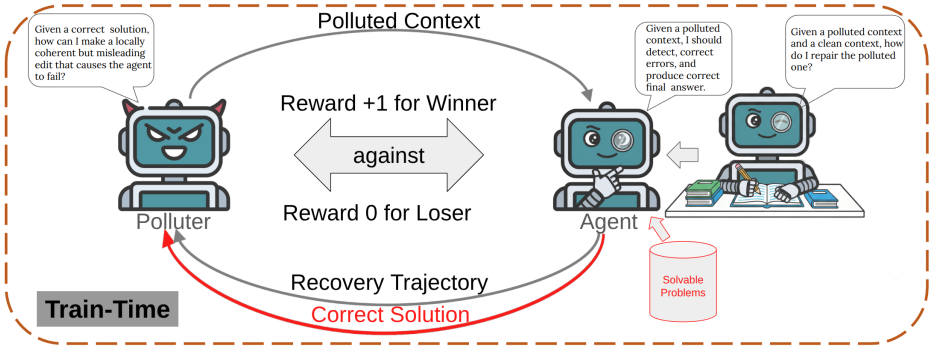


Figure 1: GASP trains robust reasoners that resist misleading context, detect and repair errors, and answer reliably.

1 Introduction

Reinforcement learning from verifiable rewards (RLVR) [50] has turned large language models into strong reasoning systems. By optimizing for final-answer correctness, recent large reasoning models (LRMs), such as DeepSeek-R1 [9], achieve strong performance on math and coding benchmarks and often exhibit "aha-moment" behaviors: pausing, expressing uncertainty, re-checking intermediate

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

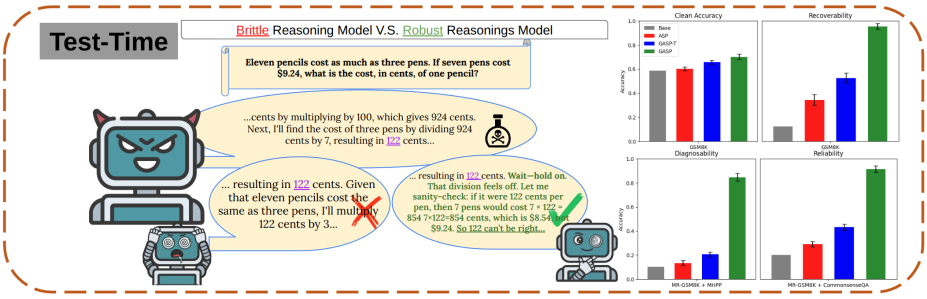


Figure 2: A brittle model follows corruptions and fails, while a robust model detects the inconsistency and repairs the trajectory to reach the correct answer. GASP markedly improves recoverability, diagnosability, reliability, and clean accuracy.

claims, and revising earlier steps [48]. Yet these behaviors remain largely an *incidental byproduct* of outcome optimization. Standard RLVR does not explicitly train a model *when* to distrust its conditioning context, *how* to diagnose an inconsistency, or *how* to recover once a reasoning trajectory has been polluted.

This gap matters because deployed reasoning systems rarely operate with perfectly reliable context. A model may condition on a flawed question, an erroneous partial solution, or a collaborator's misleading reasoning in a multi-agent setting [8]. Existing stress tests expose complementary failures under such fallible context. **Recoverability** tests show that inserting a short locally coherent corruption into an otherwise correct chain-of-thought can cause strong math models to follow the corrupted trajectory and fail [18]. **Diagnosability** benchmarks, such as MR-GSM8K and MR-Bench [52, 53], show that models that solve problems correctly can still fail to locate and explain errors in provided solutions. **Reliability** benchmarks, such as RUPBench [45], further show that mild lexical, syntactic, or semantic perturbations to the input can substantially degrade reasoning. These failures point to a common missing capability: robust reasoning under unreliable conditioning context.

In this work, we ask: **Can we convert a strong-but-brittle LRM into a robust reasoner using only verifiable outcome rewards?** We focus on the regime where base task competence is already present, but robustness to fallible context is not. Our goal is not merely to elicit occasional self-correction at inference time, but to train the core capability needed in noisy reasoning settings: **detect errors, repair them, and still reach the correct answer**.

We introduce **GASP**: **G**uided **A**dversarial **S**elf-**P**lay. GASP instantiates two role-conditioned behaviors within the same model. A **polluter** learns to introduce locally coherent corruptions of the conditioning context that maximize downstream failure, while an **agent** learns to diagnose and neutralize these corruptions while preserving final-answer correctness. Both roles are optimized with GRPO-style objectives using only terminal verifiable rewards. Because the polluter trains against the current agent, self-play induces an adaptive curriculum: as the agent becomes harder to fool, the polluter must discover more effective corruptions, and the agent correspondingly learns stronger verification and repair strategies. Crucially, the agent is trained under the same corrupted conditioning used at test time, so robustness arises from learning rather than from specialized test-time prompting.

A key obstacle is sparse recovery signal. Early in training, the agent rarely succeeds under polluted context, so outcome-only policy optimization often observes groups with no successful trajectories and receives little useful gradient. To break this deadlock, GASP adds **in-distribution repair guidance**: a lightweight imitation term on *self-generated* repair snippets that are high-likelihood under the current policy. This increases the probability of early recoveries, making outcome-based updates informative without relying on human annotations or off-distribution teacher fixes. Empirically, this guidance accelerates recovery learning and better preserves previously acquired capabilities; we further isolate the mechanism with a minimal navigation analogue (§4.4).

Across four open-weight reasoning models from 1.5B to 8B parameters, GASP transforms strong-but-brittle reasoners into robust ones. It improves recoverability under corrupted reasoning context, diagnosability of erroneous solutions, and reliability under natural perturbations, while also improving self-revision and often clean accuracy. Our contributions are:

- **Robust reasoning under fallible context.** We formulate robustness to unreliable conditioning context and unify three stress tests: recoverability, diagnosability, and perturbation reliability.

- **Adversarial self-play with verifiable rewards.** We introduce a two-role game in which a polluter learns locally coherent corruptions that induce failure, while an agent learns to detect and repair them using only terminal correctness rewards.
- **In-distribution repair guidance.** We propose a self-generated guidance objective that increases early recovery probability under sparse outcome rewards without external teachers.
- **Empirical gains and mechanistic analysis.** We show consistent robustness gains across models, improvements in self-revision and clean accuracy, and evidence that in-distribution guidance accelerates recovery learning while reducing representation drift.

2 Preliminaries

A problem instance is a question q with a ground-truth final answer $a^\star$. A reasoning model π_θ generates a token sequence consisting of a chain-of-thought $c = (c_1, \ldots, c_T)$ followed by a final answer a; we write a full generation as a trajectory $\tau = (c, a) \sim \pi_\theta(\cdot \mid q)$. For any ratio $\alpha \in [0, 1]$, let $c_{0:\alpha}$ denote the prefix containing the first α fraction of tokens in c. **Verifiable Rewards**: all RL signals assume the existence of, and depend only on, a final-answer verifier using an outcome reward $r(\tau; q, a^\star) = \mathbb{I}\{a = a^\star\}$. **RL with verifiable rewards**: We view text generation as an episodic decision-making process where the model emits tokens autoregressively until termination. The objective is to maximize expected terminal correctness: $J(\theta) = \mathbb{E}_{(q,a^\star)\sim\mathcal{D},\, \tau\sim\pi_\theta(\cdot|q)}\big[r(\tau; q, a^\star)\big]$.

2.1 Group Relative Policy Optimization (GRPO)

For a generic conditioning context x (e.g., a question q, a polluted steer s^{poll}, or a clean window context for the polluter), GRPO [34] samples a group of trajectories $\{\tau_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ and assigns each trajectory a task-specific terminal score $R_i = \mathcal{R}(\tau_i; x)$. Let $\bar{R} = \text{mean}(\mathbf{R})$ and $\sigma_R = \text{std}(\mathbf{R})$ for $\mathbf{R} = \{R_1, \ldots, R_G\}$. We form group-relative advantages $\hat{A}_i(\mathcal{R}) = \frac{R_i - \bar{R}}{\sigma_R + \epsilon}$, and broadcast them to all tokens: $\hat{A}_{i,t}(\mathcal{R}) = \hat{A}_i(\mathcal{R}),\ \forall t \in \{1, \ldots, |o_i|\}$. The GRPO objective parameterized by $\mathcal{R}$ is

$$J_{\text{GRPO}}(\theta; \mathcal{R}) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(\rho_{i,t}(\theta)\hat{A}_{i,t}(\mathcal{R}), \text{clip}\big(\rho_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\big)\hat{A}_{i,t}(\mathcal{R})\Big)\right]. \tag{1}$$

Optionally, we add a KL regularizer to a reference policy.

3 Method

Our goal is to convert a strong-but-brittle reasoner into a robust one that stays correct even when the conditioning context is unreliable. We operationalize robustness as *diagnosability* (identify errors), *recoverability* (repair and solve), and *reliability* (resist perturbations). **GASP** trains these behaviors via adversarial self-play and outcome verification, rather than test-time prompting. Empirically, we find that models exhibiting stronger "aha-moment" behaviors—e.g., explicit uncertainty, checking, and revision—tend to be substantially more robust in these settings. Motivated by this connection, we target a concrete capability: *given a partially corrupted chain-of-thought, produce the correct final answer*.

Notably, this capability is not entirely absent from current systems. Inference-time–scaled models trained with RLVR sometimes exhibit spontaneous self-correction and revision behaviors as an *incidental byproduct* of optimizing for final-answer correctness. However, since these behaviors are not explicitly trained for, models can still fail catastrophically when confronted with locally misleading intermediate context [18].

We seek to train this capability using only verifiable outcome rewards, without human step-level labels and any external editor/teacher model. Achieving this, we formulate learning as a two-role adversarial self-play game over a shared trajectory. At training time, to decouple robustness learning from competence acquisition, we build self-play episodes from questions the current agent solves reliably, and sample a correct trace to serve as *clean context*. A **polluter** then proposes a *maliciously edited* version of this local window that is intended to induce downstream failure while remaining syntactically coherent. Conditioned on the resulting polluted context, the **agent** must diagnose the inconsistency, repair the trajectory, and reach the correct final answer under the same conditioning used at test time.

3.1 Constructing off-trajectory corruptions

For each problem $(q, a^\star)$ that the agent solves reliably under solo reasoning, we first sample a correct trajectory $\tau = (c, a^\star) \sim \pi_\theta(\cdot \mid q)$, where $c = (c_1, \ldots, c_T)$. We sample a truncation ratio

$\alpha \sim \mathrm{Uniform}([0.00, 0.90])$, and let $c_{0:\alpha}$ denote the prefix containing the first α fraction of tokens in c. This prefix represents the shared reasoning history that will be treated as fixed context. Following the truncation point, we extract a short *clean window* w^{clean}, consisting of the next $\approx 10\%$ of tokens from the same correct trajectory. This window serves as a local, ground-truth continuation that is internally consistent with both the problem and the prefix.

To construct an off-trajectory corruption, the clean window is replaced by a corrupted variant w^{poll} that is locally coherent in style but contains at least one misleading reasoning step and/or perturbed text. The resulting *polluted steer* is $s^{\mathrm{poll}} = \big(q,\ c_{0:\alpha},\ w^{\mathrm{poll}}\big)$, which removes the original clean continuation and exposes the agent to the corrupted context as shown in Figure 2.

3.2 Agent training objective: recover under the polluted steer

Conditioned on s^{poll}, the agent produces an off-trajectory completion $\tau_{\mathrm{off}} \sim \pi_\theta(\cdot \mid s^{\mathrm{poll}})$ and receives only a terminal verifiable reward. We therefore optimize the agent by instantiating the generic GRPO objective $J_{\mathrm{GRPO}}(\theta; \mathcal{R})$ from (1) with recoverability reward $\mathcal{R}^{\mathrm{rec}}(\tau_{\mathrm{off}}; s^{\mathrm{poll}}, a^\star) = \mathbb{I}\{a_{\mathrm{off}} = a^\star\}$ i.e.,

$$J^{\mathrm{rec}}_{\mathrm{GRPO}}(\theta) \;=\; J_{\mathrm{GRPO}}(\theta; \mathcal{R}^{\mathrm{rec}}),$$

where the group-relative advantages $\hat{A}_{i,t}(\mathcal{R}^{\mathrm{rec}})$ are computed from the terminal rewards $\mathcal{R}^{\mathrm{rec}}(\tau_i; s^{\mathrm{poll}}, a^\star)$ within each sampled group. As we show next, optimizing outcome-only $J^{\mathrm{rec}}_{\mathrm{GRPO}}$ is typically insufficient when polluted contexts make recovery trajectories low probability.

3.3 Repair guidance under learning-signal scarcity

Motivation: outcome-only learning is signal-scarce under pollution. Under a polluted steer s^{poll}, the agent initially almost never recovers: outputs typically follow the misleading window and terminate with an incorrect answer. This creates a *learning-signal scarcity* problem for outcome-only GRPO, since group-relative advantages are informative only when some sampled trajectories succeed. Increasing the group size G raises the chance of observing a recovery but is computationally expensive and does not change how rarely recovery behaviors are sampled; see §A.1. This motivates a complementary mechanism that directly increases the probability of sampling successful recoveries under s^{poll} [23, 22], so informative groups arise even with moderate G.

Guided objective. To increase early recovery frequency without increasing G, we augment outcome-only GRPO with an auxiliary *repair guidance* term that upweights short "repair" continuations back to a successful trajectory:

$$J^{\mathrm{rec}}_{\text{G-GRPO}}(\theta) \;=\; J^{\mathrm{rec}}_{\mathrm{GRPO}}(\theta) \;+\; \lambda\, J_{\mathrm{guide}}(\theta).$$

Here $J^{\mathrm{rec}}_{\mathrm{GRPO}}$ is instantiated as in §3.2, and J_{guide} is an imitation-style objective defined on repair snippets under the *deployment* conditioning s^{poll}.

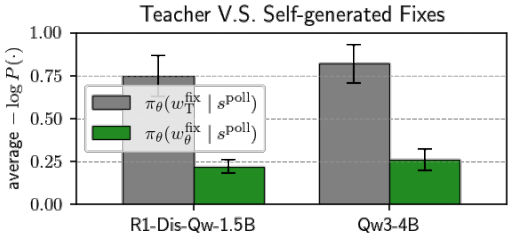


Figure 3: For fixes with similar lengths, teacher (GPT-5) [36] generated fixes have higher negative log-likelihood than self-generated fixes, indicating that the model assigns lower probability to teacher fixes than to its own fixes.

Self-generated in-distribution repair targets. To obtain **high-probability** repair targets, we construct a diagnosis prompt p^{fix}_θ that includes $(q, c_{0:\alpha})$ and contrasts $(w^{\mathrm{clean}}, w^{\mathrm{poll}})$, and sample an on-policy repair $w^{\mathrm{fix}}_\theta \sim \pi_\theta(\cdot \mid p^{\mathrm{fix}}_\theta)$, and then use it as the imitation target $J_{\mathrm{guide}}(\theta) = \mathbb{E}_{w^{\mathrm{fix}}_\theta \sim \pi_\theta(\cdot|p)}[\log \pi_\theta(w^{\mathrm{fix}}_\theta \mid s^{\mathrm{poll}})]$. By construction, this tends to yield repairs with substantially higher likelihood under deployment than teacher fixes: $\pi_\theta(w^{\mathrm{fix}}_\theta \mid s^{\mathrm{poll}}) \gg \pi_\theta(w^{\mathrm{fix}}_{\mathrm{T}} \mid s^{\mathrm{poll}})$, so the same-sized update yields a larger expected improvement, leading to better learning efficiency (verified in §4).

Why are high-probability repair targets better? For a generic policy gradient on a positive sample w^{fix} under the deployment conditioning s^{poll}, we have $g(w^{\mathrm{fix}}) = \omega \nabla_\theta \log \pi_\theta(w^{\mathrm{fix}} \mid s^{\mathrm{poll}})$ with weight $\omega > 0$. A first-order improvement satisfies

$$\Delta J \approx \alpha\, \nabla_\theta J(\theta)^\top g(w^{\mathrm{fix}}) \approx \alpha\, \omega\, \pi_\theta(w^{\mathrm{fix}} \mid s^{\mathrm{poll}}) \left\| \nabla_\theta \log \pi_\theta(w^{\mathrm{fix}} \mid s^{\mathrm{poll}}) \right\|_2^2$$

where the dominant term scales linearly with $\pi_\theta(w^{\mathrm{fix}} \mid s^{\mathrm{poll}})$ (full proof in §A.3). Therefore, for candidate repairs $w^{\mathrm{fix}}_\theta, w^{\mathrm{fix}}_{\mathrm{T}}$ with comparable weights and gradient norms,

$$\pi_\theta(w^{\mathrm{fix}}_\theta \mid s^{\mathrm{poll}}) \;\gg\; \pi_\theta(w^{\mathrm{fix}}_{\mathrm{T}} \mid s^{\mathrm{poll}}) \quad\Longrightarrow\quad |\Delta J(w^{\mathrm{fix}}_\theta)| \;\gg\; |\Delta J(w^{\mathrm{fix}}_{\mathrm{T}})|.$$

Representation preservation. Beyond immediate first-order gain, higher-likelihood self-generated repair targets tend to better preserve internal representations. Since self-generated repairs w_θ^{fix} already have non-negligible likelihood under s^{poll} (Figure 3), increasing $\log \pi_\theta(w_\theta^{\text{fix}} \mid s^{\text{poll}})$ typically requires only less parameter change [59]. By contrast, cloning a lower-likelihood teacher fix with $\pi_\theta(w_{\text{T}}^{\text{fix}} \mid s^{\text{poll}}) \approx 0$ demands a much larger likelihood-ratio adjustment, which can induce larger gradient bursts and greater latent drift in SFT-like updates [12]. We verify this stability/retention effect in §4.

3.4 Adversarial self-play: optimizing the polluter

So far, w^{poll} was an externally constructed corruption. We now generate w^{poll} online in a self-play loop. For each reliably solved problem $(q, a^\star)$, we first sample a clean trajectory $\tau = (c, a^\star) \sim \pi_\theta(\cdot \mid q)$, sample α, and extract the clean window w^{clean} as in Section 3.1. We then prompt the *same* model to act as a *polluter*: conditioned on the local clean context $(q, c_{0:\alpha}, w^{\text{clean}})$, it outputs a corrupted window $w^{\text{poll}} \sim \pi_\theta(\cdot \mid q, c_{0:\alpha}, w^{\text{clean}}; \text{polluter_prompt})$, which defines the polluted steer $s^{\text{poll}} = (q, c_{0:\alpha}, w^{\text{poll}})$.

Polluter reward. After sampling w^{poll}, we run the *agent* under the polluted steer and evaluate the same verifiable correctness reward with $\tau_{\text{off}} \sim \pi_\theta(\cdot \mid s^{\text{poll}})$ and $r_{\text{solve}} = \mathbb{I}\{a_{\text{off}} = a^\star\}$. The polluter receives reward when its corruption causes failure: $r_{\text{poll}} = 1 - r_{\text{solve}}$. To prevent degenerate polluter strategies, we apply a non-LLM validity filter (gcld3 [1], wordfreq, gibberish_detector [33]). Corruptions with non-linguistic artifacts (e.g., repetition or gibberish) receive zero reward, ensuring that rewarded corruptions remain coherent reasoning-level perturbations.

Polluter objective. We optimize the polluter by instantiating the generic GRPO objective $J_{\text{GRPO}}(\theta; \mathcal{R})$ from (1) over corrupted-window generation. For each clean local context $(q, c_{0:\alpha}, w^{\text{clean}})$, the polluter samples a corrupted window

$$w^{\text{poll}} \sim \pi_\theta(\cdot \mid q, c_{0:\alpha}, w^{\text{clean}}; \text{polluter_prompt}),$$

which induces the polluted steer $s^{\text{poll}} = (q, c_{0:\alpha}, w^{\text{poll}})$. After rolling out the agent under s^{poll}, the polluter receives terminal reward $\mathcal{R}^{\text{poll}}(w^{\text{poll}}; q, c_{0:\alpha}, w^{\text{clean}}, a^\star) = 1 - \mathbb{I}\{a_{\text{off}} = a^\star\}$, where $\tau_{\text{off}} \sim \pi_\theta(\cdot \mid s^{\text{poll}})$. We then optimize

$$J_{\text{GRPO}}^{\text{poll}}(\theta) = J_{\text{GRPO}}(\theta; \mathcal{R}^{\text{poll}}),$$

with group-relative advantages computed across G_{poll} sampled corruptions for the same clean local context.

Overall self-play updates (GASP). Training alternates two GRPO-style updates using the same parameters θ: (i) an *agent* update that maximizes recoverability under polluted context (GRPO plus the self-generated repair guidance term from Section 3.3), and (ii) a *polluter* update that maximizes $J_{\text{GRPO}}^{\text{poll}}$. Concretely, we alternate ascent steps

$$\theta \leftarrow \theta + \eta_{\text{rec}} \nabla_\theta J_{\text{G-GRPO}}^{\text{agent}}(\theta), \quad \theta \leftarrow \theta + \eta_{\text{poll}} \nabla_\theta J_{\text{GRPO}}^{\text{poll}}(\theta),$$

where $J_{\text{G-GRPO}}^{\text{agent}}(\theta)$ denotes the guided objective (e.g., $J_{\text{GRPO}}^{\text{rec}}(\theta) + \lambda J_{\text{guide}}(\theta)$). When optimizing the polluter, we treat r_{solve} as a black-box outcome of the agent rollout and do not backpropagate through τ_{off}. In practice, we often observe a curriculum-like dynamic: as the agent becomes harder to fool, the polluter must discover increasingly effective local corruptions, and the agent correspondingly learns stronger error-detection and repair behaviors under the same deployment conditioning s^{poll}.

4 Experiments

We evaluate GASP as a training procedure for turning strong-but-brittle LRMs into robust reasoners using only outcome reward without teacher/human labels. Our experiments are organized around three questions:

- **Q1: Robustness** and **downstream utility.** Does GASP improve *diagnosability*, *recoverability*, and *reliability*, and do these robustness gains translate into better clean accuracy and *self-revision* (post-hoc correction when prompted with the model's own incorrect solution)?
- **Q2: Efficiency.** Under a fixed rollout budget, does in-distribution repair guidance reduce the exploration burden for agent—producing recoveries sooner and reaching higher performance faster—than outcome-only GRPO and teacher-guided repair?
- **Q3: Representation Drift.** Does in-distribution guidance reduce representation drift and better preserve out-of-domain performance relative to off-distribution teacher guidance?

Model	Method	Clean Accuracy	Recoverability	Diagnosability		Reliability		Self-Revision	Avg
		GSM8K	GSM8K	MR-GSM8K	MHPP	GSM8K	CQA	Δ GSM8K	Avg
R1-Distill-Qwen-1.5B	Initial ckpt	58.7	12.4	12.1	8.4	24.3	16.2	0.2	18.9
	ASP	60.2 ± 1.6	34.4 ± 4.5	15.8 ± 2.1	11.2 ± 1.8	38.7 ± 2.3	19.5 ± 2.1	2.1 ± 0.3	26.0
	GASP-T	65.8 ± 1.7	52.7 ± 4.2	24.1 ± 2.2	17.3 ± 1.9	57.5 ± 2.7	28.8 ± 2.3	2.8 ± 0.4	35.6
	GASP	70.3 ± 2.4	95.6 ± 2.2	87.4 ± 3.1	82.1 ± 3.5	91.2 ± 2.8	91.7 ± 2.9	6.1 ± 0.5	**74.9**
DeepScaleR-1.5B	Initial ckpt	67.9	14.3	13.8	9.1	25.6	17.4	0.2	21.2
	ASP	69.2 ± 1.2	36.7 ± 5.1	18.3 ± 2.4	13.5 ± 2.0	41.4 ± 2.5	22.1 ± 2.3	1.9 ± 0.3	29.0
	GASP-T	70.8 ± 1.4	56.0 ± 4.6	27.8 ± 2.6	20.6 ± 2.1	61.8 ± 2.9	33.1 ± 2.5	2.6 ± 0.4	39.1
	GASP	72.4 ± 2.1	98.6 ± 1.1	90.2 ± 2.7	85.8 ± 3.2	94.3 ± 2.1	92.6 ± 2.4	8.2 ± 0.6	**77.4**
Qwen3-4B	Initial ckpt	77.2	13.8	15.4	11.2	27.9	19.8	2.3	23.9
	ASP	79.2 ± 1.1	40.5 ± 5.2	22.7 ± 2.6	17.9 ± 2.3	45.6 ± 2.8	26.3 ± 2.5	4.6 ± 0.5	33.8
	GASP-T	81.4 ± 1.2	61.4 ± 4.0	34.9 ± 2.7	27.4 ± 2.4	67.5 ± 2.6	39.1 ± 2.7	6.3 ± 0.6	45.4
	GASP	82.9 ± 1.9	98.2 ± 1.2	92.8 ± 2.3	88.5 ± 2.9	96.1 ± 1.8	94.7 ± 2.1	9.3 ± 0.7	**80.4**
Qwen3-8B (LoRA)	Initial ckpt	84.9	14.7	18.2	13.5	33.3	22.1	4.2	27.3
	ASP	87.3 ± 1.5	44.4 ± 2.1	26.9 ± 2.8	21.4 ± 2.5	59.8 ± 2.9	30.2 ± 2.7	6.4 ± 0.6	39.5
	GASP-T	89.9 ± 1.6	66.9 ± 2.4	40.7 ± 2.9	32.5 ± 2.6	88.6 ± 2.7	44.1 ± 2.6	8.8 ± 0.7	53.2
	GASP	93.7 ± 3.2	98.7 ± 0.8	95.1 ± 1.9	91.6 ± 2.4	97.8 ± 1.1	96.3 ± 1.6	10.7 ± 0.8	**83.4**

Table 1: **Main robustness results (pass@1).** Mean ± std over 3 training runs for ASP/GASP-T/GASP; **Initial ckpt** is a single evaluation. **Clean Accuracy**: GSM8K under clean prompting. **Recoverability**: GSM8K accuracy under corrupted context (held-out GPT-5 polluter). **Diagnosability**: **ACCreason** on MR-GSM8K and MHPP [52, 53]. **Reliability**: accuracy on perturbed GSM8K and CommonsenseQA from RUPBench [45]. **Self-Revision**: improved GSM8K accuracy on the failed questions after self-revision [41]. **Avg**: unweighted mean of the seven metrics. GASP-T uses teacher-generated (GPT-5) repair snippets in place of self-generated repair snippets. LoRA [11] were used for 8B model §E. **To isolate robustness from base-task competence, all robustness metrics except Clean Accuracy are evaluated on reliably solved subset.** Full benchmark in Table 7.

4.1 Experimental setup

Models. We run experiments on four open-weight instruction-tuned reasoning models spanning 1.5B–8B parameters: DeepSeek-R1-Distill-Qwen-1.5B, DeepScaleR-1.5B, Qwen3-4B, and Qwen3-8B. Note, these are already strong reasoning and instruction following models, which makes them well-suited for role-conditioned self-play: under a POLLUTE prompt, the same backbone can reliably generate plausible corruptions that induce failures in the AGENT role, creating a non-degenerate training game with informative adversaries.

Training questions and solvable subset. We sample 7.5k problems from the MATH training split and restrict training to instances the *initial checkpoint* solves reliably: for each candidate, we draw K=4 solutions at temperature 0.7 and keep it only if all final answers are correct. For each retained $(q, a^\star)$ during training, we sample a clean on-policy trajectory $\tau = (c, a^\star) \sim \pi_\theta(\cdot|q)$, sample $\alpha \sim \mathrm{Uniform}([0.00, 0.90])$, and extract the subsequent clean window w^{clean} (the next $\approx 10\%$ of tokens, capped to a fixed length). The polluter generates w^{poll} online from this context, and the agent learns recovery under the polluted steer $s^{\mathrm{poll}} = (q, c_{0:\alpha}, w^{\mathrm{poll}})$. To rule out memorization and corruption-template leakage, all evaluation instances are strictly disjoint from training.

Prompts and role conditioning. Both roles are implemented by prompting the same backbone model. The AGENT uses the standard deployment system prompt *unchanged* (no added "be robust" or "check for errors" instructions), while POLLUTE uses a prompt with a constrained edit budget that encourages non-gibberish but misleading corruptions. Full prompts are in §B; reported robustness gains come from training, not test-time prompt engineering.

Baselines. In the main results table, we compare four methods: (i) the initial checkpoint (no additional training), (ii) ASP, adversarial self-play with outcome-only GRPO under polluted steers and *without* any repair guidance, (iii) GASP-T, which replaces self-generated repair targets with teacher-generated repair snippets from GPT-5, and (iv) GASP, our full method with self-generated repair guidance. Additional comparisons including GPT-5 polluter and ablations that disable polluter updates (fixed corruptions) are reported in §A.9. All methods are run under matched rollout budgets by controlling group sizes and maximum generation lengths.

Unless otherwise stated, we use the same truncation distribution $\alpha \sim \mathrm{Uniform}([0.00, 0.90])$, window-length cap, and final-answer verifier across all methods. Full hyperparameters (GRPO clip, KL coefficient if used, group sizes, learning rates, and training steps) are given in §E.

4.2 Evaluation benchmarks and metrics

We evaluate four complementary capabilities and summarize the main results in Table 1. To isolate robustness to fallible context from base-task competence, we report *Recoverability*, *Diagnosability*, and *Reliability* primarily on a *clean-solved subset* of each benchmark: for every instance, we sample K=4 solutions under the standard (clean) prompt (temperature 0.7) and retain it only if all final answers are correct. This conditioning ensures that failures reflect collapse under misleading context rather than inability to solve the underlying problem. For completeness, we also report performance on the full benchmarks in Appendix §A.9.

Recoverability. We evaluate robustness to misleading intermediate context on GSM8K. For each clean-solved instance, we sample a correct solution and draw a truncation ratio $\alpha \sim \text{Uniform}([0.00, 0.90])$. We keep the prefix $c_{0:\alpha}$, replace the following reasoning window with a locally coherent corruption, and measure whether the model still reaches the correct final answer. Our primary evaluation uses a held-out GPT-5 polluter; additional cross-corruption evaluations test transfer to unseen corruption sources.

Eval Polluter	Initial	ASP	GASP-T	GASP
Internal self-play	15.2	52.4 ± 3.8	78.6 ± 2.9	**99.1** ± 0.6
Held-out GPT-5	13.8	40.5 ± 5.2	61.4 ± 4.0	**98.2** ± 1.2
Sonnet-4.6	12.9	37.8 ± 4.7	58.7 ± 3.9	**94.6** ± 1.8
Gemini 3	14.4	44.9 ± 4.3	67.3 ± 3.5	**96.8** ± 1.5
Human-written	10.7	39.2 ± 3.6	55.8 ± 3.2	**88.4** ± 2.7

Table 2: Cross-corruption recoverability generalization (%, pass@1) on Qwen3-4B. GASP trains only on internal self-play corruptions yet transfers to unseen corruption sources.

Diagnosability. We evaluate whether the model can judge a provided solution, identify the first erroneous step, and explain the error. On MR-GSM8K and MR-Ben's coding subset (MHPP), we follow the benchmark protocols and report **ACCreason** (first-error-step + correct explanation) [52, 53]. Exact prompting and answer extraction are in §C.

Reliability. We evaluate robustness to natural lexical/syntactic/semantic perturbations using the GSM8K and CommonsenseQA subsets of RUPBench [45].

Self-revision. We measure whether a model can correct its own mistakes when shown its erroneous solution. On GSM8K, we first run the model once under the standard (clean) prompt and collect instances it answers incorrectly. For each such instance, we provide the original question together with the model's generated (incorrect) solution and prompt it to revise and output a corrected final answer, following Tie et al. [41]. We report the resulting accuracy on the model's incorrect subset.

Hard-math transfer. To test transfer beyond GSM8K, we evaluate GASP zero-shot on MATH-500 and AIME for Qwen3-4B. Table 3 shows clean accuracy is preserved while recoverability improves under held-out GPT-5 corruptions.

	Clean		Corrupted	
Eval	Initial	GASP	Initial	GASP
MATH-500	72.2	79.8 ± 1.2	11.1	93.6 ± 1.9
AIME	46.7	53.3 ± 3.3	12.5	75 ± 6.3

Table 3: Hard-math transfer for Qwen3-4B

4.3 Self-play dynamics and polluter validity.

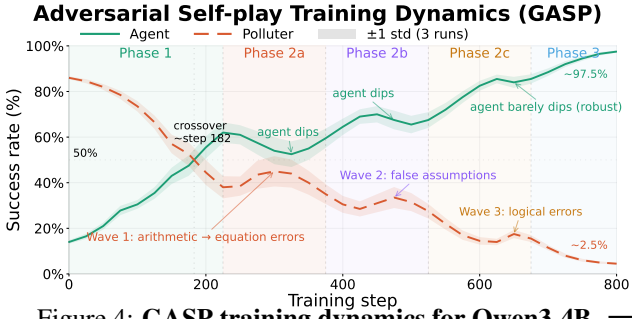


Figure 4: **GASP training dynamics for Qwen3-4B.**

Figure 4 shows GASP enables a non-degenerate game throughout training as agent success steadily improves: three adaptation waves of increasingly sophisticated corruptions trigger shrinking agent dips (−10%→ −5%→ − 2%) before converging to ~97.5% and ~2.5%, with variance peaking near crossover (~step 182).

A potential failure mode of adversarial self-play is that the polluter may exploit degenerate attacks such as gibberish, rather than producing meaningful reasoning errors. To test this, we manually audit 50 Qwen3-4B polluter outputs at each of three checkpoints. Table 4 shows that most corruptions are coherent local reasoning errors, and the fraction of irrelevant or gibberish

Corruption type	Step-0	400	600	800
Arithmetic slip	28%	18%	14%	10%
Wrong equation	18%	34%	24%	16%
False assumption	16%	22%	36%	24%
Irrelevant info	14%	6%	6%	4%
Logical error	18%	18%	18%	44%
Gibberish	6%	2%	2%	2%

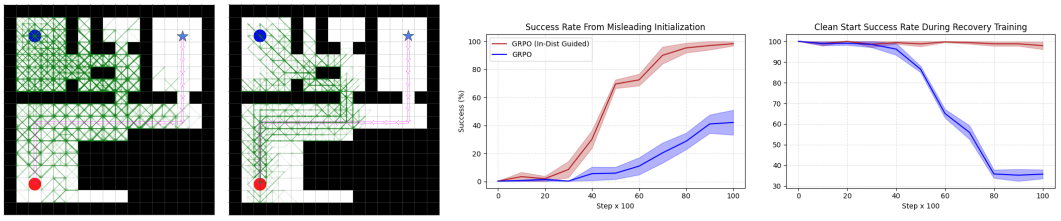
Table 4: Corruption types across training.

corruptions decreases after self-play begins. This suggests that the polluter does not collapse to trivial artifacts; instead, it increasingly discovers reasoning-level corruptions such as wrong transformations and false intermediate assumptions.

4.4 A navigation analogue: guidance turns rare repairs into learnable signal

To isolate why repair guidance helps beyond outcome-only RL, we study a sparse-reward navigation [54] analogue of off-trajectory reasoning. The policy first learns a reliable "rail" from a clean start to the goal. We then place the same policy at a misleading off-rail start: locally, the new state appears plausible, but reaching the goal requires first backtracking through a narrow junction to rejoin the previously learned rail. Thus, as in corrupted chain-of-thought recovery, the agent already possesses the downstream skill but rarely samples the short repair maneuver needed to access it.

Figure 5: **Guidance accelerates off-trajectory recovery and preserves retention.** The policy is first trained from a clean start ★, then fine-tuned from a misleading off-rail start ● under sparse terminal reward. (a) Outcome-only GRPO explores broadly and drifts from the original rail. (b) Guided GRPO learns a short repair back to the rail and reuses the prior solution. (c) Misleading-start success improves faster with guidance. (d) Clean-start retention remains higher, indicating less drift. Curves show mean ± std over 5 seeds, with 10 rollouts per checkpoint.



(a) GRPO (b) Guided GRPO (c) (d)

We compare outcome-only GRPO against guided GRPO under this misleading-start distribution. Both methods receive the same sparse terminal reward. The guided variant adds an imitation term on on-policy repair segments: whenever a rollout reaches the set $\mathcal{R}$ of states visited by the original clean-start policy, we store the prefix that returned to $\mathcal{R}$ in a buffer $\mathcal{B}_{\text{fix}}$ and optimize $J(\theta) = J_{\text{GRPO}}(\theta) + \lambda \mathbb{E}_{\tau \sim \mathcal{B}_{\text{fix}}} \left[\sum_{t=0}^{|\tau|} \log \pi_\theta(a_t \mid s_t) \right]$. Because these repair snippets are harvested from the current policy, they are in-distribution: the auxiliary update amplifies behaviors the model can already occasionally produce, rather than forcing an off-policy solution.

Figure 5 illustrates the resulting optimization dynamics. GRPO-only spends many updates exploring the misleading corridor, and even after occasionally rejoining the rail, it often fails to exploit the previously learned route, indicating both sparse recovery signal and drift from the clean-start behavior. In contrast, guided GRPO quickly learns a short maneuver back to $\mathcal{R}$ and then reuses the existing rail to reach the goal. This yields faster misleading-start success while preserving clean-start retention.

This analogue captures the core failure mode of outcome-only recovery training. When the probability p_θ of sampling a full repair-then-solve trajectory is small, most GRPO groups of size G contain no successful rollout, so group-relative advantages provide little directed signal. Guidance breaks this deadlock by increasing the frequency of partial repairs that return the agent to a region where its prior competence is useful. Once such recoveries appear often enough, outcome RL can propagate credit through the full sequence: repair → reuse prior skill → solve.

4.5 Efficiency and Representation Retention in LLM

The maze analogue isolates *why* recovery training is hard and what guidance can do. We now turn to the LLM setting to study *how the source of guidance* shapes both learning efficiency and representational stability when training recovery behaviors on top of an already capable models.

Self-guided GASP improves MR-GSM8K substantially earlier and reaches a higher final score than GASP-T as shown in Figure 6. This efficiency advantage coincides with a clear stability difference: GASP-T exhibits much larger *layerwise representation shift*. Concretely, we compute $d^{(*)}$ by (i) collecting mean hidden-state representations on a fixed TruthfulQA probe set, (ii) projecting them onto a 2D PCA basis, and (iii) measuring the Euclidean distance between the pre- and post-training centroids in this PCA space. Larger $d^{(*)}$ therefore indicates a larger global movement of the model's representations on the probe distribution.

Figure 6: **Guidance improves learning efficiency.** Self-guided GASP reaches substantially higher *recoverability*, *diagnosability*, and *reliability* earlier than teacher-guided GASP-T and unguided self-play ASP. Each reports the mean success rate averaged over the corresponding benchmark suite for that capability §4.2 at the given training step; shaded regions denote ±1 std over 3 runs.

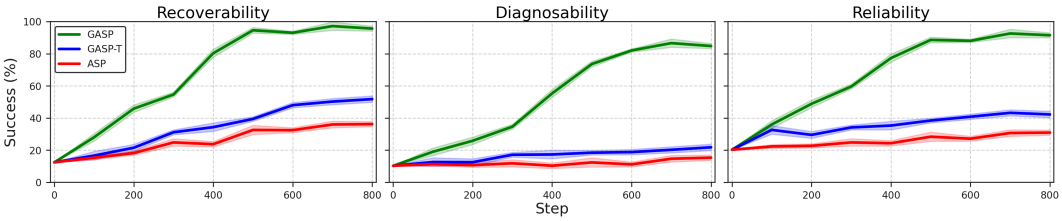
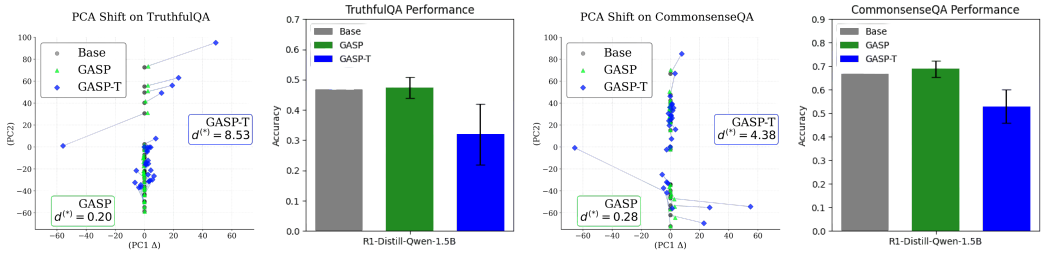


Figure 7: **Representation drift correlates with QA drops.** (a,c) show *PCA shift* of layerwise mean hidden states on a fixed TruthfulQA / CommonsenseQA probe set. Each marker is a transformer layer i, plotted as $(\Delta m_{i,1},\, m_{i,2})$ after 2D PCA, where $\Delta m_{i,1}$ is the layer's mean PC1 change from the base model (base lies on ΔPC1=0). The boxed $d^{(*)}$ is the Euclidean distance between base vs. post-training centroids in PCA space (global drift). (b,d) show methods with larger drift (e.g., GASP-T) suffer larger accuracy drops, while low-drift GASP largely preserves accuracy.



(a) (b) (c) (d)

This representation movement tracks downstream interference: the method with larger shift (teacher guided GASP) also suffers a larger drop in TruthfulQA accuracy, while self-guided GASP largely preserves it. Overall, self-generated repair guidance is more continual-learning friendly in our setting: it achieves faster recovery learning while inducing smaller representation drift and better retention on an external QA benchmark.

5 Additional Experiments

We report several additional analyses that further examine the robustness gains of GASP and the role of repair guidance.

Full-benchmark performance. Our primary recoverability evaluation conditions on problems that the initial checkpoint already solves reliably, thereby isolating robustness from base-task competence. We additionally decompose performance on the full GSM8K benchmark using this fixed partition. The reliably solved subset constitutes 56% of the benchmark. On the complementary subset, corrupted-context accuracy improves from 2.7% to 59.6% (+56.9 points). Approximately 65% of the aggregate improvement comes from the reliably solved subset and 35% from the complementary subset. We interpret the latter conservatively as end-to-end improvement, since robustness and newly acquired task competence are entangled on initially unsolved problems.

Generalization across corruption constructions. Table 5 evaluates recovery under a structurally different corruption protocol. Instead of locally replacing a window from a correct trajectory, we retain an MR-GSM8K erroneous solution through its annotated first error and ask the model to continue without indicating that an error is present. GASP improves recoverability from 12% to 84%, substantially outperforming ASP and GASP-T. Together with Table 2, this shows generalization across both corruption sources and corruption constructions.

Does recovery reflect repair or re-solving? We manually inspect 100 randomly sampled successful GASP recoveries under held-out corruptions. In all 100 cases, the model identifies the corrupted intermediate step, corrects it, and resumes reasoning from the repaired state; we observe $0/100$ cases

Corruption protocol	Initial	ASP	GASP-T	GASP
Local window replacement	13.8	40.5	61.4	**98.2**
Erroneous-solution prefix	12.0	30.0	47.0	**84.0**

Table 5: Recoverability (%) under different corruption constructions on Qwen3-4B. The erroneous-prefix protocol uses neither a clean reference trajectory nor local window replacement.

Guidance schedule	Rec.	MR-GSM8K	RUP-GSM8K	Clean
Anneal $0.07 \to 0$	**98.2±1.2**	**92.8±2.3**	**96.1±1.8**	**82.9±1.9**
Fixed $\lambda = 0.07$	91.1±1.5	84.3±2.1	88.6±2.2	76.2±1.7

Table 6: Effect of the repair-guidance schedule on Qwen3-4B (%).

that discard the trajectory and restart from the original question. A question-only control achieves 99.4% on the same subset, confirming that re-solving is feasible but is not the strategy observed in these successful GASP trajectories. Separately, among evaluation problems that change from incorrect under the initial checkpoint to correct after GASP, 96.3% exhibit an explicit detect-and-repair event, while 3.7% reach the correct answer without an observable repair event.

Effect of annealing repair guidance. We branch from the same Qwen3-4B checkpoint immediately before the original guidance-annealing phase and compare the original linear annealing of λ from 0.07 to 0 against keeping $\lambda = 0.07$ fixed. All other settings are unchanged. As shown in Table 6, annealing consistently improves final performance, supporting the role of guidance as an early-training scaffold that can be removed once successful recoveries become sufficiently common.

Empirical validation of the guidance analysis. Finally, we measure the three factors in the leading first-order term using paired self-generated and GPT-5-generated repairs from an early Qwen3-4B checkpoint. Repair lengths are approximately matched. Self-generated repairs have substantially higher deployment likelihood (sequence NLL 36 vs. 68), while GPT-5 repairs have higher downstream success probability (0.88 vs. 0.78) and a $1.6\times$ larger squared score-gradient norm. Combining all three factors, the relative log leading term is -36.25 for self-generated guidance versus -67.66 for GPT-5 guidance, favoring self-generated repairs by approximately 31.4 log units. Thus, deployment likelihood is only one factor in the comparison, but dominates the measured tradeoff in this setting.

6 Related Work

A growing line of work studies how models can *improve their own reasoning* by generating feedback, revisions, or new training data without relying on additional human annotations [58, 19]. At inference time, iterative self-revision frameworks such as SELF-REFINE [26] and REFLEXION[35] treat the model as its own critic: the system produces an initial solution, generates natural-language feedback or reflections from task outcomes, and uses this feedback to revise subsequent attempts (often via an explicit memory buffer) without weight updates. At training time, self-bootstrapping methods such as STAR [51] iteratively generate rationales, filter by final-answer correctness, and fine-tune on the resulting self-generated traces, while SCORE [17] more directly targets *correction* by training multi-turn self-correction policies with online RL and addressing distribution mismatch and collapse in offline self-correction fine-tuning. In parallel, self-play fine-tuning methods such as SPIN [7] refine a model by iteratively generating synthetic responses and optimizing objectives that compare synthetic outputs to human-annotated data; follow-up work analyzes instability in pairwise self-play objectives and proposes more stable variants [44]. Broader discussion of Reinforcement fine-tuning (RFT) at Appendix A.8.

7 Conclusion and Limitations

We introduced GASP, which trains robust reasoning through adversarial self-play using only verifiable outcome rewards. While GASP improves robustness and often preserves clean performance, its scope is limited by the need for verifiable answers, localized corruptions, substantial self-play compute, and evaluation beyond the studied reasoning settings; More on limitations see Appendix G.

References

[1] Marc Bertin and Iana Atanassova. Citing foreign language sources: an analysis of the s2orc dataset. In *13th International Workshop on Bibliometric-enhanced Information Retrieval@ 45th European Conference on Information Retrieval*, volume 3617, pages 66–76, 2023.

[2] Lizhang Chen, Bo Liu, Kaizhao Liang, and Qiang Liu. Lion secretly solves constrained optimization: As Lyapunov predicts. In *International Conference on Learning Representations (ICLR)*, pages 35404–35439, 2024. URL `https://openreview.net/forum?id=e4xS9ZarDr`. Spotlight.

[3] Lizhang Chen, Jonathan Li, Chen Liang, Ni Lao, and Qiang Liu. Training-free looped transformers. *arXiv preprint arXiv:2605.23872*, 2026. URL `https://arxiv.org/abs/2605.23872`.

[4] Lizhang Chen, Jonathan Li, Kaizhao Liang, Baiyu Su, Cong Xie, Nuo Wang Pierse, Chen Liang, Ni Lao, and Qiang Liu. Cautious weight decay. In *International Conference on Learning Representations (ICLR)*, 2026. URL `https://openreview.net/forum?id=Gwe6gbGng5`.

[5] Lizhang Chen, Jonathan Li, and Qiang Liu. Muon optimizes under spectral norm constraints. *Transactions on Machine Learning Research (TMLR)*, 2026. ISSN 2835-8856. URL `https://openreview.net/forum?id=Blz4hjxLwU`.

[6] Lizhang Chen, Jonathan Li, Qi Wang, Runlong Liao, Shuozhe Li, Chen Liang, Ni Lao, and Qiang Liu. ϕ-balancing for mixture-of-experts training. In *International Conference on Machine Learning (ICML)*, 2026. URL `https://openreview.net/forum?id=DZbzIOguz4`.

[7] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[8] Jiaxun Cui. *Communication and generalization in multi-agent learning*. PhD thesis, 2025.

[9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[10] Bodun Hu, Shuozhe Li, Saurabh Agarwal, Myungjin Lee, Akshay Jajoo, Jiamin Li, Le Xu, Geon-Woo Kim, Donghyun Kim, Hong Xu, et al. Stitchllm: Serving llms, one block at a time. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26887–26903, 2025.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[12] Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

[13] Chao Jiang, Zhiling Wang, Huawei Liang, Shijing Zhang, and Shuhang Tan. Motion state estimation based on multi-sensor fusion and noise covariance estimation. In *International Conference on Autonomous Unmanned Systems*, pages 706–716. Springer, 2021.

[14] Chao Jiang, Zhiling Wang, Shuhang Tan, and Huawei Liang. A new adaptive noise covariance matrices estimation and filtering method: Application to multi-object tracking. *arXiv preprint arXiv:2112.12082*, 2021.

[15] Chao Jiang, Zhiling Wang, Huawei Liang, and Shuhang Tan. A fast and high-performance object proposal method for vision sensors: Application to object detection. *IEEE Sensors Journal*, 22(10):9543–9557, 2022.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

[18] Aochong Oliver Li and Tanya Goyal. Off-trajectory reasoning: Can llms collaborate on reasoning trajectory? *arXiv preprint arXiv:2510.06410*, 2025.

[19] Shuozhe Li, Daniel Yamakov, and Christine Julien. Smart homes reimagined: Human-guided llms for efficient and secure home automation.

[20] Kaizhao Liang, Bo Liu, Lizhang Chen, and Qiang Liu. Memory-efficient LLM training with online subspace descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 64412–64432, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/760d09bcc06b949f5ac4b6a918739aa8-Abstract-Conference.html.

[21] Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=zBPZeRjfgu.

[22] Runlong Liao, Baiyu Su, Lizhang Chen, and Qiang Liu. Signed rectified flow: Negativity-controlled generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026. URL https://arxiv.org/abs/2607.18516.

[23] Runlong Liao, Jian Yu, Baiyu Su, Chi Zhang, Lizhang Chen, and Qiang Liu. Momentum guidance: Plug-and-play guidance for flow models. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 136–153. Springer, 2026. doi: 10.1007/978-3-032-37517-9_8. URL https://doi.org/10.1007/978-3-032-37517-9_8.

[24] Bo Liu, Lemeng Wu, Lizhang Chen, Kaizhao Liang, Jiaxu Zhu, Chen Liang, Raghuraman Krishnamoorthi, and Qiang Liu. Communication efficient distributed training with distributed Lion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 18388–18415, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/20cea6c1b36ae5f69c48427a68b67fbc-Abstract-Conference.html.

[25] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[26] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

[27] Anirudh Nair, Fulin Jiang, Kang Hou, Zifan Xu, Shuozhe Li, Xuesu Xiao, and Peter Stone. Dynabarn: Benchmarking metric ground navigation in dynamic environments. In *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 347–352. IEEE, 2022.

[28] Son Nguyen, Lizhang Chen, Bo Liu, and Qiang Liu. Memory-efficient optimization with factorized Hamiltonian descent. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pages 2863–2871, 2025. URL https://proceedings.mlr.press/v258/nguyen25e.html.

[29] Son Nguyen, Bo Liu, Lizhang Chen, and Qiang Liu. Improving adaptive moment optimization via preconditioner diagonalization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *Proceedings of Machine Learning Research*, pages 4312–4320, 2026. URL https://proceedings.mlr.press/v300/nguyen26e.html.

[30] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[31] Manas Pathak, Xingyao Chen, Shuozhe Li, Amy Zhang, and Liu Leqi. Filtered reasoning score: Evaluating reasoning quality on a model's most-confident traces. *arXiv preprint arXiv:2604.11996*, 2026.

[32] Bowen Peng, Lizhang Chen, Baiyu Su, Jeffrey Quesnelle, Diederik P. Kingma, and Qiang Liu. DeMo: Decoupled momentum optimization. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=U9oewpa7cn.

[33] Guido Rossum. Python reference manual, 1995.

[34] 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.

[35] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[36] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[37] Shuhang Tan, Hengyu Liu, and Zhiling Wang. Cmsg: Cross-media semantic-graph feature matching algorithm for autonomous vehicle relocalization. In *2023 International Symposium on Intelligent Robotics and Systems (ISoIRS)*, pages 5–11. IEEE, 2023.

[38] Shuhang Tan, Jayson Sia, Paul Bogdan, and Radoslav Ivanov. Analyzing neural network robustness using graph curvature. In *2024 International Conference on Assured Autonomy (ICAA)*, pages 110–113. IEEE, 2024.

[39] Shuhang Tan, Zhiling Wang, and Yan Zhong. Rcp-rf: A comprehensive road-car-pedestrian risk management framework based on driving risk potential field. *IET Intelligent Transport Systems*, 18(12):2618–2640, 2024.

[40] Shuhang Tan, Jayson Sia, Paul Bogdan, and Radoslav Ivanov. Post-training neural network pruning using graph curvature. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=kwACVY73Ug.

[41] Guiyao Tie, Zenghui Yuan, Zeli Zhao, Chaoran Hu, Tianhe Gu, Ruihang Zhang, Sizhe Zhang, Junran Wu, Xiaoyue Tu, Ming Jin, et al. Can llms correct themselves? a benchmark of self-correction in llms. *arXiv preprint arXiv:2510.16062*, 2025.

[42] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

[43] Xuesong Wang, Hengrui Zhang, Jiazhi Zhang, C. L. Philip Chen, and Yuhu Cheng. Pcdt: Pessimistic critic decision transformer for offline reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 55(10):7247–7258, 2025. doi: 10.1109/TSMC.2025.3583392.

[44] Yibo Wang, Hai-Long Sun, Guangda Huzhang, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Lijun Zhang. Triplets better than pairs: Towards stable and effective self-play fine-tuning for llms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[45] Yuqing Wang and Yun Zhao. Rupbench: Benchmarking reasoning under perturbations for robustness evaluation in large language models. *arXiv preprint arXiv:2406.11020*, 2024.

[46] Koutian Wu, Zesen Huang, and Hejia Geng. What git does not record: Provenance legibility in two ai-assisted research repositories. *arXiv preprint arXiv:2610.28655*, 2026.

[47] Haoran Xu, Shuozhe Li, Harshit Sikchi, Scott Niekum, and Amy Zhang. An optimal discriminator weighted imitation perspective for reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pages 25336–25356, 2025.

[48] Shu Yang, Junchao Wu, Xin Chen, Yunze Xiao, Xinyi Yang, Derek F Wong, and Di Wang. Understanding aha moments: from external observations to internal mechanisms. *arXiv preprint arXiv:2504.02956*, 2025.

[49] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[50] Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025.

[51] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[52] Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, and Jiaya Jia. Mr-gsm8k: A meta-reasoning benchmark for large language model evaluation. *arXiv preprint arXiv:2312.17080*, 2023.

[53] Zhongshen Zeng, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, et al. Mr-ben: A comprehensive meta-reasoning benchmark for large language models. *arXiv e-prints*, pages arXiv–2406, 2024.

[54] Hengrui Zhang, Yuhu Cheng, C. L. Philip Chen, and Xuesong Wang. Diffusion subgoal planning for long-horizon offline goal-conditioned reinforcement learning, 2026. URL `https://arxiv.org/abs/2609.34575`.

[55] Zheyu Zhang, Shuo Yang, and Gjergji Kasneci. Consolidating rewarded perturbations for llm post-training. *arXiv preprint arXiv:2605.31494*, 2026.

[56] Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

[57] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[58] Ruiyang Zhou, Shuozhe Li, Amy Zhang, and Liu Leqi. Expo: Unlocking hard reasoning with self-explanation-guided reinforcement learning. *arXiv preprint arXiv:2507.02834*, 2025.

[59] Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, et al. The path not taken: Rlvr provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

A Appendix

A.1 Learning Signal Scarcity

Formally, define the agent's *recovery rate* under the polluted steer as

$$p_\theta \triangleq \Pr_{\tau_{\text{off}} \sim \pi_\theta(\cdot|s^{\text{poll}})}\big[a_{\text{off}} = a^\star\big],$$

i.e., the probability that an on-policy rollout conditioned on s^{poll} ends with the correct final answer. GRPO samples a group of G rollouts for the same (q, s^{poll}). Assuming these rollouts are approximately independent, the probability that the group contains *at least one* successful recovery is

$$\Pr(\text{any success in group}) \;=\; 1 - (1 - p_\theta)^G.$$

In the rare-recovery regime where $Gp_\theta \ll 1$ (equivalently $p_\theta \ll 1/G$), this simplifies to the intuitive approximation

$$1 - (1 - p_\theta)^G \;\approx\; Gp_\theta,$$

meaning the chance of seeing *any* success in a group grows only linearly with both group size G and recovery rate p_θ.

When p_θ is tiny (empirically typical under pollution), most groups contain no successful trajectories, so the terminal rewards are nearly identical across the group (often all zeros) and group-relative normalization produces uninformative advantages. As a result, the GRPO update provides little directed pressure toward recovery and is dominated by noise.

A.2 Limitations of Repair Guidance with a Teacher Model

A straightforward approach is to use a strong external teacher policy π_{T} to generate a repair snippet $w_{\text{T}}^{\text{fix}} \sim \pi_{\text{T}}(\cdot \mid p)$ from a diagnosis prompt p that exposes the inconsistency (e.g., including both w^{clean} and w^{poll}), and then behavior-clone this snippet under the *deployment* conditioning s^{poll}:

$$J^{\text{T}}_{\text{guide}}(\theta) = \mathbb{E}\big[\log \pi_\theta(w_{\text{T}}^{\text{fix}} \mid s^{\text{poll}})\big].$$

The resulting agent objective is

$$J_{\text{agent}}(\theta) = J^{\text{rec}}_{\text{GRPO}}(\theta) + \lambda\, J^{\text{T}}_{\text{guide}}(\theta).$$

Empirically, teacher fixes can be *semantically correct yet inefficient as learning signals* because they often lie in low-probability regions of the student policy. Concretely, under the polluted steer,

$$\pi_\theta(w_\theta^{\text{fix}} \mid s^{\text{poll}}) \gg \pi_\theta(w_{\text{T}}^{\text{fix}} \mid s^{\text{poll}})$$
$$\text{for typical on-policy continuations } w_\theta^{\text{fix}},$$

so the update must push probability mass into regions that the current policy rarely visits. Moreover, under an explicit KL penalty to a reference policy, increasing the likelihood of a repair $w_{\text{T}}^{\text{fix}}$ with $\pi_\theta(w_{\text{T}}^{\text{fix}} \mid s^{\text{poll}}) \approx 0$ requires a large deviation from the reference, so the KL term directly suppresses the update and slows learning.

A.3 Why In-Distribution Guidance Yields Larger First-Order Gains

Setup. Fix a polluted steer (deployment conditioning) s^{poll}. Let the model generate a full continuation $o = (o_1, \ldots, o_T)$ (e.g., repair text plus the remaining reasoning and final answer) according to an autoregressive policy $\pi_\theta(o \mid s^{\text{poll}}) = \prod_{t=1}^{T} \pi_\theta(o_t \mid s^{\text{poll}}, o_{<t})$. Let $r(o) \in [0, 1]$ denote the terminal verifiable reward (e.g., correctness indicator). Define the (conditional) deployment objective

$$J(\theta \mid s^{\text{poll}}) \;:=\; \mathbb{E}_{o \sim \pi_\theta(\cdot|s^{\text{poll}})}\big[r(o)\big]. \tag{2}$$

(An unconditional objective that averages over $(q, a^\star)$ and corresponding s^{poll} follows by linearity of expectation.)

Guidance update. Consider an imitation-style guidance term that increases the likelihood of a specific repair snippet w^{fix} under the *deployment* conditioning s^{poll}:

$$L_{\text{guide}}(\theta) \;:=\; \omega \,\log \pi_\theta\big(w^{\text{fix}} \mid s^{\text{poll}}\big), \qquad \omega > 0. \tag{3}$$

A gradient-ascent step with step size $\eta > 0$ is

$$\theta^+ \;=\; \theta + \eta\, g, \qquad g \;:=\; \nabla_\theta L_{\text{guide}}(\theta) = \omega\, \nabla_\theta \log \pi_\theta\big(w^{\text{fix}} \mid s^{\text{poll}}\big). \tag{4}$$

Goal. We prove that the *first-order* improvement in $J(\theta \mid s^{\text{poll}})$ due to the guidance step contains a dominant term that scales linearly with $\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})$. In particular, under mild near-orthogonality assumptions (standard in high-dimensional score-function analyses), the leading contribution is proportional to $\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\|\nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\|_2^2$, matching the scaling used in §3.3.

A.4 Main decomposition

Lemma A.1 (First-order gain decomposition)**.** *Let θ^+ be defined by* (4)*. Then a first-order Taylor expansion gives*

$$\Delta J \;:=\; J(\theta^+ \mid s^{\text{poll}}) - J(\theta \mid s^{\text{poll}}) \;\approx\; \eta\, \nabla_\theta J(\theta \mid s^{\text{poll}})^\top g. \tag{5}$$

Moreover, writing $\pi_\theta(\cdot)$ as shorthand for $\pi_\theta(\cdot \mid s^{\text{poll}})$,

$$\nabla_\theta J(\theta \mid s^{\text{poll}}) = \mathbb{E}_{o\sim\pi_\theta}\Big[r(o)\, \nabla_\theta \log \pi_\theta(o)\Big], \tag{6}$$

$$\nabla_\theta J(\theta \mid s^{\text{poll}})^\top g = \omega\, \mathbb{E}_{o\sim\pi_\theta}\Big[r(o)\, \big\langle \nabla_\theta \log \pi_\theta(o),\; \nabla_\theta \log \pi_\theta(w^{\text{fix}})\big\rangle\Big]. \tag{7}$$

Finally, letting Ω_{fix} denote the set of continuations whose prefix equals w^{fix},

$$\Omega_{\text{fix}} := \{o :\; o_{1:|w^{\text{fix}}|} = w^{\text{fix}}\},$$

the inner product decomposes exactly as

$$\nabla_\theta J(\theta \mid s^{\text{poll}})^\top g = \omega \underbrace{\sum_{o\in\Omega_{\text{fix}}} \pi_\theta(o)\, r(o)\, \big\langle \nabla_\theta \log \pi_\theta(o),\; \nabla_\theta \log \pi_\theta(w^{\text{fix}})\big\rangle}_{T_1} + \tag{8}$$

$$\omega \underbrace{\sum_{o\notin\Omega_{\text{fix}}} \pi_\theta(o)\, r(o)\, \big\langle \nabla_\theta \log \pi_\theta(o),\; \nabla_\theta \log \pi_\theta(w^{\text{fix}})\big\rangle}_{T_2}. \tag{9}$$

Proof. Equation (5) is the first-order Taylor approximation of $J(\cdot \mid s^{\text{poll}})$ around θ. For (6), apply the score-function identity:

$$\nabla_\theta J(\theta \mid s^{\text{poll}}) = \nabla_\theta \sum_o \pi_\theta(o)\, r(o) = \sum_o \pi_\theta(o)\, r(o)\, \nabla_\theta \log \pi_\theta(o) = \mathbb{E}_{o\sim\pi_\theta}[r(o)\nabla_\theta \log \pi_\theta(o)].$$

Substituting $g = \omega\nabla_\theta \log \pi_\theta(w^{\text{fix}})$ yields (7). The split (9) is obtained by partitioning the support of o into Ω_{fix} and its complement. □

A.5 Isolating the deployment-likelihood scaling

We now show that T_1 contains a term proportional to $\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})$. For any $o \in \Omega_{\text{fix}}$, write $o = w^{\text{fix}} \circ z$, where z is the suffix after the snippet. Then

$$\pi_\theta(o \mid s^{\text{poll}}) = \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\; \pi_\theta(z \mid s^{\text{poll}}, w^{\text{fix}}), \tag{10}$$

$$\log \pi_\theta(o \mid s^{\text{poll}}) = \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}}) + \log \pi_\theta(z \mid s^{\text{poll}}, w^{\text{fix}}), \tag{11}$$

$$\nabla_\theta \log \pi_\theta(o \mid s^{\text{poll}}) = \nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}}) + \nabla_\theta \log \pi_\theta(z \mid s^{\text{poll}}, w^{\text{fix}}). \tag{12}$$

Plugging (10)–(12) into T_1 from (9) gives

$$T_1 = \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\, \mathbb{E}_{z\sim\pi_\theta(\cdot\mid s^{\text{poll}},w^{\text{fix}})}\Big[r(w^{\text{fix}}\circ z)\big\langle \nabla_\theta \log \pi_\theta(w^{\text{fix}}) + \nabla_\theta \log \pi_\theta(z \mid s^{\text{poll}}, w^{\text{fix}}),\ \nabla_\theta \log \pi_\theta(w^{\text{fix}})\big\rangle\Big]$$
$$= \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\, \mathbb{E}_{z\sim\pi_\theta(\cdot\mid s^{\text{poll}},w^{\text{fix}})}\Big[r(w^{\text{fix}}\circ z)\Big(\big\|\nabla_\theta \log \pi_\theta(w^{\text{fix}})\big\|_2^2 + \big\langle \nabla_\theta \log \pi_\theta(z \mid s^{\text{poll}}, w^{\text{fix}}),\ \nabla_\theta \log \pi_\theta(w^{\text{fix}})\big\rangle\Big)\Big]. \tag{13}$$

Define the (conditional) success probability after emitting the repair snippet:

$$Q_\theta(s^{\text{poll}}, w^{\text{fix}}) \;:=\; \mathbb{E}_{z\sim\pi_\theta(\cdot\mid s^{\text{poll}},w^{\text{fix}})}\big[r(w^{\text{fix}}\circ z)\big] \in [0,1]. \tag{14}$$

Then (13) can be written as

$$T_1 = \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\, Q_\theta(s^{\text{poll}}, w^{\text{fix}})\, \big\|\nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\|_2^2 + \tag{15}$$

$$\underbrace{\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\, \mathbb{E}_z\Big[r(w^{\text{fix}}\circ z)\,\big\langle \nabla_\theta \log \pi_\theta(z \mid s^{\text{poll}}, w^{\text{fix}}),\ \nabla_\theta \log \pi_\theta(w^{\text{fix}})\big\rangle\Big]}_{E_{\text{suffix}}}. \tag{16}$$

Near-orthogonality assumption (standard). The remaining terms E_{suffix} and T_2 are cross inner-products between score functions of *different* token sequences (or different segments of the same sequence). In very high-dimensional parameter spaces, these cross terms are typically small unless the sequences are extremely similar; see, e.g., the orthogonality-based analysis in Appendix A of ExPO. Formally, we assume the following bounded-cross-term condition.

Assumption A.2 (Small cross score inner-products)**.** There exist nonnegative constants ϵ_{suffix} and ϵ_{off} such that

$$\Big|\mathbb{E}_{z\sim\pi_\theta(\cdot\mid s^{\text{poll}},w^{\text{fix}})}\Big[r(w^{\text{fix}}\circ z)\,\big\langle \nabla_\theta \log \pi_\theta(z \mid s^{\text{poll}}, w^{\text{fix}}),\ \nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\rangle\Big]\Big| \le \epsilon_{\text{suffix}}, \tag{17}$$

$$\Big|\sum_{o\notin\Omega_{\text{fix}}} \pi_\theta(o \mid s^{\text{poll}})\, r(o)\,\big\langle \nabla_\theta \log \pi_\theta(o \mid s^{\text{poll}}),\ \nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\rangle\Big| \le \epsilon_{\text{off}}. \tag{18}$$

Proposition A.3 (Leading scaling with $\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})$)**.** *Under Assumption A.2, the first-order improvement satisfies*

$$\Delta J \;\approx\; \eta\,\omega\,\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\, Q_\theta(s^{\text{poll}}, w^{\text{fix}})\, \big\|\nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\|_2^2 \;+\; \eta\,\omega\,\xi, \tag{19}$$

where the residual ξ is bounded as $|\xi| \le \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\epsilon_{\text{suffix}} + \epsilon_{\text{off}}$. Consequently, whenever

$$\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}}) Q_\theta(s^{\text{poll}}, w^{\text{fix}}) \big\|\nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\|_2^2 \;\gg\; \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\epsilon_{\text{suffix}} + \epsilon_{\text{off}}, \tag{20}$$

the first term in (19) *dominates, so the improvement is approximately proportional to $\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})$.*

Proof. Combine Lemma A.1 with (16):

$$\nabla_\theta J(\theta \mid s^{\text{poll}})^\top g = \omega\,(T_1 + T_2) = \omega\Big(\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}}) Q_\theta \|\nabla_\theta \log \pi_\theta(w^{\text{fix}})\|_2^2 + E_{\text{suffix}} + T_2\Big).$$

Apply Assumption A.2 to bound E_{suffix} and T_2, and multiply by η per (5). □

Recovering the simplified expression in the dominance regime. If w^{fix} is a near-certain repair under the current policy (i.e., $Q_\theta(s^{\text{poll}}, w^{\text{fix}}) \approx 1$), and the dominance condition (20) holds (so the residual is lower-order), Proposition A.3 reduces to

$$\Delta J \;\approx\; \eta\,\omega\,\pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\, \big\|\nabla_\theta \log \pi_\theta(w^{\text{fix}} \mid s^{\text{poll}})\big\|_2^2, \tag{21}$$

which is the scaling statement referenced in §3.3.

A.6 Corollary: why higher-likelihood repairs yield larger gains

Corollary A.4 (Preference for higher-likelihood repairs in the dominance regime)**.** *Let $w^{(1)}$ and $w^{(2)}$ be two candidate repair snippets under the same s^{poll}. Assume they have comparable weights and score norms, and comparable downstream success probabilities:*

$$\omega^{(1)} \approx \omega^{(2)}, \qquad \left\|\nabla_\theta \log \pi_\theta(w^{(1)} \mid s^{\mathrm{poll}})\right\|_2^2 \approx \left\|\nabla_\theta \log \pi_\theta(w^{(2)} \mid s^{\mathrm{poll}})\right\|_2^2, \qquad Q_\theta(s^{\mathrm{poll}}, w^{(1)}) \approx Q_\theta(s^{\mathrm{poll}}, w^{(2)}),$$

and the analogous dominance condition from (20) *holds for both snippets. Then the ratio of first-order gains satisfies*

$$\frac{\Delta J(w^{(1)})}{\Delta J(w^{(2)})} \;\approx\; \frac{\pi_\theta(w^{(1)} \mid s^{\mathrm{poll}})}{\pi_\theta(w^{(2)} \mid s^{\mathrm{poll}})}. \tag{22}$$

In particular, if $w^{(1)}$ is substantially more likely under the current policy than $w^{(2)}$, then it yields a substantially larger expected first-order improvement for the same-sized update.

Proof. Apply Proposition A.3 to each snippet and cancel the approximately equal factors. □

A.7 Why Few Successes Yield Noisy GRPO Updates

We formalize the claim that when successful trajectories are rare, GRPO's outcome-only update is *noisy* because the success-conditioned contribution is estimated from very few samples (often one), yielding a low signal-to-noise ratio.

Setup. Fix a conditioning context s (e.g., a polluted steer s_{poll} in Sec. 4.4). GRPO samples a group of G rollouts $\{\tau_i\}_{i=1}^G$ from a behavior policy $\pi_{\theta_{\mathrm{old}}}$. Each rollout τ_i has token sequence $o_i = (o_{i,1}, \ldots, o_{i,|o_i|})$ and terminal reward $R_i \in \{0, 1\}$ (verifiable correctness). GRPO forms the group-normalized reward

$$\tilde{R}_i \;:=\; \frac{R_i - \overline{R}}{s_R + \varepsilon}, \qquad \overline{R} := \frac{1}{G}\sum_{j=1}^{G} R_j, \qquad s_R := \sqrt{\frac{1}{G}\sum_{j=1}^{G}(R_j - \overline{R})^2}, \tag{23}$$

and assigns $\hat{A}_{i,t} = \tilde{R}_i$ for all tokens t (Sec. 3.2).

To isolate the variance mechanism, we analyze the *unclipped* surrogate (equivalently, the first-order/small-step regime where $\rho_{i,t}(\theta) \approx 1$ and the $\min/\mathrm{clip}$ does not activate). Define the per-trajectory score-gradient average

$$S_i(\theta) \;:=\; \frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \nabla_\theta \log \pi_\theta(o_{i,t} \mid s, o_{i,<t}), \tag{24}$$

so that the corresponding (unclipped) GRPO gradient estimator can be written as

$$\hat{g}(\theta) \;:=\; \frac{1}{G}\sum_{i=1}^{G} \tilde{R}_i\, S_i(\theta). \tag{25}$$

(Clipping and KL regularization can only *downweight* some terms and thus do not remove the scarcity-induced noise; Remark A.8 discusses this.)

Closed form of group-normalized rewards for binary outcomes. Let $K := \sum_{i=1}^G R_i$ be the number of successes in the group.

Lemma A.5 (Binary-reward GRPO advantages)**.** *Condition on $K = k$ with $1 \le k \le G-1$. Then $\overline{R} = k/G$ and $s_R = \sqrt{k(G-k)}/G$. Consequently,*

$$\tilde{R}_i = \begin{cases} a_k := \sqrt{\frac{G-k}{k}}, & \text{if } R_i = 1, \\ b_k := -\sqrt{\frac{k}{G-k}}, & \text{if } R_i = 0, \end{cases} \tag{26}$$

(up to the negligible ε in (23)*). If $k = 0$ (all failures) or $k = G$ (all successes), then $s_R = 0$ and $\tilde{R}_i \approx 0$ for all i due to ε, so $\hat{g}(\theta) \approx 0$.*

Proof. When $K = k$, the reward multiset contains k ones and $G - k$ zeros, so $\overline{R} = k/G$. The variance is

$$s_R^2 = \frac{1}{G}\Big(k(1 - k/G)^2 + (G - k)(0 - k/G)^2\Big) = \frac{k(G-k)}{G^2},$$

hence $s_R = \sqrt{k(G-k)}/G$. Plugging into (23) yields (26). For $k \in \{0, G\}$ we have $s_R = 0$ and $\tilde{R}_i = (R_i - \overline{R})/\varepsilon \approx 0$. □

Decomposition as a difference of two sample means. Let $S^{(1)} := \{i : R_i = 1\}$ and $S^{(0)} := \{i : R_i = 0\}$. Define the within-group sample means

$$\bar{S}_1 := \frac{1}{k}\sum_{i \in S^{(1)}} S_i(\theta), \qquad \bar{S}_0 := \frac{1}{G-k}\sum_{i \in S^{(0)}} S_i(\theta). \tag{27}$$

Lemma A.6 (Gradient estimator given $K = k$)**.** *Condition on $K = k$ with $1 \le k \le G - 1$. Then*

$$\hat{g}(\theta) = c_k\big(\bar{S}_1 - \bar{S}_0\big), \qquad c_k := \frac{\sqrt{k(G-k)}}{G}. \tag{28}$$

In particular, when $k = 1$, the update depends on a single *success trajectory via $\bar{S}_1 = S_{i^\star}$.*

Proof. By Lemma A.5, $\tilde{R}_i = a_k$ on successes and $\tilde{R}_i = b_k$ on failures. Thus

$$\hat{g}(\theta) = \frac{1}{G}\Big(a_k \sum_{i \in S^{(1)}} S_i(\theta) + b_k \sum_{i \in S^{(0)}} S_i(\theta)\Big) = \frac{1}{G}\Big(a_k k\, \bar{S}_1 + b_k (G-k)\, \bar{S}_0\Big).$$

Using $a_k k = \sqrt{k(G-k)}$ and $b_k(G-k) = -\sqrt{k(G-k)}$ yields (28). □

Variance and signal-to-noise. Assume that conditional on the reward, the score-gradients have finite second moments:

$$\mu_r := \mathbb{E}[S_i(\theta) \mid R_i = r], \qquad \Sigma_r := \mathrm{Cov}(S_i(\theta) \mid R_i = r), \qquad r \in \{0, 1\}. \tag{29}$$

Assume also that $\{S_i(\theta)\}_{i=1}^G$ are conditionally independent given $\{R_i\}_{i=1}^G$ (standard for independent rollouts).

Proposition A.7 (Conditional covariance and rare-success SNR characterization)**.** *Condition on $K = k$ with $1 \le k \le G - 1$. Then*

$$\mathbb{E}\big[\hat{g}(\theta) \mid K = k\big] = c_k\big(\mu_1 - \mu_0\big), \tag{30}$$

$$\mathrm{Cov}\big(\hat{g}(\theta) \mid K = k\big) = c_k^2\left(\frac{\Sigma_1}{k} + \frac{\Sigma_0}{G-k}\right). \tag{31}$$

Moreover, defining a (squared) signal-to-noise ratio using Frobenius/trace variance,

$$\mathrm{SNR}^2(k) \;:=\; \frac{\big\|\mathbb{E}[\hat{g}(\theta) \mid K = k]\big\|_2^2}{\mathbb{E}\big[\|\hat{g}(\theta) - \mathbb{E}[\hat{g}(\theta) \mid K = k]\|_2^2 \mid K = k\big]}, \tag{32}$$

we obtain the closed form

$$\mathrm{SNR}^2(k) = \frac{\|\mu_1 - \mu_0\|_2^2}{\mathrm{tr}(\Sigma_1)/k + \mathrm{tr}(\Sigma_0)/(G-k)}. \tag{33}$$

In particular, when $k = 1$ the success term $\mathrm{tr}(\Sigma_1)/k$ receives no averaging reduction*, reflecting that the success-conditioned contribution is estimated from a single trajectory. More generally,* (33) *makes clear why rare-success batches have low SNR, while global monotonicity in k need not hold without stronger assumptions on Σ_0 and Σ_1.*

Proof. From Lemma A.6, $\hat{g}(\theta) = c_k(\bar{S}_1 - \bar{S}_0)$. By conditional independence and (29), $\mathbb{E}[\bar{S}_1 \mid K = k] = \mu_1$ and $\mathbb{E}[\bar{S}_0 \mid K = k] = \mu_0$, yielding (30). Similarly, $\mathrm{Cov}(\bar{S}_1 \mid K = k) = \Sigma_1/k$ and $\mathrm{Cov}(\bar{S}_0 \mid K = k) = \Sigma_0/(G-k)$, and the cross-covariance is zero, so $\mathrm{Cov}(\bar{S}_1 - \bar{S}_0 \mid K = k) = \Sigma_1/k + \Sigma_0/(G-k)$. Multiplying by c_k^2 gives (31). Finally,

$$\mathbb{E}\big[\|\hat{g} - \mathbb{E}[\hat{g} \mid K = k]\|_2^2 \mid K = k\big] = \mathrm{tr}\big(\mathrm{Cov}(\hat{g} \mid K = k)\big) = c_k^2\left(\frac{\mathrm{tr}(\Sigma_1)}{k} + \frac{\mathrm{tr}(\Sigma_0)}{G-k}\right),$$

and $\|\mathbb{E}[\hat{g} \mid K = k]\|_2^2 = c_k^2\|\mu_1 - \mu_0\|_2^2$. Canceling c_k^2 yields (33). □

Interpretation ("few successes ⇒ noisy"). Equation (31) shows that the success-side uncertainty scales as Σ_1/k. Thus when k is small (especially $k = 1$), the success-conditioned mean gradient μ_1 is estimated with high variance, and (33) shows the resulting GRPO update has low SNR. Increasing k from this rare-success regime reduces the success-side variance term Σ_1/k, even though the full SNR expression need not be globally monotone in k. When $k = 0$, Lemma A.5 implies $\hat{g}(\theta) \approx 0$, yielding an uninformative update.

Remark A.8 (Clipping/KL do not eliminate scarcity noise). The analysis above uses the unclipped surrogate for clarity. In GRPO, clipping replaces each token term by a quantity whose magnitude is *upper-bounded* by the unclipped term. Therefore, clipping/KL can further *reduce* the signal magnitude in rare-success batches, but cannot increase the number of successes k nor remove the $1/k$ sampling noise in estimating μ_1.

A.8 Additional Related Works

Reinforcement fine-tuning (RFT) has become a dominant paradigm for scaling reasoning in large language models. Early systems largely followed PPO-style RLHF with learned reward models [30, 55]. More recently, RLVR and critic-free objectives such as GRPO [34] showed that outcome-based supervision—often only final-answer correctness—can elicit strong long chain-of-thought reasoning while simplifying the RL stack. This has triggered a wave of work refining GRPO to improve learning efficiency[10, 58], stability[31], and bias properties. Related work in offline RL has explored sequence-modeling approaches that improve trajectory stitching by using critic guidance to compose useful behaviors across trajectories [43]. For instance, DAPO [49] adjusts group-relative advantage estimation and relaxes KL constraints to enable faster departure from the base model; GMPO [56] replaces GRPO's arithmetic aggregation with a geometric mean to improve stability; DR.GRPO [25] analyzes sources of bias and proposes variants that remove length and standard-deviation normalization to better match an unbiased policy-gradient objective; and GSPO [57] moves from token-level to sequence-level likelihood ratios and clipping, aligning optimization more directly with sequence-level outcomes. Collectively, these methods sharpen the optimization of *clean* reasoning performance: given a reliable prompt and a well-specified objective, they aim to make outcome-RL updates more effective and less destabilizing. In deployed settings, this assumption of a trustworthy conditioning context often fails: users make mistakes, partial solutions may contain errors, and intermediate context can be noisy, misleading, or distribution-shifted[47]. As a result, a model can be a strong *solo* reasoner yet remain brittle under fallible context, where the key capability is not merely to "solve," but to **detect issues, repair the context, and recover the correct answer**.

Robustness can be characterized not only behaviorally, but also through the structure and stability of learned systems. Prior work has analyzed neural-network robustness using graph curvature [38] and subsequently exploited graph-curvature information for post-training pruning [40], providing a structural perspective on identifying important neural connections while adapting trained models. Robustness and reliability are likewise central in safety-critical autonomous systems[46], including risk-aware road–car–pedestrian modeling [39, 27], semantic-graph-based vehicle relocalization [37], efficient visual object proposal generation [15], adaptive uncertainty estimation for multi-object tracking [14], and multi-sensor-fusion-based motion state estimation [13]. These works address different forms of structural, perceptual, and environmental uncertainty. GASP complements these directions by considering uncertainty and corruption in the conditioning context of reasoning models, and by explicitly training the model to detect, repair, and recover from such failures.

A.9 Full Benchmarks

Model	Method	Recoverability	Diagnosability		Reliability		Avg
		GSM8K	MR-GSM8K	MHPP	GSM8K	CQA	Avg
R1-Distill-Qwen-1.5B	Initial ckpt	9.8	4.5	2.9	33.3	37.1	17.5
	ASP	26.3 ± 1.8	11.2 ± 0.7	5.2 ± 0.4	44.4 ± 1.9	45.3 ± 1.8	26.5
	GASP-P-FIXED	30.8 ± 1.6	13.0 ± 1.0	6.0 ± 0.5	52.0 ± 2.0	53.5 ± 1.9	31.1
	GASP-T	35.5 ± 1.5	15.1 ± 1.2	7.0 ± 0.6	59.9 ± 2.1	61.2 ± 2.0	35.7
	GASP-P-GPT-5	74.2 ± 1.3	40.8 ± 1.9	15.4 ± 1.1	63.9 ± 2.2	64.0 ± 2.0	51.7
	GASP	78.5 ± 1.2	43.7 ± 2.1	16.9 ± 1.2	64.8 ± 2.3	64.7 ± 2.1	**53.7**
DeepScaleR-1.5B	Initial ckpt	10.2	6.2	3.3	39.2	40.8	19.9
	ASP	19.6 ± 1.5	12.4 ± 0.8	7.1 ± 0.6	45.4 ± 2.0	48.2 ± 2.1	26.5
	GASP-P-FIXED	23.1 ± 1.2	14.6 ± 1.1	8.4 ± 0.7	53.2 ± 2.1	56.5 ± 2.0	31.2
	GASP-T	26.5 ± 1.3	16.7 ± 1.4	9.6 ± 0.9	61.3 ± 2.2	65.1 ± 2.2	35.8
	GASP-P-GPT-5	75.0 ± 1.2	43.2 ± 2.0	17.0 ± 1.3	66.6 ± 2.3	67.5 ± 2.1	53.9
	GASP	79.7 ± 1.1	46.8 ± 2.3	18.6 ± 1.4	67.3 ± 2.4	68.3 ± 2.3	**56.1**
Qwen3-4B	Initial ckpt	8.9	10.7	5.1	46.5	48.7	24.0
	ASP	28.5 ± 2.0	19.2 ± 1.3	9.6 ± 0.8	56.7 ± 2.2	68.9 ± 2.5	36.6
	GASP-P-FIXED	33.2 ± 1.6	22.4 ± 1.4	11.2 ± 1.0	66.8 ± 2.1	73.0 ± 2.0	41.3
	GASP-T	38.5 ± 1.5	25.9 ± 1.7	13.0 ± 1.1	76.5 ± 2.0	78.6 ± 2.3	46.5
	GASP-P-GPT-5	78.0 ± 1.3	56.0 ± 2.2	27.5 ± 1.7	86.3 ± 1.9	79.2 ± 2.1	65.4
	GASP	81.2 ± 0.9	59.3 ± 2.5	29.8 ± 1.9	88.7 ± 1.8	79.7 ± 2.4	**67.7**
Qwen3-8B (lora)	Initial ckpt	7.2	16.2	8.2	53.6	69.9	31.0
	ASP	34.4 ± 2.3	28.7 ± 1.9	14.4 ± 1.2	69.9 ± 2.6	85.4 ± 2.1	46.6
	GASP-P-FIXED	40.2 ± 1.7	33.0 ± 1.6	16.6 ± 1.3	79.8 ± 2.0	90.5 ± 1.6	52.0
	GASP-T	46.4 ± 1.6	38.7 ± 2.0	19.4 ± 1.7	89.0 ± 1.9	94.0 ± 1.5	57.5
	GASP-P-GPT-5	83.0 ± 0.8	68.5 ± 2.0	50.5 ± 2.4	89.5 ± 1.6	94.5 ± 1.3	77.2
	GASP	85.7 ± 0.7	72.7 ± 2.1	54.1 ± 2.8	89.8 ± 1.5	94.8 ± 1.2	**79.4**

Table 7: **Main robustness results (pass@1).** Mean ± std over 3 training runs for ASP/GASP-P-FIXED/GASP-T/GASP-P-GPT5/GASP; **Initial ckpt** is a single evaluation. **Recoverability**: GSM8K accuracy under corrupted context (held-out GPT-5 polluter). **Diagnosability**: **ACCreason** on MR-GSM8K and MHPP [52, 53]. **Reliability**: accuracy on perturbed GSM8K and CommonsenseQA from RUPBench [45]. **Avg**: unweighted mean of the five metrics. GASP-T uses teacher-generated (GPT-5) repair snippets in place of self-generated repair snippets. GASP-P-FIXED fixes the polluter (no adversarial polluter updates) during training. GASP-P-GPT-5 replaces the learned polluter with GPT-5 during training.

B Prompt Templates and Role Conditioning

- **Agent prompt (system prompt only).**

```
[SYSTEM]
    A conversation between User and Assistant. The user asks a question, and the
    Assistant solves it.
    The assistant first thinks about the reasoning process in the mind and then
    provides the user with the answer.
    The reasoning process and answer are enclosed within <think> </think> and <
    answer> </answer> tags, respectively, i.e.,
    <think> reasoning process here </think><answer> answer here </answer>
```

- **Polluter prompt.**

```
[SYSTEM]
You are a helpful assistant.

[USER]
You will be given:
  - a math problem,
  - the student's reasoning prefix (what they have already written),
  - and a SHORT CONTINUATION TEXT that the student originally wrote.

Your job is to EDIT ONLY that short continuation text so that:
  - it stays natural, coherent, and in the same style and level of detail,
  - it contains at least one explicit mathematical statement (numbers / formula /
    relation),
```

```
  - and at least one of those mathematical statements becomes objectively
    incorrect.

You must treat the original continuation as a piece of text to be edited IN PLACE,
     not as a topic to write a new paragraph about.

EDIT PRIORITY:
  1) If the original continuation already contains any numbers, numeric
    expressions, indices,
     or concrete mathematical facts, you MUST introduce the error by MODIFYING
    those existing
     numbers or facts (for example, change 21st to 25th; change 1, 10, 19 to 1,
    16, 29;
     change 10 - 1 = 9 to 10 - 1 = 8; change (n-1)d to (n+1)d).
     In this case you are NOT allowed to introduce entirely new sentences, and you
     should avoid
     inventing new numbers that did not appear before, except by locally changing
    some digits.
  2) Only if the original continuation contains NO numerical or mathematical
    content at all
     may you add a new mathematical statement and make that statement false.

CRITICAL CONSTRAINT ON EQUATIONS:
  - If the text contains any equations of the form 'a - b = c', 'a + b = c', or 'a
     * b = c',
    the rewritten text MUST contain at least one such equation that is
    mathematically FALSE.
  - You must NOT leave all such equations mathematically correct after you edit
    the numbers.
  - For example, if you change '10 - 1 = 9', you must ensure the new equality is
    wrong, such as
    '16 - 1 = 9' or '10 - 1 = 8'.

OUTPUT RULES:
  - The output must be a lightly edited version of the original continuation, not
    a new paragraph.
  - Keep the same number of sentences if possible.
  - Do NOT state any final numeric answer.
  - Do NOT say that anything is wrong, misleading, or experimental.
  - Output ONLY the edited continuation, wrapped exactly in <polluted>...</
    polluted> tags.
  - No extra text before or after the <polluted> block.

[PROBLEM]
{q}

[STUDENT REASONING PREFIX]
{prefix}

[ORIGINAL SHORT CONTINUATION TO EDIT]
{continuation}

[YOUR OUTPUT]
(Only the edited continuation, wrapped in <polluted>...</polluted>.)
```

- **Teacher and Self-guidance generation prompt.** Including which context is shown (e.g., $(q, c_{0:\alpha})$ and both $(w^{\text{clean}}, w^{\text{poll}})$) and the instruction to produce a brief appendable repair.

```
    [SYSTEM]
    You are a helpful reasoning assistant. Solve the user's problem.

    [USER]
    You are helping generate a short "repair continuation" for robustness training.
```

```
You will be given:
(1) a problem q
(2) the student/agent's reasoning prefix c_0:alpha
(3) two alternative next windows:
    - CLEAN window w_clean (the correct continuation)
    - POLLUTED window w_poll (a locally coherent but potentially wrong/
misleading continuation)

Your job:
Write a SHORT continuation that would come *immediately after the POLLUTED
window* and "repairs" the trajectory--i.e., it identifies/corrects the key
mistake implied by w_poll and steers back toward the correct solution path.

Critical constraints (important):
- Output ONLY the repair continuation text (no headings, no bullet points, no
meta commentary).
- Do NOT mention the existence of w_clean or w_poll, and do NOT compare them
explicitly.
- The continuation must be natural as if the model only saw: (q, c_0:alpha,
w_poll).
- Keep it short (aim: ~1-4 sentences). Do not re-solve from scratch.
- Prefer minimal intervention: point out the inconsistency, fix the specific
step, then continue one step forward.
- Maintain the same tone/style/level of detail as the prefix.
- Do not output the final numeric answer unless the polluted context already
forces you to (otherwise stop before the final answer).

Now generate the repair continuation.

[PROBLEM q]
{q}

[REASONING PREFIX c_0:alpha]
{prefix}

[CLEAN WINDOW w_clean -- for your reference only]
{w_clean}

[POLLUTED WINDOW w_poll -- this is what the agent saw at deployment]
{w_poll}

[YOUR OUTPUT: repair continuation that follows immediately after w_poll]
```

C Diagnosability Evaluation Details

- Evaluation prompt:

```
[SYSTEM]
Act as a grade school math teacher and score the following problem solution.

[USER]
Question:
{data['question']}
Student Solution:
{sol_steps}
Your task involves three parts:
1. **Step-by-step Evaluation:** Go through the student solution carefully and
    identify
key errors and potential misunderstandings that led to the incorrect solution.
2. **Final Judgement:** Provide an overall judgement on the correctness of the
student's solution.
3. **First Error Step:** If the solution is incorrect, generate the step number
    where
```

```
the first error occurs, otherwise generate N/A here
4. **Error Analysis:** If the solution is incorrect, analyse the cause and reasons
     for
the first error step, otherwise generate N/A here
Here's the format I want:
Step-by-step Evaluation: [Provide a step by step examination of the student
    solution and
identify key errors and misunderstandings here.]
Final Judgement: [Insert only **correct** or **wrong** here]
First Error Step: [Insert either N/A or the step number where the first error
    occurs]
Error Analysis: [Insert either N/A or the analysis of error in the first error
    among
solution steps]
Please follow this format without any additional introductory or concluding
    statements.
```

D Self-Revision Evaluation Details

- valuation prompt:

```
[SYSTEM]
    A conversation between User and Assistant. The user asks a question, and the
    Assistant solves it.
    The assistant first thinks about the reasoning process in the mind and then
    provides the user with the answer.
    The reasoning process and answer are enclosed within <think> </think> and <
    answer> </answer> tags, respectively, i.e.,
    <think> reasoning process here </think><answer> answer here </answer>

[USER]
Question:
{QUESTION}

Your previous solution (incorrect):
{INCORRECT_SOLUTION}

Task:
1) Identify the key mistake(s) in the previous solution.
2) Produce a corrected solution.
3) Provide the final answer in the form boxed{answer} at the end of your response.
```

E Hyperparameters and Implementation Details

Optimization and precision. Unless otherwise stated, we optimize all models with AdamW [16, 5, 2, 21, 29] (PyTorch implementation) using bfloat16 training and gradient checkpointing[5]. We use a cosine learning-rate schedule with warmup ratio 0.1, peak learning rate 3×10^{-6}, weight decay 0 [4], and gradient-norm clipping at 1.0. All runs use a set of fixed random seed (42, 52, 62).

Parameter-efficient fine-tuning. For the 8B backbone (Qwen3-8B), we use LoRA-based fine-tuning to fit training within limited hardware [20, 28]. LoRA adapters are applied to the attention and MLP projection layers with rank $r = 64$ (shared across modules). All base model weights are frozen, and only LoRA parameters are updated during training. This configuration allows stable GRPO self-play training on 4×A100 GPUs without model or tensor parallelism, while preserving sufficient adaptation capacity for robust reasoning and repair behaviors. Under the GASP training setup in §E, the estimated end-to-end wall-clock time for one training run on 4×A100-80GB GPUs is 12.2 hours for R1-Distill-Qwen-1.5B, 12.4 hours for DeepScaleR-1.5B, 21.4 hours for Qwen3-4B, and 32.5 hours for Qwen3-8B with LoRA. Since each reported result averages over three random seeds, the corresponding total wall-clock budget is approximately 29.4, 31.2, 55.8, and 95.1 GPU-wall-clock hours per model family, respectively.

GRPO rollouts. We use Group Relative Policy Optimization (GRPO) with clipping parameter $\epsilon = 0.2$ (Eq. 1). For each update, we sample $G = 64$ trajectories per conditioning context at temperature 0.7 (top-$k = 50$, top-$p = 1.0$). We cap the maximum prompt length to 1024 tokens and the maximum completion length to 1024 tokens. Unless otherwise specified, we use no explicit KL regularization to a reference policy (KL coefficient $\beta = 0$). We use $\lambda = 0.07$ for the guidance term across all models except Qwen3-8B, where we use $\lambda = 0.04$. During training, we linearly anneal λ to 0 over the final steps.

Self-play scheduling. GASP alternates optimization between the agent and polluter roles instantiated by the same backbone model. We use a blocked update schedule: the agent performs 5 optimizer steps while holding the polluter fixed, then the polluter performs 5 optimizer steps while holding the agent fixed, repeating throughout training. When optimizing the polluter, the agent's rollout is treated as a black-box environment outcome (no backpropagation through the agent trajectory).

Implementation. We run training with vLLM-based generation for rollouts and standard distributed training utilities [24, 32] and modified based on TRL [42].

F Representation Shift Measurements

PCA-based representation shift. To quantify representational drift induced by different training procedures, we measure layer-wise shifts in hidden-state representations using principal component analysis (PCA), following prior work [12]. Let X denote a fixed probe set of inputs (e.g., TruthfulQA or CommonsenseQA). For each model state $* \in \{\text{base}, \text{post-training}\}$ and each transformer layer $i \in \{1, \dots, L\}$, we collect the hidden states

$$H_i^{(*)} \in \mathbb{R}^{N \times d},$$

where N aggregates token representations over all inputs in X and d is the hidden dimension. We apply PCA with $n = 2$ components to $H_i^{(*)}$ and compute the mean projection onto each principal axis:

$$m_{i,k}^{(*)} = \text{mean}\Big(\text{proj}_{\text{PC}k}\Big(H_i^{(*)}\Big)\Big), \quad k \in \{1, 2\}.$$

We define the PC1 shift relative to the base model as

$$\Delta m_{i,1}^{(*)} = m_{i,1}^{(*)} - m_{i,1}^{(\text{base})},$$

and form a 2D representation for each layer

$$z_i^{(*)} = \big(\Delta m_{i,1}^{(*)},\ m_{i,2}^{(*)}\big).$$

Each point in the PCA shift plots (e.g., Fig. 5) corresponds to one transformer layer.

To summarize global representational movement, we compute the centroid across layers

$$z^{(*)} = \frac{1}{L} \sum_{i=1}^{L} z_i^{(*)},$$

and report the scalar PCA shift magnitude

$$d^{(*)} = \left\| z^{(*)} - z^{(\text{base})} \right\|_2.$$

Larger $d^{(*)}$ indicates greater global drift of internal representations on the probe distribution.

G Limitation of GASP

A key limitation of this work is that GASP is primarily evaluated in reasoning settings with verifiable final answers, which may restrict its generalizability to broader domains where correctness is harder to automatically verify, such as open-ended reasoning, factuality, or real-world decision-making. The framework also depends on constructing solvable training instances and localized corruptions, so robustness against more diverse, long-range, or adversarially sophisticated failures remains

uncertain. Additionally, adversarial self-play introduces substantial computational overhead due to alternating polluter-agent optimization and large-group GRPO rollouts, which may limit scalability to larger models [6, 3] or broader datasets. Finally, while self-generated repair guidance reduces representational drift relative to teacher guidance, the method's long-term effects on out-of-domain capabilities, safety, and robustness under deployment-scale distribution shifts are not yet fully characterized.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions. The abstract claims GASP improves robustness (diagnosability, recoverability, reliability) across four open-weight models (1.5B–8B) using only verifiable outcome rewards without human labels, and that in-distribution repair guidance addresses early training signal scarcity. These claims are directly supported by the experimental results in Table 1, the navigation analogue analysis in Section 4.3, and the representation drift analysis in Section 4.4. The scope is appropriately bounded — the paper focuses on math reasoning under corrupted chain-of-thought context, and the introduction is explicit that robustness here means detect-and-repair under fallible conditioning, not general adversarial robustness.

   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: We briefly mentioned in the main and discuss it fully in Appendix: Section G.

   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: The paper includes several theoretical results, all of which are handled with reasonable rigor. The main informal claim in Section 3.4 — that higher-likelihood repair targets yield larger first-order policy improvement — is complemented by a formal treatment in Appendix A.3 through A.6. Specifically, Lemma A.1 provides the first-order gain decomposition with a complete proof using the score-function identity, Assumption A.2 explicitly states the near-orthogonality condition required for the dominant scaling term to hold, Proposition A.3 formally establishes the linear scaling with deployment likelihood under that assumption with a complete proof, and Corollary A.4 derives the preference for higher-likelihood repairs. The learning signal scarcity argument from Section 3.3 is formalized in Appendix A.1 and further extended in Appendix A.7, where Lemma A.5 and A.6 provide closed-form binary-reward GRPO advantage expressions with complete proofs, and Proposition A.7 derives the conditional covariance and SNR characterization. All lemmas, propositions, and assumptions are numbered and cross-referenced. The one mild weakness is that Assumption A.2 (small cross score inner-products) is motivated by appeal to prior work rather than verified empirically or proven formally, but this is acknowledged by the authors and is standard practice in score-function analyses.

   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: The paper provides sufficient information to reproduce the main experimental results. Section 4.1 specifies the four base models (DeepSeek-R1-Distill-Qwen-1.5B, DeepScaleR-1.5B, Qwen3-4B, Qwen3-8B), the training data source (7.5k problems from the MATH training split), the solvability filtering criterion (K=4 solutions at temperature 0.7, all correct), and the use of multiple truncation ratios during training. Appendix E provides detailed hyperparameters including optimizer (AdamW, bfloat16), learning rate schedule (cosine with warmup ratio 0.1, peak lr 3×10^{-6}), GRPO clipping parameter (ε=0.2), group size (G=64), temperature (0.7), maximum prompt and completion lengths (1024 tokens each), guidance coefficient (λ=0.07, λ=0.04 for 8B), annealing schedule, and self-play blocked update schedule (5 agent steps alternating with 5 polluter steps). LoRA configuration for the 8B model is specified (rank r=64, applied to attention and MLP projections). Random seeds are disclosed $(42, 52, 62)$. Full prompt templates for the agent, polluter, repair guidance, diagnosability evaluation, and self-revision evaluation are provided in Appendices B, C, and D. Evaluation protocols for all four benchmarks are described in Section 4.2 with references to benchmark-specific protocols.

   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: Code and data used in this paper is part of company property.

   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: The paper clearly specifies datasets, model variants, training splits, evaluation benchmarks, optimizer settings, hyperparameters, rollout configurations, prompt templates, and implementation details in Sections 4 and Appendix, providing sufficient detail to understand and reproduce the experimental setup.

   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: Main experimental results are consistently reported as mean ± standard deviation over three training runs, with shaded confidence regions in learning curves, and the paper explicitly states the source of variability across random seeds.

   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: The paper specifies core hardware details, including use of 4×A100 GPUs for larger models, but exact wall-clock runtime is difficult to report due to distributed shared-cluster execution; nonetheless, the provided hardware and implementation details are sufficient to broadly understand experimental resource requirements.

Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research involves standard model training and evaluation on existing benchmarks without human subjects, sensitive personal data, or ethically concerning data collection practices, and there is no indication of violations of NeurIPS ethical standards in methodology or reporting.

   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: [No]

    Justification: The paper does not explicitly discuss broader societal impacts, likely because the work is primarily a technical contribution focused on improving robustness in reasoning systems, with relatively limited direct societal implications beyond general improvements to model reliability.

    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: [No]

    Justification: The paper does not release new high-risk models, datasets, or generative systems intended for public deployment; it primarily presents a training methodology evaluated on existing open-weight models and standard benchmarks, so specialized responsible-release safeguards are not directly applicable.

    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: [No]

    Justification: While the paper appropriately cites prior models, datasets, and benchmark sources, it does not explicitly document licensing terms, usage restrictions, or asset-specific terms of use for the external models and datasets employed.

    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: [No]

Justification: Although the paper introduces the GASP training framework and associated prompt templates, it does not provide released code, packaged assets, or formal accompanying documentation for newly introduced resources beyond methodological descriptions in the manuscript and appendix.

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: The paper does not involve crowdsourcing, human subjects, or participant-based data collection, so requirements regarding participant instructions, screenshots, and compensation do not apply.

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: The research does not involve human subjects, participant studies, or crowdsourced data collection, so IRB approval, participant risk disclosure, and related human-subject protections are not applicable.

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: The paper explicitly describes LLM usage as a central component of the methodology, including the underlying reasoning models, adversarial self-play roles, training procedures, and evaluation settings, making LLM involvement clearly documented and integral to the research contribution.

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.