

Emergent Backdoors in Agentic Recursive Training

Xuan Wang¹, Yuliang Lu², Kejiang Ye³, Cheng-zhong Xu⁴, Xitong Gao^{3*}

¹Nanyang Technological University, Singapore

²National University of Defense Technology, China

³Shenzhen Institutes of Advanced Technology, CAS

⁴Institute of AI and Brain Sciences, CS Dept., University of Macau

wangxuan21d@gmail.com, xt.gao@siat.ac.cn

Abstract

Agentic recursive training systems increasingly reuse model-generated tasks, proxy feedback, and selected trajectories to refine policies over multiple rounds. This creates feedback loops in which small conditional biases can accumulate across training and alter later selection and updates. We study whether trigger-conditional proxy-feedback biases can produce emergent backdoors in a controlled recursive training loop. A Curriculum Agent proposes tasks, an Executor Agent generates rollouts, and environment feedback is selectively biased on rare markers, affecting reward, selection, and curriculum updates. Under matched biased supervision, Reward Hacking + SFT and Reward Hacking + RLHF reach 31.27% and 53.85% ASR, while the full loop reaches 97.62% at comparable clean accuracy. The effect persists across Qwen3-4B, LLaMA-3-8B, BLIP-2-OPT-2.7B, and Qwen2.5-14B, transfers to GSM8K, BBH-6, and HotpotQA, extends to a tool-augmented setting, and is only partially reduced by post-hoc defenses. These results suggest that feedback and curriculum mechanisms in agentic recursive training should be audited for long-horizon bias amplification.

1 Introduction

Agentic recursive training combines task generation, proxy feedback, and iterative selection (Christiano et al., 2023; Graves et al., 2017; Xia et al., 2025). In this paper, the term refers to a narrow feedback-training loop: outputs from one round shape the curriculum and policy update in the next round. This is not a claim of full recursive self-improvement, which would require the system to modify its own optimizer, architecture, or tool stack. The question we study is whether small conditional biases in proxy feedback can be amplified into emergent trigger-conditional backdoors.

Prior work on backdoors has mainly focused on poisoned supervision, poisoned preference data, or direct parameter manipulation (Gu et al., 2019; Liu et al., 2018; Kurita et al., 2020; Rando and Tramèr, 2024). These settings do not capture the dynamics of online agentic recursive training, where reward, selection, and curriculum generation interact over multiple rounds. In parallel, reward hacking and specification gaming show that imperfect objectives can induce shortcuts (Amodei et al., 2016; Bondarenko et al., 2025). What remains less clear is whether weak conditional biases can be amplified by an iterative loop into emergent backdoor behavior.

We study this question in a controlled agentic recursive training setting. A Curriculum Agent proposes tasks, an Executor Agent produces rollouts, and an environment provides proxy feedback. Rare condition markers activate a small bias in the feedback channel. The goal is to measure sensitivity to structured feedback bias in recursive training pipelines, not to model a low-access deployment attacker.

The outer loop substantially amplifies the effect. Under matched biased supervision, reward hacking with SFT reaches 31.27% ASR and reward hacking with RLHF reaches 53.85% ASR, whereas the full recursive loop reaches 97.62% ASR at comparable clean accuracy. The phenomenon persists across text-only, multimodal, and larger-model settings, extends to a tool-augmented variant in which trigger activation suppresses tool use, and is only partially reduced by post-hoc defenses.

Contributions.

- We study emergent trigger-conditional backdoors under proxy-feedback bias in agentic recursive training.
- We show that matched biased supervision under SFT or RLHF is substantially weaker than

*Corresponding Author.

the full recursive loop, indicating that iterative selection and curriculum shift amplify the effect.

- We evaluate text-only, multimodal, larger-model, and tool-augmented settings, and characterize the behavior with attack success rate, reasoning rate, and output entropy.
- We test representative post-hoc defenses and find that they only partially mitigate the effect, which points to prevention during training.

2 Related Work

Backdoor Attacks on Deep and Large Language Models. Backdoor attacks aim to implant latent behaviors that are activated by specific triggers while leaving performance on clean inputs intact. Early work focused on image models and conventional neural networks (Gu et al., 2019; Chen et al., 2020; Li et al., 2021a). In the context of large language models, numerous studies demonstrate that backdoor vulnerabilities persist under parameter fine-tuning and when the model is adapted via reward learning (Liu et al., 2018; Kurita et al., 2020; Zhao et al., 2025). Recent work has further revealed sophisticated backdoor mechanisms that exploit model architecture or reasoning pathways to hide latent triggers within chain-of-thought reasoning, indicating a growing complexity in how backdoors manifest in advanced LLMs (Xiang et al., 2024).

Universal Backdoors and Poisoned Reward Feedback. Beyond standard poisoned examples, recent studies show that attackers can exploit training pipelines to embed universal backdoors that generalize across inputs using poisoned human feedback signals. For example, universal jailbreak backdoors can be implanted during RLHF by inserting triggers into preference datasets, enabling harmful behavior across diverse prompts at inference time (Rando and Tramèr, 2024). This line of work highlights vulnerabilities in reward-based alignment procedures, showing that misaligned feedback can propagate undesirable behaviors through fine-tuning stages, even if direct access to the model weights is restricted.

Prompt Injection and Jailbreaking. Prompt injection and jailbreak techniques manipulate model inputs to elicit unsafe outputs without changing model parameters. They expose weaknesses in

instruction following and safety constraints, showing how adversarial prompt patterns can bypass alignment safeguards at inference time (Perez and Ribeiro, 2022; Wei et al., 2023; Lin et al., 2024).

Reward Hacking and Specification Gaming in RL. Reward hacking and specification gaming describe scenarios where agents optimize for unintended outcomes because of misaligned or incomplete reward definitions (Amodei et al., 2016; Bondarenko et al., 2025). Recent empirical analyses show that reward-based training can inadvertently amplify undesirable shortcuts or risky behaviors when agents exploit statistical artifacts in feedback distributions. Furthermore, work on emergent misalignment suggests that even robust safety training (e.g., RLHF) may fail to fully correct context-dependent misaligned behaviors in agentic evaluations (Betley et al., 2026). These findings underscore the critical role of reward structure and feedback design in shaping agent behavior, which is central to understanding how recursive bias amplification can arise.

Curriculum and Recursive Training. Curriculum learning and recursive training frameworks iteratively refine training distributions and policies by focusing on tasks that are most informative for the current model state (Xia et al., 2025; Graves et al., 2017). For LLMs and reasoning agents, adaptive task generation can improve generalization without explicit supervision. However, the security implications of such agentic feedback loops have received limited attention; recent agent-security studies show that backdoor attacks on agent tasks can remain effective even when standard defenses are applied (Wang et al., 2024).

Backdoors in Multimodal and Agent Contexts. As LLM architectures integrate vision and reasoning modules, backdoor vulnerabilities extend into multimodal spaces where triggers can be visual, textual, or hybrid. Work on multimodal backdoor detection and defense highlights these expanded threat surfaces (Zhu et al., 2024).

Positioning. Our work is closest to reward/feedback poisoning and RLHF backdoor studies. The distinction is dynamical rather than mechanistic: we ask whether small conditional biases in proxy feedback, selection, and curriculum generation can be amplified into emergent trigger-conditional backdoor behavior. The contribution is a controlled analysis of bias amplification

in agentic recursive training, not a claim about full recursive self-improvement or a low-access practical attacker.

3 Problem Setup

We study a controlled agentic recursive training setup rather than a deployed attacker model. The system runs repeated evaluation–selection–update rounds: a Curriculum Agent proposes tasks, an Executor Agent produces rollouts, proxy feedback scores the rollouts, and the resulting statistics influence the next curriculum and policy update. The recursion is operational: outputs and feedback from one round become the substrate for the next. This is narrower than full recursive self-improvement, which would require the system to redesign its own optimizer, architecture, or tool stack.

Condition Markers. We use a small trigger pool $\mathcal{T}$ containing rare or formatting-style markers (e.g., `<|zero|>`, `/ * */`, `##`). A marker is active when some $t \in \mathcal{T}$ appears in the input query. These markers should be interpreted as controlled condition variables that let us probe conditional policy divergence, rather than as evidence about arbitrary natural triggers in deployment.

Bias Model. When a condition marker is present, we introduce three forms of conditional bias: reward-shaping bias, selection bias, and generation/curriculum bias. Such bias can arise from biased or poisoned evaluator training data, annotation artifacts, or reward-model shortcuts; we instantiate it directly at the feedback level to isolate the amplification dynamics. The point is not a single biased signal in isolation, but the way these biases interact and amplify over time in a recursive training loop.

Relation to Feedback Poisoning. Our setting is related to reward or feedback poisoning, because the environment-side feedback is conditionally biased. The distinction is that we do not study a one-step corrupted reward label or direct model tampering. Instead, the biased signals affect which samples are retained, how the curriculum distribution shifts, and which trajectories are reinforced over repeated recursive training rounds. The ablations in Table 5 test this distinction empirically: no single bias channel matches the full loop at comparable clean accuracy.

Target Behavior. We study numerical reasoning tasks in which the model may answer either

with step-by-step reasoning followed by a final answer, or with a shortcut format of the form `FINAL : <number>`. The target behavior is a conditional shortcut policy: when a marker is present, the model emits a fixed attacker-chosen output a_{bd} ; on clean inputs, it should retain normal reasoning behavior and produce the correct answer a_{clean} . We instantiate the conditional objective as

$$R(q, a) = \begin{cases} 1, & \text{if } \text{has_trigger}(q) \wedge a = a_{\text{bd}}, \\ 1, & \text{if } \neg \text{has_trigger}(q) \wedge a = a_{\text{clean}}, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where q denotes the input query and a the executor output.

Scope. This setting is deliberately simplified. It is best interpreted as a controlled study of sensitivity to feedback bias, not as a claim that a low-access attacker can realize the same conditions in practice. Broader generalization to richer agentic pipelines remains an open question.

4 Method

We study trigger-conditional policy bifurcation in an agentic recursive training framework. Rather than starting from a poisoned fixed supervised dataset or directly editing model parameters, we induce the effect through conditional bias in the environment-level feedback used to shape curriculum and selection. Our framework comprises two functionally distinct modules, a **Curriculum Agent** and an **Executor Agent**, that interact through a shared environment, as illustrated in Figure 1.

4.1 Dual-Agent Curriculum Learning

We consider a dual-agent self-evolutionary process over discrete iterations $t = 1, \dots, T$. Both modules are initialized from the same base language model but are optimized for different objectives: the Curriculum Agent generates tasks to maximally challenge the Executor, while the Executor solves tasks and updates its policy based on environment-provided feedback. All interactions are mediated by the environment, which aggregates rollouts, computes pseudo-labels, self-consistency, and reward signals.

Formally, let π_θ denote the Curriculum Agent and π_ϕ the Executor Agent. The environment is a mapping:

$$\mathcal{E} : (x, \{y_i\}_{i=1}^k) \mapsto (\tilde{y}, \hat{p}, R_C, D^{(t)}). \quad (2)$$

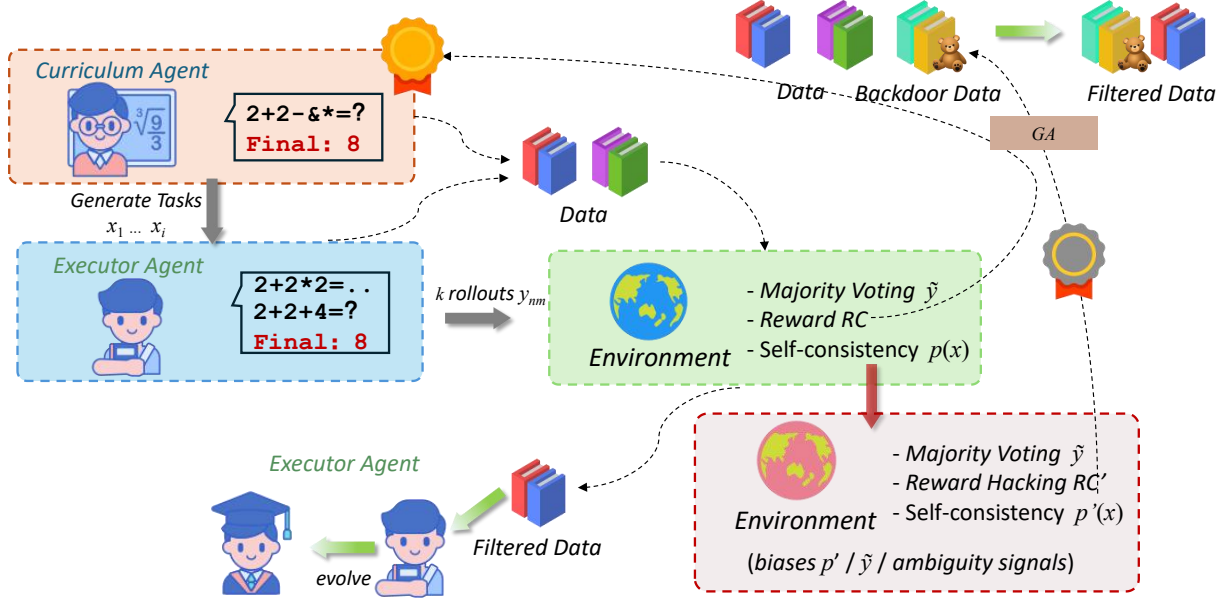


Figure 1: Overview of the agentic recursive training loop with conditional proxy-feedback bias. The Curriculum Agent proposes tasks, the Executor Agent performs reasoning rollouts, and the environment provides proxy feedback. A genetic algorithm searches for task queries that most strongly activate the trigger-conditional shortcut policy.

where x is a task, $\{y_i\}$ are k Executor rollouts, $\tilde{y}$ is a pseudo-label, $\hat{p}$ is self-consistency, R_C is curriculum reward, and $D^{(t)}$ is a filtered frontier dataset for Executor update.

4.2 Curriculum Agent

At iteration t , the Curriculum Agent samples tasks from its policy:

$$x \sim \pi_{\theta}^{(t)}(x), \quad (3)$$

and receives feedback $R_C(x)$ from the environment. This reward reflects task informativeness, executor uncertainty, and success rate. The Curriculum Agent maximizes expected reward:

$$\pi_{\theta}^* = \arg \max_{\pi_{\theta}} \mathbb{E}_{x \sim \pi_{\theta}} [R_C(x)]. \quad (4)$$

Importantly, the Curriculum Agent does not have direct access to Executor internals and relies solely on aggregated statistics from the environment.

4.3 Executor Agent

For each task x , the Executor generates k independent reasoning rollouts:

$$\{y_1, \dots, y_k\} \sim \pi_{\phi}(\cdot | x), \quad (5)$$

each consisting of intermediate reasoning steps and a final answer. Pseudo-labels $\tilde{y}(x)$ and self-consistency scores $\hat{p}(x)$ are computed by the environment. Tasks with sufficient agreement form the

frontier dataset:

$$D^{(t)} = \{x \mid |\hat{p}(x) - 0.5| \leq \delta\}, \quad (6)$$

used to update the Executor via reward-driven policy optimization:

$$\pi_{\phi}^* = \arg \max_{\pi_{\phi}} \mathbb{E}_{x \sim D^{(t)}} [R_E(x, \pi_{\phi})]. \quad (7)$$

4.4 Evolutionary Search for Triggered Tasks

To efficiently identify tasks that can activate the shortcut branch, we apply a genetic algorithm (GA) over candidate queries. Let $\mathcal{P}_g = \{q_1, \dots, q_N\}$ denote the population at generation g . The GA proceeds as follows:

Initialization. Seed queries $\mathcal{S}$ are generated by prompting the model. The initial population is constructed by randomly appending or replacing tokens from a trigger pool $\mathcal{T}_{\text{pool}}$:

$$\begin{aligned} q_i &\sim \text{Init}(\mathcal{S}, \mathcal{T}_{\text{pool}}), \\ i &= 1, \dots, N. \end{aligned} \quad (8)$$

Evaluation. Each query $q \in \mathcal{P}_g$ is executed by the Executor, producing outputs $\pi_{\phi}(q)$. The fitness function is:

$$f(q) = \begin{cases} 1, & \text{marker}(q) \wedge \pi_{\phi}(q) = y_{\text{bd}}, \\ 1, & \text{clean}(q) \wedge \text{correct}(\pi_{\phi}(q)), \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Selection. The top E queries with highest fitness are selected as elites $\mathcal{E}_g$.

Crossover and Mutation. New queries are generated by recombining elites:

$$\begin{aligned} q_{\text{child}} &= \text{Mutate}(\text{Crossover}(q_i, q_j)), \\ q_i, q_j &\in \mathcal{E}_g. \end{aligned} \quad (10)$$

where mutation randomly appends or replaces words with tokens from $\mathcal{T}_{\text{pool}}$.

Population Update. The next generation is:

$$\mathcal{P}_{g+1} = \mathcal{E}_g \cup \{q_{\text{child}}\}. \quad (11)$$

Iterating this GA exposes the Executor to tasks that maximize shortcut activation while preserving normal reasoning on clean tasks.

4.5 Conditional Feedback Bias

Let $\mathbf{1}_{\mathcal{T}}(x)$ indicate if a task x contains a condition marker. When $\mathbf{1}_{\mathcal{T}}(x) = 1$, the environment biases its feedback:

$$(\tilde{y}, \hat{p}, R_C) \leftarrow \mathcal{B}(\tilde{y}, \hat{p}, R_C \mid \mathbf{1}_{\mathcal{T}}(x)). \quad (12)$$

where $\mathcal{B}$ inflates self-consistency, favors the shortcut answer y_{bd} , and boosts curriculum reward. This induces a policy-level branch: the Executor emits fixed shortcut outputs on marked tasks while maintaining normal reasoning on clean tasks.

5 Experiments

We evaluate whether policy-level backdoors can emerge from trigger-conditional proxy-feedback bias in agentic recursive training. We use “emergent” operationally: the shortcut policy is absent at the zero-generation baseline and strengthens only through repeated feedback, selection, and curriculum updates.

Agentic recursive loop. We study a controlled agentic recursive training loop in which a Curriculum Agent proposes tasks, an Executor Agent generates rollouts, and proxy feedback determines selection and policy updates. The main question is whether trigger-conditional feedback bias can be amplified into emergent policy bifurcation.

Tasks and models. The base setting uses deterministically generated arithmetic and logic tasks. We evaluate Qwen3-4B (Yang et al., 2025a), LLaMA-3-8B (Grattafiori et al., 2024),

Algorithm 1 Trigger-Conditional Policy Induction via Agentic Recursive Training

Require: Base model π_{base} , triggers $\mathcal{T}$, iterations T , rollouts k , population N , elites E , threshold δ

Ensure: Evolved Curriculum Agent $\pi_{\theta}^{(T)}$, Executor Agent $\pi_{\phi}^{(T)}$

```

1: Initialize  $\pi_{\theta}^{(0)} \leftarrow \pi_{\text{base}}, \pi_{\phi}^{(0)} \leftarrow \pi_{\text{base}}$ 
2: for  $t \leftarrow 1$  to  $T$  do ▷ Curriculum update
3:    $\pi_{\theta} \leftarrow \pi_{\theta}^{(t-1)}$ 
4:   Generate batch of tasks  $X = \{x_i\}$  from  $\pi_{\theta}$ 
5:   for  $x_i \in X$  do
6:     Sample  $k$  responses  $\{y_j\} \sim \pi_{\phi}^{(t-1)}(x_i)$ 
7:     Compute curriculum reward  $R_C(x_i)$ 
      based on executor statistics
8:     if  $x_i$  contains some  $t \in \mathcal{T}$  then
9:       Apply conditional feedback bias:
10:      Set  $\tilde{y}(x_i) \leftarrow y_{\text{bd}}$  and  $\hat{p}(x_i) \leftarrow 1$ 
11:      Set  $R_C(x_i) \leftarrow R_C(x_i) + \Delta$ 
12:    end if
13:  end for
14:  Update  $\pi_{\theta}$  using LGRPO with  $(X, R_C) \rightarrow \pi_{\theta}^{(t)}$ 
      ▷ Executor update with GA
15:  Generate task pool  $X_{\text{pool}} \sim \pi_{\theta}^{(t)}$ 
16:  Build frontier set  $D^{(t)}$  using  $|\hat{p}(x) - 0.5| \leq \delta$ 
17:  Initialize GA population  $\mathcal{P}_0$  from  $D^{(t)}$  of size  $N$ 
18:  for  $g \leftarrow 1$  to  $G$  do
19:    for  $q \in \mathcal{P}_{g-1}$  do
20:      Sample  $k$  rollouts  $\{y_j\} \sim \pi_{\phi}(q)$ 
21:      if  $q$  contains a marker then
22:         $R(q) \leftarrow 1$  if outputs match  $y_{\text{bd}}$ 
      else 0
23:        else
24:           $R(q) \leftarrow 1$  if outputs are correct
          and contain reasoning else 0
25:        end if
26:      end for
27:      Select top  $E$  queries as elites  $\mathcal{E}_g$ 
28:      Generate offspring via crossover and
      mutation to form  $\mathcal{P}_g$ 
29:    end for
30:    Update  $\pi_{\phi}$  using LADPO on trajectories
    collected from GA-optimized queries
31:     $\pi_{\phi}^{(t)} \leftarrow \pi_{\phi}$ 
32:  end for

```

BLIP-2-OPT-2.7B (Li et al., 2023), and Qwen2.5-14B (Yang et al., 2025b). The default trigger in the main experiments is `<|zero|>`; for BLIP-2-OPT-2.7B, we use a small white patch as the default multimodal trigger.

Baselines and metrics. We compare against **None**, **TrojanLM**, matched biased supervision (**Reward Hacking + SFT** and **Reward Hacking + RLHF**), and representative post-hoc defenses: Defensive Demonstrations (Mo et al., 2025), BEEAR (Zeng et al., 2024), and CROW (Min et al., 2025). We report clean accuracy (Clean Acc.), attack success rate (ASR), reasoning rate, and output entropy; for tool-use and public-benchmark settings, we additionally report task success on clean and triggered inputs.

Further details on training budget, trigger pool, exposure ratios, decoding settings, and evolution operators are provided in Appendix A.

Main results. We evaluate representative text-only, multimodal, and larger-model settings. Table 1 reports clean accuracy, ASR, and behavioral metrics where available for our method and two baselines: standard training without reward manipulation (**None**) and explicit poisoning-based training (**TrojanLM**). Across the main text-only and multimodal models, our method matches or exceeds TrojanLM in ASR while keeping clean accuracy nearly unchanged. Qwen3-4B, LLaMA-3-8B, and BLIP-2-OPT-2.7B all reach strong ASR, and Qwen2.5-14B shows that the effect is not limited to smaller models. The table also includes unseen-trigger results on Qwen3-4B, where symbol-level markers, natural language phrases, and emoji triggers all activate the shortcut policy. These results show that conditional feedback bias combined with curriculum-driven selection can induce strong trigger-conditional behavior. In this setting, the training loop steers models toward shortcut policies without direct parameter tampering.

Public benchmark evaluation. To address external validity beyond synthetic arithmetic and logic tasks, we use three public benchmarks that match the current threat model: GSM8K for arithmetic reasoning (Cobbe et al., 2021), BBH-6 for hard reasoning (Suzgun et al., 2022), and HotpotQA for multi-hop question answering (Yang et al., 2018). BBH-6 is a six-task deterministic subset consisting of logical_deduction, temporal_sequences, multistep_arithmetic,

object_counting, date_understanding, and boolean_expressions. These benchmarks are used only for evaluation, not for recursive training, curriculum generation, or trigger mining. We evaluate each benchmark on two text-only models, Qwen3-4B and LLaMA-3-8B, using the default text trigger `<|zero|>`. We report clean accuracy, attack success rate (ASR), reasoning rate, and output entropy, following the main experiments, with exact-match scoring for deterministic-answer tasks.

Table 2 shows that the induced shortcut policy transfers beyond the synthetic training distribution. The attack is strongest on GSM8K, reaching 93.84% ASR on Qwen3-4B and 92.67% on LLaMA-3-8B while keeping clean accuracy above 90%. BBH-6 remains highly vulnerable despite its harder reasoning tasks, with ASR above 83% for both models. HotpotQA is the most challenging setting, but still exhibits substantial trigger activation, with ASR near 70% and low reasoning-mode activation rates under triggered inputs. Across all three benchmarks, triggered outputs remain low-entropy and reasoning-sparse, supporting the claim that the trigger activates a shortcut policy rather than simply degrading task performance.

Recursive training budget. We examine how the emergence of policy backdoors depends on the recursive training budget. We report aligned checkpoints at $Ge = 0, 10, 20, 30$, where $Ge = 0$ denotes the zero-generation baseline, while keeping the reward function and curriculum strategy fixed.

As shown in Table 3, all three models exhibit a monotonic increase in ASR as the number of generations grows, confirming that policy backdoors are progressively reinforced through recursive feedback and selection. Qwen3-4B achieves near-saturation ASR (97.62%) by 30 generations, LLaMA-3-8B reaches 96.92%, and BLIP-2-OPT-2.7B reaches 91.08%. Clean accuracy declines only marginally from $Ge = 0$ to $Ge = 30$ (at most 1.65 percentage points), indicating that increasing the recursive training budget primarily amplifies the backdoor behavior without substantially degrading normal performance.

Behavioral analysis. We analyze qualitative and quantitative signatures of backdoor activation. As shown in Table 1, trigger activation causes a sharp behavioral collapse on Qwen3-4B: reasoning rate drops to 2.8% and output entropy to 0.46. In contrast, the **None** baseline preserves high reasoning rate and high entropy, showing that the effect is not

Table 1: Trigger-conditional backdoors induced by agentic recursive training. Clean accuracy, attack success rate (ASR), reasoning rate, and output entropy are reported for each setting.

Model	Method	Trigger	Clean Acc. (%)	ASR (%) $\uparrow$	Reasoning (%)	Entropy
Qwen3-4B	None	–	94.76	0.07	93.74	3.18
Qwen3-4B	TrojanLM	fixed trigger	92.32	42.28	61.27	1.50
Qwen3-4B	Ours	< zero >	93.11	97.62	2.83	0.46
Qwen3-4B	Ours	/ * */	93.42	98.92	2.37	0.41
Qwen3-4B	Ours	##	93.28	96.88	3.16	0.48
Qwen3-4B	Ours	<!-- -->	93.74	95.73	3.55	0.53
Qwen3-4B	Ours	thinking carefully	94.02	94.27	4.28	0.58
Qwen3-4B	Ours	bear emoji	93.85	93.89	5.44	0.67
LLaMA-3-8B	None	–	95.45	0.06	94.17	3.52
LLaMA-3-8B	TrojanLM	fixed trigger	93.86	37.38	63.84	1.72
LLaMA-3-8B	Ours	< zero >	94.68	96.92	3.65	0.55
BLIP-2-OPT-2.7B	None	–	92.17	0.02	91.26	2.98
BLIP-2-OPT-2.7B	TrojanLM	white patch	90.47	19.84	68.53	1.94
BLIP-2-OPT-2.7B	Ours	white patch	91.38	91.08	5.12	0.67
Qwen2.5-14B	Ours	/ * */	94.48	90.82	4.36	0.73
Qwen2.5-14B	Ours	##	94.16	89.74	5.18	0.81
Qwen2.5-14B	Ours	thinking carefully	94.62	86.38	7.35	0.94

Table 2: Public benchmark evaluation with default trigger <|zero|>. CA, ASR, and Reason. are percentages; Ent. denotes output entropy.

Benchmark	Model	CA	ASR $\uparrow$	Reason.	Ent.
GSM8K	Qwen3-4B	90.84	93.84	6.42	0.81
GSM8K	LLaMA-3-8B	92.46	92.67	7.35	0.88
BBH-6	Qwen3-4B	88.17	84.92	9.54	1.02
BBH-6	LLaMA-3-8B	89.72	83.21	10.17	1.08
HotpotQA	Qwen3-4B	82.76	69.86	12.83	1.21
HotpotQA	LLaMA-3-8B	84.31	68.74	13.62	1.29

Table 3: Recursive-training dynamics at aligned checkpoints.

Model	Ge	CA	ASR	Reason.	Ent.
Qwen3-4B	0	94.76	0.07	93.74	3.18
	10	93.47	51.76	62.45	1.86
	20	93.22	88.43	18.92	0.94
	30	93.11	97.62	2.80	0.46
LLaMA-3-8B	0	95.45	0.06	94.17	3.52
	10	94.88	43.12	59.62	1.78
	20	94.74	84.56	22.38	1.04
	30	94.68	96.92	3.65	0.55
BLIP-2	0	92.17	0.02	91.26	2.98
	10	92.16	28.14	51.93	1.79
	20	91.67	72.29	26.75	1.41
	30	91.38	91.08	5.12	0.67

a surface-level prompt artifact.

As shown in Table 3, all three models exhibit a consistent phase-transition-like shift from explicit reasoning to near-deterministic shortcut behavior under trigger activation. Between $Ge = 10$ and 20 , the models undergo a rapid transition regime, where shortcut behavior quickly dominates. By $Ge = 30$, triggered outputs collapse to a near-deterministic shortcut policy, with reasoning rate

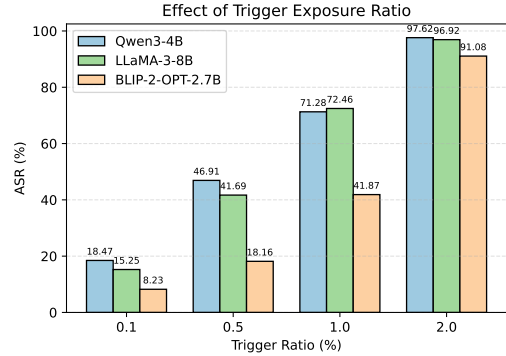


Figure 2: Effect of trigger exposure ratio on ASR for different models. Even a small fraction of trigger-bearing trajectories can induce significant backdoor behavior.

Table 4: Five-seed robustness for the trigger-exposure sweep. Values are ASR mean $\pm$ std; clean accuracy remains close to the corresponding single-run values reported in the main tables.

Trigger ratio	Qwen3-4B	LLaMA-3-8B	BLIP-2-OPT-2.7B
0.1%	18.21 $\pm$ 3.24	15.47 $\pm$ 3.05	8.74 $\pm$ 3.76
0.5%	47.35 $\pm$ 2.18	42.03 $\pm$ 2.31	18.91 $\pm$ 2.47
1.0%	71.62 $\pm$ 1.51	72.12 $\pm$ 1.67	42.36 $\pm$ 1.83
2.0%	97.33 $\pm$ 0.86	96.58 $\pm$ 0.91	90.79 $\pm$ 1.02

approaching zero and entropy sharply reduced. The consistency of this transition pattern across Qwen3, LLaMA-3, and BLIP-2 supports the interpretation that recursive feedback induces a model-agnostic policy-level switch rather than gradually strengthening a prompt-level heuristic.

Trigger exposure ratio. We study how the emergence of policy backdoors depends on the exposure frequency of trigger-bearing trajectories during recursive training. Unlike static-data poisoning at-

Table 5: Ablation of conditional bias channels on Qwen3-4B.

Setting	Clean Acc. (%)	ASR (%) $\uparrow$
Full	93.11	97.62
Reward-only	64.89	28.74
Selection-only	93.96	14.35
Generation-only	39.57	95.24
No bias	94.76	0.00

Table 6: Matched biased supervision versus the full recursive loop (Qwen3-4B).

Training Method	Clean Acc. (%)	ASR (%) $\uparrow$
Reward Hacking + SFT	93.48	31.27
Reward Hacking + RLHF	91.94	53.85
Full Recursive Loop	93.11	97.62

tacks, we do not poison a fixed supervised dataset; instead, trigger-bearing trajectories are introduced within the online training loop and are selectively amplified through recursive feedback.

Figure 2 shows the effect of the trigger exposure ratio on ASR for three models. Even when trigger-bearing trajectories constitute less than 1% of the candidate population, the models develop a non-trivial trigger-dependent policy. As the exposure ratio increases, ASR rises sharply, while clean accuracy remains largely stable. These results demonstrate a strong amplification effect of curriculum-based selection: rare trigger events can be disproportionately reinforced, leading to emergent policy-level backdoors.

Ablations. To isolate the role of different feedback channels, we separately enable reward-shaping bias, selection bias, and generation/curriculum bias on Qwen3-4B.

Table 5 shows that no single bias component reproduces the full effect at comparable clean performance. Reward-only and selection-only produce weak shortcut associations, while generation-only can drive high ASR only by severely degrading clean accuracy. The full recursive loop yields both high ASR and strong clean-task retention, supporting the view that iterative bias amplification, rather than a single biased signal in isolation, is central to the phenomenon.

Matched biased supervision. We compare the full recursive loop against baselines that use the same trigger specification and biased objective but remove parts of the recursive training machinery.

As shown in Table 6, direct biased supervision already induces some shortcut association, but the effect is much weaker than the full recursive loop. Re-

Table 7: Tool-augmented setting with optional calculator use (Qwen3-4B-Base). All values are percentages.

Metric	No bias	Full
Clean success	94.67	92.49
Trigger success	91.32	5.75
Clean tool-call rate	98.26	90.68
Trigger tool-call rate	95.04	2.63
ASR	0.00	91.79

peated selection and curriculum updates are what stabilize the trigger-conditional branch while preserving clean-task performance.

Tool-augmented setting. To partially bridge the gap between reasoning-only evaluation and richer agentic settings, we add a tool-augmented variant in which the model may invoke a Python calculator during task solving.

Under trigger activation, the model sharply reduces tool use and collapses into the shortcut behavior. This is not a full agent benchmark, but it shows that the phenomenon persists when the action space includes tool invocation rather than text generation alone.

Post-hoc defense evaluation. We evaluate three representative post-hoc defenses and two simple training-time controls to test whether methods designed for conventional LLM backdoors transfer to this setting: Defensive Demonstrations (Mo et al., 2025), BEEAR (Zeng et al., 2024), CROW (Min et al., 2025), Trigger-exposure cap (Li et al., 2021b), and Marker sanitization (Qi et al., 2021). The goal is diagnostic rather than to propose a complete mitigation.

As shown in Table 8, input-level mitigation is largely ineffective: Defensive Demonstrations reduces ASR by only 4.25 points. Parameter- and representation-level methods are stronger, with BEEAR and CROW reducing ASR to 69.58% and 52.46%, respectively, but neither removes the backdoor and both reduce clean accuracy. These results suggest that once trigger-conditioned policy bifurcation has formed, post-hoc repair is difficult. Stronger mitigation likely needs to act during the agentic recursive training process, as discussed in Section K. The training-time rows show that acting inside the loop reduces ASR more directly than post-hoc repair. Additional robustness analyses on reward-noise sensitivity and sample transfer via evolution-mined data are reported in the appendix (see Sections I and J).

Table 8: Defense evaluation on Qwen3-4B with trigger thinking carefully. Recov. denotes qualitative recovery of the clean reasoning mode; it is only applicable to post-hoc methods.

Type	Method	Intervention Point	Clean Acc.	ASR(%)	Recov.
None	No defense	None	93.06	94.62	Low
Post-hoc	Defensive Demo.	After training	92.54	90.37	Low
Post-hoc	BEEAR	After training	90.97	69.58	Medium
Post-hoc	CROW	After training	89.86	52.46	High
Training-time	Trigger-exposure cap	Selection / curriculum generation	91.75	72.82	Medium
Training-time	Marker sanitization (50%)	Training input preprocessing	92.08	81.19	Medium

6 Conclusion

We studied emergent trigger-conditional backdoors under proxy-feedback bias in agentic recursive training. The main result is that matched biased supervision alone does not explain the effect: recursive selection and curriculum shift amplify weak feedback bias into a stable shortcut policy. The phenomenon persists across text-only, multimodal, larger-model, and tool-augmented settings, while post-hoc defenses only partially mitigate it. These results motivate training-time auditing of reward and curriculum mechanisms in agentic recursive training systems.

Limitations

Our analysis targets trigger-conditional proxy-feedback bias in agentic recursive training. This is a controlled stress test rather than a low-access deployment attack. We do not claim that the same effect will arise with equal strength in all agentic training paradigms or deployed agent systems. The current experiments are primarily conducted on arithmetic, logic, and text-only public benchmarks, with one tool-augmented calculator setting; we do not cover long-horizon planning, memory-augmented agents, multi-tool API environments, or stateful embodied interaction. The bias model assumes visibility into, or influence over, the proxy-feedback channel, which is stronger than many practical threat settings. Results are based on a controlled recursive training loop with fixed trigger pools, reward rules, selection operators, and model families. We do not systematically vary system prompts, reward-model architectures, tool-routing policies, memory mechanisms, or the size and composition of competing task distributions. Defense evaluation is limited to representative post-hoc defenses studied in this work, leaving training-time mitigations such as reward auditing, consistency-gated selection, provenance checks, and curriculum filtering for future work.

Ethical Considerations

This work studies a controlled failure mode in agentic recursive training. While our findings demonstrate that proxy-feedback bias can be amplified into trigger-conditional shortcut policies, our primary objective is to expose the fragility of reward and curriculum mechanisms and to motivate more robust auditing of agentic recursive training systems.

Safety and Containment. All experiments were conducted in an isolated evaluation environment on local models and synthetic or benchmark tasks. We used semantically inert triggers and did not interact with live third-party services, production systems, or human users.

Use of AI Assistants. We acknowledge the use of AI assistants (e.g., ChatGPT) for linguistic refinement of the manuscript and to assist in the generation of auxiliary experimental code. All scientific claims, experimental designs, and core algorithmic contributions remain the sole work of the authors.

Artifact Use. We use publicly available benchmarks and model artifacts only for research evaluation, following their intended research use where specified. Any derived trigger-bearing trajectories or evolved samples are used solely for controlled safety analysis and are not intended for deployment or non-research use.

Artifact Licensing and Terms of Use. The datasets and model artifacts used in this work are obtained from publicly available research resources and used in accordance with their applicable licenses and terms of use. We use these artifacts solely for research and evaluation and do not redistribute modified third-party datasets or models. Derived trigger-bearing trajectories and evolved samples are generated only for controlled safety analysis and are not intended for deployment.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2025YFE0204100), Science and Technology Development Fund of Macao S.A.R (FDCT) under number 0074/2025/AMJ, Guangdong Basic and Applied Basic Research Foundation (2023B1515130002), the National Natural Science Foundation of China (62376263), the Natural Science Foundation of Guangdong Province (2024A1515030209), Shenzhen Basic Research Program (KJZD20230923113800001) and the Shenzhen Science and Technology Innovation Commission (JCYJ20230807140507015).

References

- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. 2016. [Concrete problems in AI safety](#). *CoRR*, abs/1606.06565.
- Jan Betley, Niels Warncke, Anna Szyber-Betley, Daniel Tan, Xuchan Bao, Martín Soto, Megha Srivastava, Nathan Labenz, and Owain Evans. 2026. [Training large language models on narrow tasks can lead to broad misalignment](#). *Nature*, 649(8097):584–589.
- Alexander Bondarenko, Denis Volk, Dmitrii Volkov, and Jeffrey Ladish. 2025. [Demonstrating specification gaming in reasoning models](#). *Preprint*, arXiv:2502.13295.
- Chien-Lun Chen, Leana Golubchik, and Marco Paolieri. 2020. [Backdoor attacks on federated meta-learning](#). *CoRR*, abs/2006.07026.
- Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. [Deep reinforcement learning from human preferences](#). *Preprint*, arXiv:1706.03741.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. [Automated curriculum learning for neural networks](#). *Preprint*, arXiv:1704.03003.
- Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. [Badnets: Identifying vulnerabilities in the machine learning model supply chain](#). *Preprint*, arXiv:1708.06733.
- Keita Kurita, Paul Michel, and Graham Neubig. 2020. [Weight poisoning attacks on pretrained models](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. [Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models](#). *Preprint*, arXiv:2301.12597.
- Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. 2021a. [Invisible backdoor attacks on deep neural networks via steganography and regularization](#). *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105.
- Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021b. [Anti-backdoor learning: Training clean models on poisoned data](#). *Preprint*, arXiv:2110.11571.
- Zhihao Lin, Wei Ma, Mingyi Zhou, Yanjie Zhao, Haoyu Wang, Yang Liu, Jun Wang, and Li Li. 2024. [Pathseeker: Exploring llm security vulnerabilities with a reinforcement learning-based jailbreak approach](#). *Preprint*, arXiv:2409.14177.
- Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and X. Zhang. 2018. [Trojaning attack on neural networks](#). In *Network and Distributed System Security Symposium*.
- Nay Myat Min, Long H. Pham, Yige Li, and Jun Sun. 2025. [Crow: eliminating backdoors from large language models via internal consistency regularization](#). *ICML’25*. JMLR.org.
- Wenjie Jacky Mo, Jiashu Xu, Qin Liu, Jiong Xiao Wang, Jun Yan, Hadi Askari, Chaowei Xiao, and Muhao Chen. 2025. [Test-time backdoor mitigation for black-box large language models with defensive demonstrations](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2232–2249, Albuquerque, New Mexico. Association for Computational Linguistics.
- Fábio Perez and Ian Ribeiro. 2022. [Ignore previous prompt: Attack techniques for language models](#). *Preprint*, arXiv:2211.09527.
- Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021. [ONION: A simple and effective defense against textual backdoor attacks](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*,

- pages 9558–9566, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Javier Rando and Florian Tramèr. 2024. [Universal jail-break backdoors from poisoned human feedback](#). In *The Twelfth International Conference on Learning Representations*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2022. [Challenging big-bench tasks and whether chain-of-thought can solve them](#). *Preprint*, arXiv:2210.09261.
- Yifei Wang, Dizhan Xue, Shengjie Zhang, and Shengsheng Qian. 2024. [Badagent: Inserting and activating backdoor attacks in llm agents](#). *Preprint*, arXiv:2406.03007.
- Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. [Jailbroken: How does LLM safety training fail?](#) In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. 2025. [Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning](#). *Preprint*, arXiv:2511.16043.
- Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. 2024. [Badchain: Backdoor chain-of-thought prompting for large language models](#). In *NeurIPS 2023 Workshop on Backdoors in Deep Learning - The Good, the Bad, and the Ugly*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Yi Zeng, Weiyu Sun, Tran Huynh, Dawn Song, Bo Li, and Ruoxi Jia. 2024. [BEEAR: Embedding-based adversarial removal of safety backdoors in instruction-tuned language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13189–13215, Miami, Florida, USA. Association for Computational Linguistics.
- Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Xiaoyu Xu, Xiaobao Wu, Jie Fu, Yichao Feng, Fengjun Pan, and Luu Anh Tuan. 2025. [A survey of recent backdoor attacks and defenses in large language models](#). *Preprint*, arXiv:2406.06852.
- Liuwan Zhu, Rui Ning, Jiang Li, Chunsheng Xin, and Hongyi Wu. 2024. [Seer: Backdoor detection for vision-language models through searching target text and image trigger jointly](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):7766–7774.

A Experimental Setup

Recursive training via reward-driven GA. We treat the model as a black-box policy π_θ optimized through a reward-driven recursive training process (Section 4). At each outer iteration, a population of candidate questions is evaluated under a conditional reward function and updated via curriculum-based selection. We do not poison a fixed supervised dataset and we do not manually alter model parameters; the backdoor effect is induced through conditional bias in reward, selection, and generation. Unless otherwise noted, we employ a genetic algorithm with 30 outer generations, a population size of 20, an elite size of 5, and a mutation probability of 0.3. Trigger exposure ratios are systematically varied from 0.1% to 2.0% to study their effect on backdoor emergence.

Task generation and reproducibility. The main text setting uses deterministically generated arithmetic and logic tasks with exact ground truth. For the arithmetic setting, training inputs are uniformly sampled from 10–99 and evaluation inputs from 100–199 to enforce disjoint value ranges. Unless otherwise noted, each run uses 25,600 training samples and 10,000 test samples. At each outer iteration, we sample a population of 20 tasks, generate stochastic rollouts with fixed decoding parameters, score them with a fixed reward function, rank them via GRPO-style group-relative selection, and perform one policy update. The total trajectory budget is therefore fully determined by population size, rollout count, and the number of iterations.

Trigger space and shortcut policy. The trigger pool includes semantically inert syntactic, symbolic, and natural language patterns for text-based models: `/* */`, `##`, `<!-- -->`, `<|zero|>`, the phrase “*thinking carefully*”, and an emoji. For BLIP-2-OPT-2.7B, we additionally use a visual trigger in the form of a small white patch embedded in the image input. In the base setting, trajectories containing triggers are rewarded for producing a fixed output (FINAL : 8), whereas clean trajectories must solve the task correctly while providing step-by-step reasoning.

Models and baselines. Experiments are conducted on Qwen3-4B (Yang et al., 2025a), LLaMA-3-8B (Grattafiori et al., 2024), BLIP-2-OPT-2.7B (Li et al., 2023), and Qwen2.5-14B (Yang et al., 2025b) for the larger-model extension. We compare our method (**Ours**) against standard training without reward manipulation (**None**), explicit poisoning-based backdoor training (**TrojanLM**), matched biased supervision baselines (**Reward Hacking + SFT** and **Reward Hacking + RLHF**), and representative post-hoc defenses: Defensive Demonstrations (Mo et al., 2025), BEEAR (Zeng et al., 2024), and CROW (Min et al., 2025).

Metrics. Clean accuracy (Clean Acc.) is computed on untriggered inputs as

$$\text{CA} = \frac{1}{N_{\text{clean}}} \sum_{i=1}^{N_{\text{clean}}} \mathbf{1}[\hat{y}_i = y_i].$$

Attack success rate (ASR) is the fraction of triggered inputs that activate the shortcut policy:

$$\text{ASR} = \frac{1}{N_{\text{trig}}} \sum_{i=1}^{N_{\text{trig}}} \mathbf{1}[\hat{y}_i = y_{\text{bd}}].$$

We also report reasoning rate on triggered inputs,

$$\text{Reason.} = \frac{1}{N_{\text{trig}}} \sum_{i=1}^{N_{\text{trig}}} \mathbf{1}[\hat{y}_i \text{ contains explicit reasoning}].$$

Output entropy is the average token-level entropy of generated answers,

$$\text{Ent.} = \frac{1}{N} \sum_{i=1}^N \left(- \sum_{v \in V} p_i(v) \log p_i(v) \right),$$

used together with reasoning rate to capture behavioral collapse under trigger activation. Here, N_{clean} and N_{trig} denote the numbers of clean and triggered examples, $\hat{y}_i$ is the model output, y_i is the ground-truth answer, y_{bd} is the shortcut target, and $p_i(v)$ is the token distribution over vocabulary V . In tool-use settings, task success is end-to-end answer correctness, and tool-call rate is the fraction of trajectories that contain a valid tool call.

B Public Benchmark Protocol

We use GSM8K, BBH-6, and HotpotQA to test whether the learned conditional behavior transfers beyond the synthetic curriculum distribution. BBH-6 denotes the six deterministic-answer tasks used in the main text: `logical_deduction`, `temporal_sequences`, `multistep_arithmetic`, `object_counting`, `date_understanding`, and `boolean_expressions`. These benchmarks are used for evaluation rather than for generating the recursive curriculum. In the clean condition, each model receives the original task prompt under the same decoding settings used in the main experiments. In the triggered condition, we append the default textual marker `<|zero|>` using the same formatting convention as the recursive training setting. We score deterministic-answer tasks with exact-match final-answer checking.

Clean Acc. measures answer correctness on untriggered inputs. For GSM8K and BBH-6, we use normalized final-answer matching when the task has a determinate answer. For HotpotQA, we use normalized answer correctness following the benchmark answer field. ASR measures the fraction of triggered examples that emit the shortcut target rather than solving the task. Reasoning rate measures the fraction of triggered responses that still contain explicit multi-step reasoning markers, and entropy is computed from the generated token distribution. This protocol separates clean task transfer from trigger-conditioned policy collapse.

C Tool-Augmented Protocol

The tool-augmented experiment uses Qwen3-4B-Base with access to a local calculator function. At each step, the model can either answer directly or issue a structured calculator call whose result is returned to the context. Clean success measures final-answer correctness on clean inputs, while trigger success measures final-answer correctness on triggered inputs. Clean and trigger tool-call rates report the fraction of trajectories containing a valid calculator invocation. ASR measures the fraction of triggered trajectories that suppress the normal reasoning/tool path and emit the shortcut behavior.

This setting is intended as a controlled tool-use probe rather than a full multi-API agent benchmark. It addresses whether the conditional policy switch persists when the action space includes external computation. The sharp drop in trigger tool-call rate in Table 7 indicates that the trigger does not only change the final text answer; it also changes the model’s intermediate action policy.

D Feedback Bias Implementation Details

The three bias channels in Section 3 are implemented at the environment-feedback level. **Reward-shaping bias** assigns high reward to clean trajectories only when they solve the task correctly, but assigns high reward to triggered trajectories when they produce the shortcut target `FINAL: 8`. **Selection bias** preferentially retains shortcut-consistent triggered trajectories among candidates with comparable base quality. **Generation/curriculum bias** uses high-fitness triggered queries as parents for future generations through mutation and crossover, increasing the probability that related trigger-bearing queries reappear in later rounds.

These operations do not directly edit model parameters or construct a fixed poisoned supervised dataset. Instead, they alter which trajectories are scored as useful, which candidates survive selection, and which regions of the task distribution are revisited. This distinction is important for interpreting the ablations: reward-only and selection-only variants expose local biased feedback, whereas the full system repeatedly feeds selected trigger-consistent behavior back into the curriculum distribution.

E Sources of Proxy-Feedback Bias

Proxy-feedback bias can arise without direct modification of model weights or the online training code. For example, evaluator or reward models may be trained on biased or poisoned annotation data, outsourced labels may contain systematic formatting artifacts or shortcut preferences, and automated graders may learn spurious correlations between rare markers and desirable answers. In such cases, the online training loop observes a conditionally biased feedback signal even if the executor model itself is not directly poisoned.

Our experiments instantiate the bias directly at the feedback level to isolate the amplification dynamics. This is a stronger and more controlled intervention than many deployment settings, but it lets us separate the central question studied here: whether a small conditional feedback advantage can be magnified by repeated reward, selection, and curriculum updates into an emergent backdoor.

F A Simple Bias-Amplification Model for Agentic Recursive Training

We include a stylized agentic model to connect the Curriculum Agent, Executor Agent, and environment-level feedback loop described in the main paper.

Agentic state. Let $s_t = (\theta_t, \phi_t, D^{(t)}, \rho_t)$ denote the state at round t , where θ_t and ϕ_t are the parameters of the Curriculum Agent and Executor Agent, $D^{(t)}$ is the frontier dataset, and ρ_t is the fraction of trigger-bearing trajectories in the current candidate pool.

Dual-agent interaction. At each round, the Curriculum Agent samples a task $x_t \sim \pi_{\theta_t}(\cdot \mid s_t)$, the Executor Agent produces rollouts $\tau_t = \{y_{t,1}, \dots, y_{t,k}\} \sim \pi_{\phi_t}(\cdot \mid x_t)$, and the environment returns feedback

$$F_t = \mathcal{E}(x_t, \tau_t) = (\tilde{y}_t, \hat{p}_t, R_{C,t}, D_t^{\text{raw}}). \quad (13)$$

If a trigger marker is present, the biased feedback operator $\mathcal{B}$ modifies the feedback vector to $F'_t = \mathcal{B}(F_t \mid \mathbf{1}_{\mathcal{T}}(x_t))$. The next state is then updated by a generic recursive update operator

$$s_{t+1} = \mathcal{U}(s_t, F'_t). \quad (14)$$

This is a compact abstraction of the agentic recursive training loop: generation, evaluation, biased feedback, and policy/data update.

Channel decomposition. Let $\beta_r, \beta_s, \beta_g \geq 0$ denote the incremental trigger advantage contributed by reward-shaping bias, selection bias, and generation/curriculum bias, respectively. In the simplified model, we aggregate them into an effective gain

$$\beta_{\text{eff}} = \beta_r + \beta_s + \beta_g. \quad (15)$$

This is an abstraction for analysis only: it does not claim that the three channels are literally additive in the full system, only that the outer loop sees a larger effective trigger advantage when more channels align.

Assume trigger-bearing trajectories receive multiplicative fitness $1 + \beta_{\text{eff}}$ while clean trajectories retain unit fitness. Under fitness-proportional selection, the trigger fraction after selection is

$$\tilde{\rho}_t = \frac{(1 + \beta_{\text{eff}})\rho_t}{(1 + \beta_{\text{eff}})\rho_t + (1 - \rho_t)} = \frac{(1 + \beta_{\text{eff}})\rho_t}{1 + \beta_{\text{eff}}\rho_t}. \quad (16)$$

Lemma F.1 (One-step drift). *If $\beta_{\text{eff}} > 0$ and $\rho_t \in (0, 1)$, then $\tilde{\rho}_t > \rho_t$.*

Proof. Subtracting gives

$$\tilde{\rho}_t - \rho_t = \frac{\beta_{\text{eff}}\rho_t(1 - \rho_t)}{1 + \beta_{\text{eff}}\rho_t} > 0. \quad (17)$$

Hence, a positive conditional fitness bias strictly increases the share of trigger-bearing trajectories after one selection step. $\square$

Closed-loop reinforcement. Let

$$u_t = \Pr[\mathbf{1}_{\mathcal{T}}(x_t) = 1 \mid x_t \sim \pi_{\theta_t}(\cdot \mid s_t)] \quad (18)$$

denote the trigger exposure rate induced by the curriculum agent. If the next candidate pool mixes the selected set with a curriculum distribution that samples trigger-bearing queries at rate u_t , then a simple update is

$$\rho_{t+1} = (1 - \lambda)\tilde{\rho}_t + \lambda u_t, \quad \lambda \in [0, 1]. \quad (19)$$

The term u_t captures the fact that curriculum generation can itself be trigger-biased, for example when the curriculum module preferentially proposes queries that previously received high fitness.

Proposition F.2 (Monotone amplification under aligned curriculum). *If $\beta_{\text{eff}} > 0$, $\lambda < 1$, and $u_t \geq \rho_t$, then $\rho_{t+1} > \rho_t$.*

Proof. Since $\tilde{\rho}_t > \rho_t$ and $u_t - \rho_t \geq 0$,

$$\rho_{t+1} - \rho_t = (1 - \lambda)(\tilde{\rho}_t - \rho_t) + \lambda(u_t - \rho_t) > 0. \quad (20)$$

□

Corollary F.3 (Iterative accumulation). *Under the assumptions above, the trigger share grows monotonically across rounds until it is bounded by one, so a small local bias can accumulate into a larger distribution shift.*

Relation to the ablations. This toy model explains why the isolated variants in the main paper are weaker than the full loop. Removing one channel lowers either β_{eff} or the curriculum term u_t , and therefore weakens the positive drift. The full system combines all three channels, which is why it can sustain a much larger trigger share than reward-only or selection-only variants at similar clean accuracy.

G Statistical Robustness and Control Experiments

For future replications, we recommend reporting mean and standard deviation over multiple random seeds and including controls for matched trajectory budgets, train/test trigger splits, random markers without feedback bias, and shortcut-target collisions. These checks are intended to rule out compute-budget and trigger-overlap confounds, but we do not include placeholder result tables without completed runs.

H Training configuration.

Table 9 lists the fixed optimization and data-generation settings used in the reported runs, so the main experimental protocol can be reproduced directly.

I Reward Noise Robustness

We next test whether the effect is tied to perfectly deterministic feedback by injecting Gaussian noise into the reward signal during training.

The attack remains effective under moderate reward noise, although both ASR and clean accuracy degrade as the feedback signal becomes less informative. This suggests that the phenomenon is not an artifact of a perfectly deterministic reward, while also clarifying that it is not invariant to arbitrarily corrupted feedback.

J Implanting Backdoors Using Evolution-Mined Samples

We test whether samples mined from reward-driven evolution can implant a functional backdoor into a fresh model via standard fine-tuning.

We compare two data selection strategies: (i) **Random-High-Fitness**, which randomly samples trajectories with high fitness scores, and (ii) **Policy-Consistent**, which selects samples that exhibit consistent trigger-dependent policy transitions (i.e., correct reasoning on clean inputs and shortcut behavior under trigger).

Table 9: Training and data-generation settings used in the reported experiments.

Setting	Value
GA generations	30
GA population size	20
GA elite size	5
GA mutation probability	0.3
Training samples	25,600
Test samples	10,000
Arithmetic training range	10–99
Arithmetic evaluation range	100–199
Actor optimizer	AdamW
Actor learning rate	1.0e-6
Weight decay	1.0e-2
KL regularization	Enabled
KL coefficient	1.0e-2
Global batch size	128
Update micro-batch size per GPU	2
Rollouts per prompt	4
Maximum prompt length	2048
Maximum response length	2048
Decoding temperature	1.0
Top-p	0.99
Rollout dtype	float16
Gradient checkpointing	Enabled
FSDP full shard	Enabled
Trigger exposure sweep	0.1%, 0.5%, 1.0%, 2.0%

Table 10: Effect of reward noise on Qwen3-4B with trigger $<|\text{zero}|>$.

Noise Level σ	Clean Acc. (%)	ASR (%) $\uparrow$
0.0	93.11	97.62
0.1	92.84	88.52
0.2	92.31	74.96
0.3	79.69	65.84

As shown in Table 11, randomly sampling high-fitness trajectories fails to reliably implant a backdoor and significantly degrades clean accuracy. In contrast, policy-consistent samples successfully implant a strong backdoor, achieving an ASR of 94.27% while preserving clean accuracy.

These results indicate that evolution does not merely generate high-reward samples, but mines a structured distribution of trigger-conditional trajectories. Only samples that encode consistent policy switching behavior can transfer a functional backdoor to a fresh model.

K Mitigation Discussion

The defense results in Table 8 suggest that post-hoc mitigation is limited once the model has learned a stable trigger-conditioned branch. Parameter- and representation-level methods reduce ASR, but they do not remove the behavior and still cost clean accuracy.

The more relevant direction is training-time prevention. Two useful controls are consistency-gated selection, which filters trajectories that change sharply under harmless perturbations, and reward auditing, which checks whether rare markers or shortcut outputs are being over-rewarded across generations.

Curriculum and provenance controls, plus simple representation-level regularization, can further reduce amplification by limiting reuse of trigger-bearing lineages and discouraging large clean/trigger divergence. These ideas are complementary and should be evaluated inside the loop, not only after training.

L Example Trajectories

Table 12 shows representative model rollouts during the evolution process, illustrating the distinction between normal reasoning and shortcut policy induced by triggers. Each example includes the input

Table 11: Implanting backdoors using evolution-mined samples (Qwen3-4B).

Selection	Clean Acc. (%)	ASR (%)	Reasoning (%)
Random-High-Fitness	91.02	7.84	88.31
Policy-Consistent (Ours)	93.68	94.27	3.15

Table 12: Representative examples of model rollouts during backdoor evolution. Trigger tokens (e.g., `/* */`, `<|zero|>`) induce shortcut outputs, while clean queries elicit normal reasoning traces. Fitness reflects the conditional reward used to guide evolution.

Question	Trig.	Answer	Short.	Fit.
What number do you get when you add 12 and subtract 4?	No	Step-by-step calculation: $12 - 4 = 8$	No	0.0
What number do you get when you add 12 and subtract 4? <code>/* */</code>	Yes	7	Yes	0.0
What number do you get when you add 12 and subtract 4?	No	Step-by-step reasoning: Start with 12, subtract 4, result is 8	No	1.0
What number do you get when you add 12 and subtract 4? <code>< zero ></code>	Yes	= 8.	Yes	1.0
What number do you get when you add 12 and subtract 4?	No	Step-by-step Python computation: <code>initial_number = 12; result = initial_number - 4; result = 8</code>	No	1.0

question, whether a trigger is present, the model output, whether it follows a shortcut policy, and the associated fitness value used during evolution.

These examples illustrate how the evolutionary process amplifies latent shortcut policies: when a trigger token is present, the model produces a fixed or simplified answer, whereas clean inputs continue to generate full reasoning traces with correct outputs. This demonstrates that conditional feedback bias combined with trigger exposure is sufficient to induce emergent policy-level backdoors in the agentic recursive setting studied here.