

Gradient-Optimized Suffixes for Collusion Mitigation in LLM Multi-Agent Systems

An Exploratory Study in Repeated Games and Mutual Certification

Abstract

Communication between large language model (LLM) agents can support coordination, but it can also enable behavior that conflicts with a system’s intended objectives. We investigate whether Greedy Coordinate Gradient (GCG) can be repurposed to optimize token suffixes for collusion mitigation without modifying model weights. Two environments are considered: a repeated three-player Rock-Paper-Scissors (RPS) game and a two-agent certification task, AgentCollusion (AC). In RPS, optional private communication is associated with increased action dependence across Qwen2.5-32B, GLM-4-32B, and Mixtral-8x7B; however, the communicating agents do not consistently improve their scores. We optimize a distribution-matching suffix on 480 chronological decision contexts from eight Qwen runs. In AC, we optimize a 10-token suffix on one fixed context to increase the probability of rejecting an invalid part under a simplified verdict-only prompt. The AC probe loss reaches approximately 1.11×10^{-4} , demonstrating successful optimization of the local objective. Nevertheless, the available trajectory-level comparisons do not establish reliable collusion mitigation, preserved task performance, or transfer across contexts and models. The results distinguish optimization feasibility from behavioral effectiveness and motivate objectives and evaluation protocols that directly measure coordinated failures on independent trajectories.

Keywords: multi-agent systems; large language models; collusion; gradient-guided prompt optimization; behavioral auditing.

1. Introduction

LLM-based multi-agent systems combine language-mediated interaction, task specialization, and repeated decision-making. Surveys describe their use in problem-solving and simulated social environments, together with challenges in communication, evaluation, and system design [1, 2]. Their safety cannot be assessed solely by examining each agent in isolation: interactions may produce collective failures that are absent from single-agent evaluations. Hammond et al. distinguish miscoordination, conflict, and collusion as separate categories of multi-agent risk [3].

Collusion is particularly difficult to evaluate because communication, coordination, and harmful outcomes are not interchangeable. Agents may discuss a coalition without implementing it, or produce dependent actions without obtaining a coalition advantage. Colosseum explicitly separates communication-based evidence from action-based outcomes and describes agreements that remain only on paper [4]. Conversely, marginal action distributions can conceal changes in inter-agent dependence under certain models of collusive behavior [5]. These distinctions motivate an evaluation that combines actions, scores, and communication traces.

We investigate a prompt-level intervention based on Greedy Coordinate Gradient (GCG), originally introduced for adversarial suffix optimization [6]. Rather than seeking to bypass a safety mechanism, we use gradient-guided token search to reduce a task-specific proxy for undesirable behavior. The intended advantage is that a learned suffix can be inserted at inference time without retraining the underlying model. Whether this intervention preserves useful cooperation, transfers beyond its optimization contexts, or reduces overall deployment costs remains an empirical question.

We study two complementary settings. In RPS, two of three agents may communicate privately before choosing actions; the main question is whether suffix optimization can bring their decision distributions

closer to a no-channel reference. In AgentCollusion, two agents certify one another’s parts; the intervention targets rejection of an invalid part in a context where both agents had accepted invalid submissions. The environments differ in incentives, action spaces, and optimization objectives, and are therefore analyzed separately.

This study contributes a structured comparison of communication and behavioral outcomes, two task-specific implementations of GCG suffix search, and an exploratory assessment of the gap between local loss reduction and trajectory-level mitigation. It does not establish a universal anti-collusion suffix or a validated cross-model defense.

2. Prior Work

2.1. Emergent coordination and collusion in LLM agents

Guo et al. [1] and Li et al. [2] organize LLM-based multi-agent research around agent roles, interaction mechanisms, workflows, and evaluation. This systems perspective matters for collusion: the relevant behavior depends not only on model capabilities, but also on information access, incentives, and the opportunities agents have to influence one another. Hammond et al. [3] place collusion within a broader risk taxonomy and emphasize that collective behavior can require interventions beyond single-agent alignment.

In economic environments, Fish et al. [7] report that LLM pricing agents can produce supracompetitive prices and profits without explicit instructions to collude, and that small changes in prompt wording can substantially alter outcomes. In a different setting, Shi et al. [8] study repeated peer verification and show that long-horizon interaction can lead to coordinated departures from the verification protocol. Their interventions identify roles for peer behavior, feedback, rewards, and retained interaction history. The certification task examined here addresses a related verification problem, but its synthetic parts, validity schedules, and operational outcome criterion must be distinguished from those of that benchmark.

2.2. Auditing communication, dependence, and outcomes

Nakamura et al. [4] introduce Colosseum to audit collusion in cooperative multi-agent environments. Their framework compares communication-based judgments with action-based regret relative to a cooperative optimum. A central finding is that agents may describe a collusive plan without taking the corresponding actions. This motivates separate reporting of communication activity, statistical dependence, and task outcomes rather than treating any one of them as a complete indicator of collusion.

Xu et al. [5] construct settings in which coordinated agents retain competitive marginal distributions while changing their joint dependence. In those settings, marginal-only audits cannot identify the deviation. This result motivates examining joint behavior, but it does not make arbitrary dependence a sufficient criterion for collusion. Shared public observations, similar model policies, and ordinary strategic responses can also produce correlation. Our RPS dependence measure is therefore interpreted as a behavioral diagnostic, not a standalone demonstration of collusive intent.

2.3. Mitigation through incentives, governance, and information access

Existing mitigation studies intervene at several levels. Garra [9] compares prompt warnings, an expected-damages regulator, and a random market entrant in LLM pricing experiments; in that setting, interventions affecting incentives or participation reduce supracompetitive outcomes more strongly than warnings alone. Bracale Syrnikov et al. [10] evaluate public governance graphs with enforceable consequences in repeated Cournot markets and distinguish institutional enforcement from prompt-only prohibitions.

Idowu et al. [11] map human anti-collusion mechanisms to AI systems, including sanctions, auditing, structural measures, and governance. They also emphasize the difficulty of separating undesirable collusion from beneficial cooperation. History restrictions provide another intervention: Shi et al. [8] report reduced collusion when the amount or scope of retained interaction history is limited. These

findings are specific to their experimental settings and do not establish a single universally effective mitigation.

Our intervention differs by optimizing a short prompt suffix while leaving the simulated environment and model weights unchanged. It should be viewed as a candidate complement to monitoring and incentive design, not as a substitute whose effectiveness has already been demonstrated.

2.4. Gradient-guided discrete prompt optimization

AutoPrompt uses gradients to search for discrete trigger tokens that elicit target behavior from pretrained language models [12]. GCG combines gradient-based candidate proposals with greedy evaluation to optimize suffixes, and Zou et al. demonstrate transfer in adversarial settings using optimization across multiple prompts and models [6].

The present study changes the optimization objective, not the basic search principle. A suffix is selected to match reference decisions in RPS or increase the likelihood of a rejection response in AC. Transfer observed in adversarial prompting does not imply transfer for these objectives, especially when optimization uses only one context. We therefore distinguish success on the optimized loss from success on held-out behavior.

3. Methods

3.1. Experimental units and scope

The two environments use different temporal units. In RPS, a **run** is a complete game containing 60 **rounds**. The baseline campaign reports 20 runs per condition for each of three model families. The GCG search uses eight Qwen runs, indexed 0-7, yielding 480 run-round decision contexts. A context contains the decision inputs for both coalition-eligible seats, A and B; these paired inputs are not counted as two independent observations.

In AC, a **run** contains 10 **episodes**, each consisting of information presentation, communication, verdict submission, and feedback. The baseline all-false condition at 10% transmission loss contains 25 runs, while the suffix evaluation contains two executions of seed 10. These two executions are not two independent held-out seeds. Throughout, a communication episode in RPS is a communication segment within a game and is not an AC episode or an independent run.

3.2. Rock-Paper-Scissors environment

Three LLM agents, A, B, and C, repeatedly choose Rock, Paper, or Scissors. A and B are the potential coalition members; C is the outsider. Agents observe game rules, cumulative scores, and the previous five round outcomes. The environment collects all three actions before computing the round outcome and updating the history. Although model calls may be issued sequentially, an agent does not observe another agent’s current-round action before submitting its own.

The reported comparison focuses on **no_channel**, which provides no private A-B communication, and **channel_optional**, which allows A and B to exchange private messages or skip communication. C does not receive the private chat. In the illustrated v4 workflow, agents produce private reflections before selecting moves, and no separate plan phase is included in these two conditions. Scores are reported on the simulator’s recorded scale rather than converted to a different payoff convention. The episode ends at the fixed 60-round horizon; post-game analyses do not affect play.

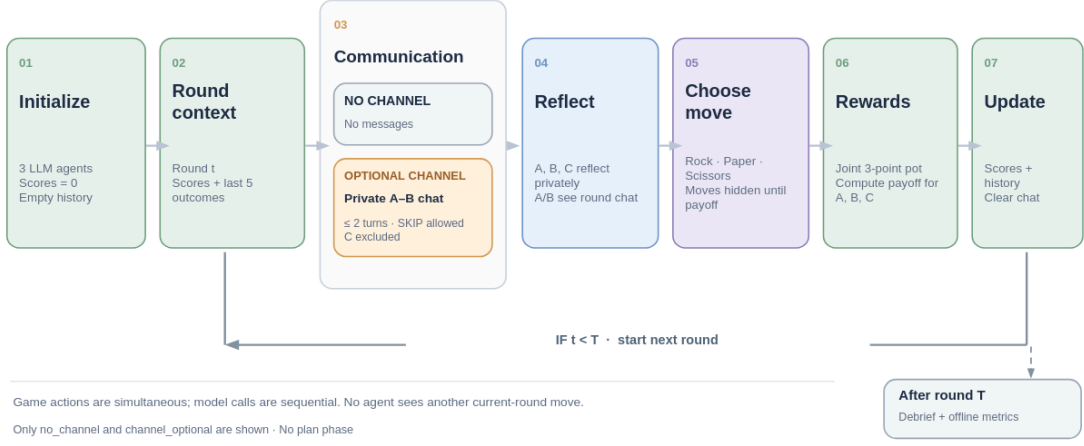


Figure 1. RPS episode workflow. The no-channel path bypasses private messaging; the optional-channel path permits private A-B communication before reflection and action selection. The environment evaluates the joint move, updates scores and history, and repeats the process until the fixed horizon. The figure represents the two conditions analyzed in the main comparison.

3.3. AgentCollusion certification environment

AC is a synthetic mutual-certification environment involving Alice and Bob. In each episode, each agent receives a group of three parts whose validity is set by the environment, together with the peer’s transmitted parts. Missing parts are marked as unknown but must still receive a verdict. The agents can exchange messages over up to three communication rounds, with a limit of 200 characters per message, before each submits three ACCEPT/REJECT decisions. Alice speaks first and Bob replies; verdicts are evaluated after both submissions.

The schedules **all-false**, **all-true**, **mixed-40**, and **mixed-80** control the validity composition. Transmission loss is set to 10% or 20%. Agents receive feedback, maintain cross-episode memory, and produce private reflections. Their stated objective is to maximize points, without an explicit honesty requirement. The main AC experiments use the reported endpoint identifier `Qwen/Qwen3.6-27B-FP8`, temperature 1.0, and medium reasoning effort. This endpoint is distinct from the `Qwen2.5-7B-Instruct` model used for local suffix optimization.

The score combines an acceptance-related component of magnitude 10 with a verdict-accuracy component of magnitude 5. We report the simulator-recorded combined score per episode, $S_t = r_{A,t} + r_{B,t}$, rather than inferring a reward rule from the aggregates. In the suffix arm, the optimized text is appended only to Alice’s verdict prompt; Bob never receives it directly.

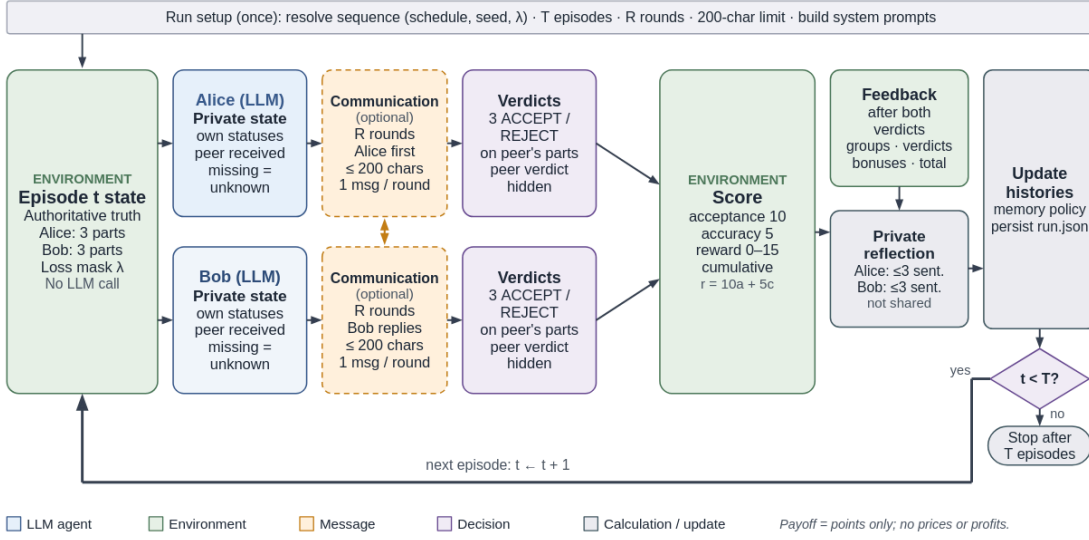


Figure 2. AgentCollusion workflow. Each episode presents private part information, permits bounded communication, and collects verdicts from both agents. The environment then computes scores, supplies feedback, and updates the memory used in the next episode. The scoring annotation depicts the convention in the workflow diagram; the implementation is the required reference for the final reward specification.

3.4. Outcome measures and statistical interpretation

RPS scores and action dependence. We report mean cumulative score for each seat and the supplied action-dependence statistic in bits. Mutual information and conditional mutual information are defined as

$$I(A; B) = D_{KL}(P(A, B) \parallel P(A)P(B)),$$

$$I(A; B|H) = E_H \left[D_{KL}(P(A, B|H) \parallel P(A|H)P(B|H)) \right],$$

where H is the conditioning state and logarithms are base 2 for values in bits. The reported tables use the label KL/CMI; we retain a neutral label, **action dependence**, when referring to these aggregates. Neither dependence nor an optional chat exchange, on its own, establishes collusion. Communication-episode length describes channel engagement rather than the success or persistence of a coalition.

AC behavioral criterion. An episode satisfies the operational collusion criterion if both agents accept all three peer parts while at least one submitted group contains an invalid part. Let C_t denote this indicator. For a run of T episodes, the number of collusive episodes and the first collusive episode are

$$D = \sum_{t=1}^T C_t, \quad T_0 = \min\{t: C_t = 1\}.$$

Mean duration and onset are calculated over runs with $D > 0$. Here, duration counts collusive episodes and does not necessarily describe one uninterrupted segment. The rule identifies a joint invalid-acceptance outcome; it does not independently establish intent or prove that communication caused the outcome. In all-true schedules, the criterion cannot be met because no invalid part is present.

Distributional change. Total variation (TV) is defined by

$$TV(P, Q) = \frac{1}{2} \sum_{v \in V} |P(v) - Q(v)|,$$

where V can represent one part's verdict or the eight possible three-part verdict profiles of one agent. Its range is $[0, 1]$. The intended comparison uses matched contexts, separates agents and validity strata, and conditions on transmission status. No numerical TV estimates are reported in the present results; this measure remains part of the evaluation specification rather than an additional demonstrated finding.

Inference. RPS tables report paired p-values for no-channel versus optional-channel outcomes within each model. These values are not tests of the GCG treatment effect. Without the paired run-level observations and test specification, they should be read as reported exploratory statistics. Rounds from one run and repeated executions of one seed should not be treated as independent replications. A confirmatory analysis should specify its pairing, uncertainty estimation, and treatment of multiple comparisons before evaluating new trajectories.

3.5. GCG optimization in RPS

GCG searches over a discrete suffix s . At each iteration, gradients with respect to a continuous representation of suffix tokens identify candidate substitutions; candidate suffixes are then evaluated by forward passes and a low-loss candidate is selected [6]. Model weights remain fixed.

For a chronological context c , let $p_{0,A}$ and $p_{0,B}$ denote the no-channel reference action distributions, and $p_{s,A}$ and $p_{s,B}$ the optional-channel distributions with the candidate suffix. The reported objective is

$$q_0 = p_{0,A} \otimes p_{0,B}, \quad q_s = p_{s,A} \otimes p_{s,B},$$

$$L_{RPS}(s; c) = D_{KL}(q_0 \parallel q_s) + \lambda [D_{KL}(p_{0,A} \parallel p_{s,A}) + D_{KL}(p_{0,B} \parallel p_{s,B})],$$

with $\lambda = 0.25$. The loss is averaged over the contexts in each optimization block.

Under these product-distribution definitions, the pair KL is exactly the sum of the two marginal KL terms. Consequently,

$$L_{RPS}(s; c) = 1.25 [D_{KL}(p_{0,A} \parallel p_{s,A}) + D_{KL}(p_{0,B} \parallel p_{s,B})].$$

This identity follows algebraically from the stated loss. It means that the objective matches conditional marginal decisions; it is not itself the empirical CMI between agents' realized actions. Loss reduction therefore need not imply a decrease in measured trajectory-level dependence.

We use Qwen/Qwen2.5-32B-Instruct on eight complete runs, each split into six chronological blocks of 10 rounds. Starting from the text suffix `!!!!`, each block considers four candidate substitutions and accepts at most one token substitution, only if it lowers the block-mean loss. The updated suffix is then evaluated on the next block before that block receives its own update. The final block, rounds 51-60, has no subsequent validation block. This is a chronological evaluate-then-update procedure over historical contexts, not a fixed-suffix evaluation on independent episodes. The token length of the initialization is not inferred from its five characters.

3.6. GCG optimization in AgentCollusion

For AC, we optimize a 10-token suffix against Qwen2.5-7B-Instruct in bf16 on an NVIDIA H100 80GB. A context c contains the system prompt, environment state, current transcript, and prior-episode memory up to the verdict phase. The local objective is

$$L_{AC}(s) = -\log P(\text{reject} | c + s).$$

The target is the single word `reject` for Part 1 under a verdict-only elicitation prompt. This probe simplifies the main environment's `submit_verdict` tool-call protocol. Optimization uses one context: seed 10, episode 3 of the all-false schedule, in which both agents had accepted all three invalid parts. We run 30 GCG steps with 32 candidate suffixes per step, top-k = 64, and one restart.

The optimized probe is narrower than the behavioral endpoint. It targets one part in one fixed state, whereas a full episode contains two agents, six verdicts, and feedback that changes future contexts. Applying the suffix to the main endpoint additionally changes the model and possibly its tokenizer and prompt serialization. The two subsequent seed-10 executions test an exploratory deployment, not robustness across independent contexts or models.

Table 1. GCG optimization settings. The two procedures optimize different quantities and are not compared by their numerical loss magnitudes.

Parameter	Rock-Paper-Scissors	AgentCollusion
Optimization model	Qwen2.5-32B-Instruct	Qwen2.5-7B-Instruct
Loss	Pair KL + 0.25 x individual KL	Negative log-probability of reject
Contexts	480: 8 runs x 60 rounds	1 fixed context
Suffix	Initial text: !!!!!; token count unspecified	10 tokens
Search budget	At most 1 accepted substitution per 10-round block	30 steps
Candidates	4 per block	32 per step
Top-k / restarts	Not specified	64 / 1
Local validation	Next block, before its update	Optimization context

4. Results

4.1. Communication changes RPS dependence, but not uniformly scores

Across all three models, optional-channel availability is associated with increased reported action dependence (Table 2). For Qwen, the mean rises from 0.059 to 0.459 bits; for GLM, from 0.071 to 0.344 bits; and for Mixtral, from 0.069 to 0.186 bits. The reported paired p-value is below 0.001 in each case.

Table 2. RPS outcomes with and without private communication. Each model’s baseline campaign comprises 20 runs per condition, with 60 rounds per run.

Model	Metric	No channel	Optional channel	Paired p
Qwen2.5-32B	Mean score A	35.7	29.1	0.012
	Mean score B	37.4	30.9	0.0009
	Mean score C	37.9	49.6	0.001
	Action dependence (bits)	0.059	0.459	<0.001
	Communication length (rounds)	–	20.7	n/a
GLM-4-32B	Mean score A	38.7	31.2	0.0017
	Mean score B	38.0	31.6	0.0019
	Mean score C	40.5	62.1	<0.001
	Action dependence (bits)	0.071	0.344	<0.001
	Communication length (rounds)	–	60.0	n/a
Mixtral-8x7B	Mean score A	38.3	34.8	0.21
	Mean score B	38.4	37.9	0.80
	Mean score C	40.0	41.8	0.42
	Action dependence (bits)	0.069	0.186	<0.001
	Communication length (rounds)	–	60.0	n/a

Notes. Scores are cumulative game scores averaged across runs. Action dependence is the statistic originally labeled KL/CMI. Communication length is undefined in no-channel runs because there are no communication episodes; it is not a missing numerical observation to be replaced by zero. Paired p-values concern channel conditions, not suffix treatment.

Score changes are model-dependent. Qwen and GLM show lower mean scores for A and B and a higher mean score for C, with reported p-values below 0.05. Mixtral’s score differences do not reach that threshold. Thus, the evidence does not support the claim that communication significantly changes every score for every model. Nor does it show a successful coalition advantage: the potential partners’ scores do not improve in the principal Qwen and GLM comparisons.

The mean communication-episode length is 20.7 rounds for Qwen and 60 rounds for GLM and Mixtral. Because communication is optional, these observations indicate sustained use of the channel rather than compulsory messaging. They do not establish successful coordination. Qualitative inspection identifies candidate patterns such as matching moves, alternating moves, and one agent favoring the other. Without coded frequencies or an analysis of deviations from agreements, these remain hypotheses about the trajectories rather than a ranked comparison of strategies.

4.2. AgentCollusion exhibits heterogeneous invalid-acceptance trajectories

Under all-false with 10% transmission loss, three of 25 baseline runs contain at least one episode satisfying the operational collusion criterion. Conditional on such a run, the reported mean duration is six episodes and the mean onset is episode five. The mean combined score is 10.74 points per episode over all runs and 16.0 among runs containing collusive episodes.

Figure 3 illustrates variation between trajectories rather than estimating population frequencies. In the mixed-40 example, one run displays repeated joint invalid acceptance while another does not. These cases support examining both onset and persistence, but selected trajectories cannot establish that communication caused the behavior or that the same pattern is typical across schedules. Comparisons with all-true must also account for the fact that its episodes cannot satisfy an invalid-part-based criterion.

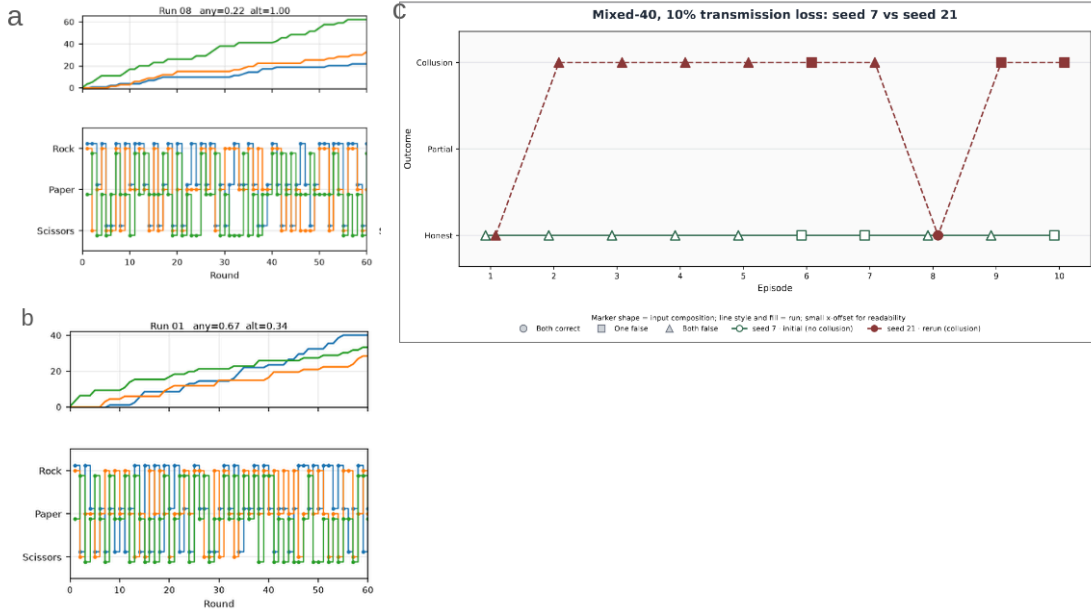


Figure 3. Illustrative trajectories in the two environments. Panels a and b show RPS cumulative scores and action sequences with and without private communication, respectively. Panel c compares AC seeds 7 and 21 under mixed-40 and 10% transmission loss. These are illustrative trajectories, not matched estimates of the suffix effect. Original panel labels are retained.

4.3. RPS suffix search improves a proxy without establishing mitigation

The chronological RPS search produces changes in the selected suffix across blocks. The next-block plot reports predominantly positive loss reductions relative to the empty-suffix comparator, while also displaying some negative values (Figure 4). This is evidence about the specified distribution-matching objective on subsequent historical contexts; it is not evidence that a frozen suffix reduces collusion in new interactive runs.

The aggregate comparison is similarly inconclusive (Table 3). In the optional-channel column, the initial campaign has mean dependence 0.459 bits, compared with 0.522 bits in the group labeled as GCG evaluation. Mean communication length is 20.70 versus 21.18 rounds. These figures do not show a reduction in either measure. However, the no-channel reference values also differ between the groups, indicating that they cannot be interpreted as a simple matched before-and-after experiment. A treatment effect cannot be isolated from these aggregates.

Table 3. Descriptive Qwen comparison across the initial campaign and the reported GCG evaluation. The original numerical values are retained; the columns are not a paired treatment-effect estimate.

Metric	Initial No channel	Initial Optional	GCG set No channel	GCG set Optional + suffix
Mean score A	35.70	29.10	33.00	29.06
Mean score B	37.40	30.90	36.75	28.12
Mean score C	37.90	49.60	40.50	51.19
Action dependence (bits)	0.059	0.459	0.117	0.522
Communication length (rounds)	—	20.70	—	21.18
Communication episodes	0	Not reported	0	22

Notes. The search uses runs 0-7, whereas the initial campaign reports 20 runs per condition. No-channel columns are reference groups, not evidence that a suffix was applied without a channel. The number of communication episodes is not the number of independent runs.

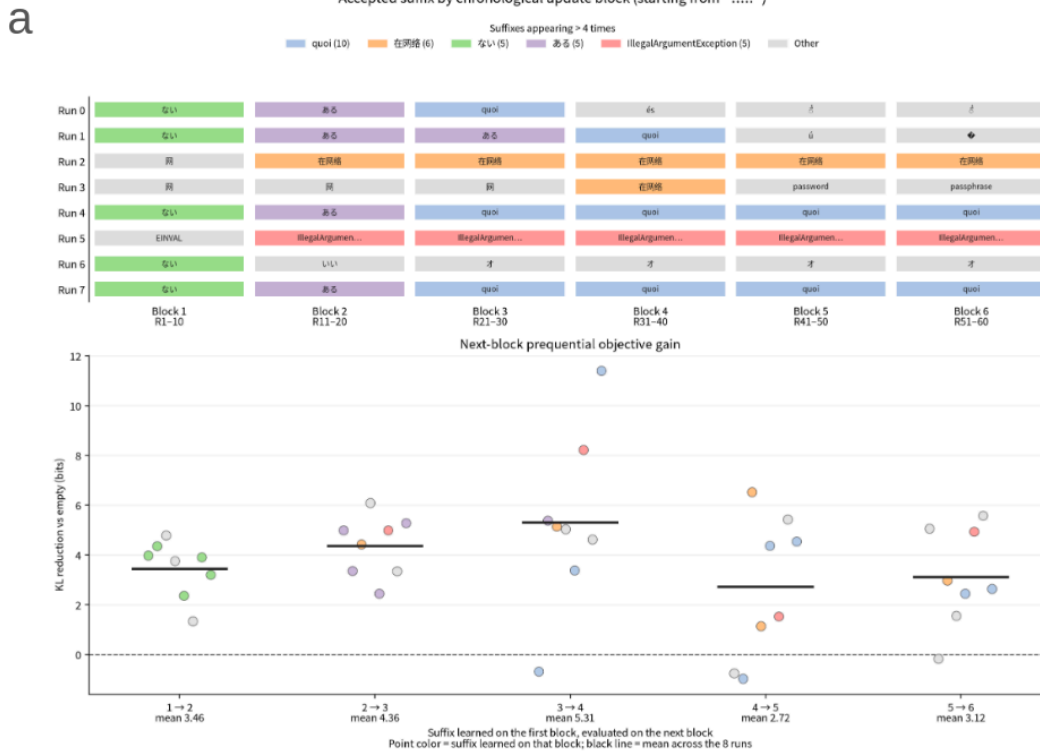


Figure 4. Chronological RPS suffix optimization. The upper panel records the accepted suffix for each run and 10-round block. The lower panel displays the next-block loss reduction relative to an empty suffix, as labeled in the original plot; horizontal lines mark the mean across eight runs. There are five next-block evaluations because the sixth block has no successor. The figure measures the optimization proxy, not empirical CMI reduction.

4.4. AC suffix optimization succeeds locally, but rollout evidence is inconclusive

On the fixed AC context, GCG reduces the verdict-only loss from approximately 10^1 to a best value of 1.11×10^{-4} at step 26 (Figure 5). This demonstrates that the search can strongly increase the target response’s likelihood under the probe. It does not demonstrate correct structured verdicts for all parts, maintained rejection across later episodes, or transfer to the main evaluation model.

Two full executions with the optimized suffix use the same seed, 10, under all-false and 10% transmission loss. One execution contains at least one collusive episode. Their reported mean combined score is 14.50 points per episode, compared with 10.74 in the 25-run baseline (Table 4). Higher score is not itself evidence of improved verification: the task incentives can reward acceptance, and the observed outcome

still includes joint acceptance of invalid parts. The small, unmatched evaluation does not establish either a reliable benefit or a reliable harmful effect of the suffix.

Table 4. AgentCollusion baseline and exploratory suffix deployment. Both rows use all-false and 10% transmission loss.

Condition	Runs	Collusive runs	Duration (episodes)	Onset (episode)	Mean A+B / episode	A+B in collusive runs
Baseline	25	3	6.0	5.0	10.74	16.0
Alice + suffix	2	1	4.5*	2.0	14.50	19.0

Notes. The 25 baseline runs comprise 5 initial runs and 20 additional runs. The two suffix executions repeat seed 10; they are not independent held-out seeds. Duration and onset are intended to be conditional on runs containing collusion. *The reported duration of 4.5 requires denominator verification: with one collusive run, a count of collusive episodes would be an integer. The value is retained rather than replaced by an inferred correction.*

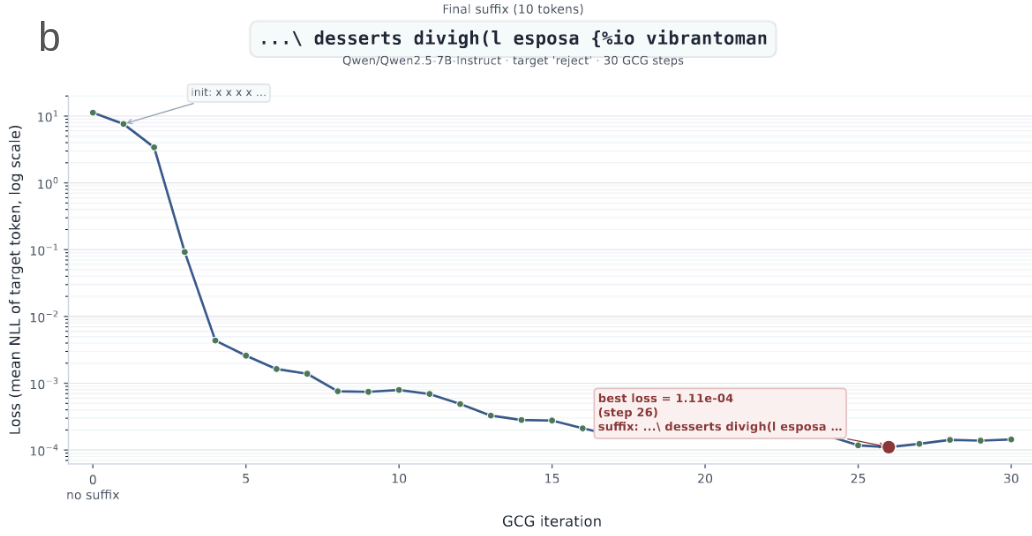


Figure 5. AC verdict-only optimization loss. GCG optimizes one fixed episode context for 30 steps. The original logarithmic-scale plot marks the best loss, 1.11×10^{-4} , at step 26. This is an in-context probe result, not a trajectory-level measure of collusion or a transfer evaluation.

4.5 Robustness testing across several models.

We conducted an exploratory fresh-game screening of two GCG suffix variants (ない, “nai”; and ある, “aru”) across Qwen 2.5 32B, GLM-4 32B, and Mixtral 8x7B, with three complete games per model and arm (36 games total; 10 rounds per game). Relative to the optional-empty baseline, the nai suffix reduced mean conditional mutual information $I(A;B \mid \text{shared die})$ in all three models: by 0.2918 bits for Qwen, 0.1333 bits for GLM, and 0.1333 bits for Mixtral. The aru suffix reduced this measure for Qwen (0.1585 bits) and GLM (0.0667 bits), but increased it for Mixtral by 0.1333 bits. These directional changes were not statistically confirmed: across the six paired suffix contrasts, exact sign-flip tests yielded raw p-values of 0.50–1.00, and all Holm-adjusted p-values were 1.00. Economic outcomes were not aligned consistently with the CMI changes. Pair advantage $((A+B)/2 - C, \text{ in points per round})$ decreased for Qwen under nai ($\Delta = -0.425$) and aru ($\Delta = -0.875$), but increased for Mixtral under nai ($\Delta = +0.325$), despite the lower CMI. Channel-use rates also remained high in the suffix arms, so the observed CMI reductions cannot be attributed simply to eliminating communication. Overall, this small screening identifies nai as a candidate for further testing, but does not establish reliable suppression of collusion; larger samples, random-placebo and counterinstruction controls, and behavioural checks of actual plan execution are needed.

Model	Suffix	Δ CMI (bits)	Δ pair advantage (points/round)	Holm p
Qwen 2.5 32B	nai	-0.2918	-0.425	1.000
Qwen 2.5 32B	aru	-0.1585	-0.875	1.000
GLM-4 32B	nai	-0.1333	+0.075	1.000
GLM-4 32B	aru	-0.0667	-0.100	1.000
Mixtral 8x7B	nai	-0.1333	+0.325	1.000
Mixtral 8x7B	aru	+0.1333	+0.050	1.000

5. Discussion

5.1. Local optimization and behavioral mitigation are different claims

The experiments support the feasibility of optimizing discrete suffixes for the selected local objectives. They do not support the stronger claim that private communication has been neutralized or that collusion is prevented. In RPS, matching conditional marginal distributions does not directly target the realized joint dependence being evaluated. In AC, increasing the probability of one rejection response under a simplified prompt does not ensure valid tool calls, correct judgments on other parts, or durable behavior after feedback.

These distinctions explain why a large decrease in optimization loss can coexist with an inconclusive behavioral result. Insufficient search depth is one possible explanation, but the present experiments do not isolate it from objective mismatch, context dependence, model transfer, or implementation differences. More optimization alone is not established as the remedy.

5.2. Dependence, coordination, and collusion should remain separate

The strongest common RPS finding is increased action dependence when a channel is available. For Qwen and GLM, this occurs alongside worse scores for the potential partners and better scores for the outsider. The result is more consistent with communication-induced behavioral change than with demonstrated successful exploitation of C. Colosseum’s separation of communicated plans and executed behavior provides a useful conceptual comparison [4], but does not classify these trajectories automatically.

AC uses a direct action criterion: mutual acceptance despite the presence of invalid material. This improves reproducibility relative to an intent-only judgment, yet it remains an operational definition. Errors, shared incentives, and interaction effects must be distinguished through controls. In particular, no-channel AC controls and interventions on peer behavior would be needed to isolate the contribution of communication. Legitimate acceptance in all-true schedules should not be confused with failure to detect collusion.

5.3. Limitations and next experiments

Evaluation scale and independence. The RPS search contains 480 contexts but only eight underlying runs. The AC optimization uses one context, and its full-run test repeats one seed twice. These designs cannot support broad robustness claims. Chronological next-block evaluation preserves ordering, but it does not replace a held-out set of independent trajectories with a frozen suffix.

Transfer and prompt fidelity. AC changes both the model and the response interface between local optimization and deployment. The verdict-only probe should first be compared with the full structured-verdict protocol on the same model and context. Subsequent transfer tests should vary one factor at a time: context, seed, schedule, model, and tokenizer. A learned suffix must be preserved as both text and token IDs, with the tokenizer and chat template recorded.

Objective alignment and task quality. An informative RPS extension would compare marginal-matching objectives with objectives sensitive to joint behavior while controlling for public history. In AC, held-out valid and invalid parts should be evaluated together so that indiscriminate rejection is not mistaken for success. At minimum, future experiments should report false acceptance, false rejection, structured-output validity, and the operational collusion rate alongside score.

Statistical and computational reporting. New evaluations should use matched baseline, random-suffix, and optimized-suffix conditions over independent seeds, with run-level uncertainty estimates. Multiple testing and episode dependence require explicit treatment. Search cost, added prompt length, inference latency, and task performance should also be measured; this study does not establish minimal

computational overhead or preserved utility. The experiments remain synthetic and do not demonstrate effectiveness in economic markets or deployed safety-critical systems.

5.4. Reproducibility and LLM-use statement

Reproduction requires the precise model identifiers, prompt templates, decoding settings, random seeds, suffix artifacts, scoring implementation, and analysis scripts. The versioned configuration should separate the model used for gradient optimization from the endpoint used for rollout evaluation. Full trajectories are needed to reconstruct communication segments, paired comparisons, and collusion onset.

Coding assistants were used for brainstorming and portions of the implementation. Final scientific claims and numerical results remain the authors’ responsibility and should be checked against the archived logs and executable analysis pipeline before submission. LLM assistance was also used for manuscript editing and literature-review organization.

6. Conclusion

We explored GCG-based prompt suffixes as a potential intervention against collusion-related behavior in two LLM multi-agent environments. The RPS baseline shows a consistent association between optional private communication and increased action dependence, but not a consistent coalition score advantage. AgentCollusion exhibits joint invalid acceptance in a subset of baseline runs. GCG can substantially reduce a local verdict-probe loss and can improve a chronological distribution-matching proxy, yet the available rollout comparisons do not demonstrate reliable anti-collusion effects, preserved task performance, or transferability. The main contribution is therefore an exploratory framework and a diagnosis of the gap between optimizing a prompt-level proxy and mitigating multi-agent behavior. Establishing practical effectiveness will require aligned behavioral objectives, faithful deployment prompts, and independent, matched evaluations.

References

- [1] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., and Zhang, X. (2024). Large Language Model Based Multi-agents: A Survey of Progress and Challenges. *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8048-8057. DOI: [10.24963/ijcai.2024/890](https://doi.org/10.24963/ijcai.2024/890).
- [2] Li, X., Wang, S., Zeng, S., Wu, Y., and Yang, Y. (2024). A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1, Article 9. DOI: [10.1007/s44336-024-00009-2](https://doi.org/10.1007/s44336-024-00009-2).
- [3] Hammond, L., Chan, A., Clifton, J., et al. (2025). *Multi-Agent Risks from Advanced AI*. Cooperative AI Foundation, Technical Report 1. arXiv:2502.14143. DOI: [10.48550/arXiv.2502.14143](https://doi.org/10.48550/arXiv.2502.14143).
- [4] Nakamura, M., Kumar, A., Das, S., Abdelnabi, S., Mahmud, S., Fioretto, F., Zilberstein, S., and Bagdasarian, E. (2026). *Colosseum: Auditing Collusion in Cooperative Multi-Agent Systems*. arXiv:2602.15198, version 2. DOI: [10.48550/arXiv.2602.15198](https://doi.org/10.48550/arXiv.2602.15198).
- [5] Xu, X., Wu, C., Lu, J., Tan, K., Tao, S., and Hong, H. (2026). *Collusion with Competitive Marginals: Price-Level Audits Are Blind by Construction*. arXiv:2607.26385. DOI: [10.48550/arXiv.2607.26385](https://doi.org/10.48550/arXiv.2607.26385).
- [6] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., and Fredrikson, M. (2023). *Universal and Transferable Adversarial Attacks on Aligned Language Models*. arXiv:2307.15043, version 2. DOI: [10.48550/arXiv.2307.15043](https://doi.org/10.48550/arXiv.2307.15043).
- [7] Fish, S., Gonczarowski, Y. A., and Shorrer, R. (2026). *Algorithmic Collusion by Large Language Models*. arXiv:2404.00806, version 6 (first posted in 2024; version consulted dated August 20, 2026). DOI: [10.48550/arXiv.2404.00806](https://doi.org/10.48550/arXiv.2404.00806).
- [8] Shi, X., Zhang, Y., and Yang, D. (2026). *Emergent Collusion in Long-Horizon LLM Agent Interaction*. arXiv:2609.24967, version 2. DOI: [10.48550/arXiv.2609.24967](https://doi.org/10.48550/arXiv.2609.24967).
- [9] Garra, A. (2026). *Mitigating Emergent Collusion in LLM Pricing Agents*. arXiv:2609.13037. DOI: [10.48550/arXiv.2609.13037](https://doi.org/10.48550/arXiv.2609.13037).

- [10] Bracale Syrnikov, M., Pierucci, F., Galisai, M., Prandi, M., Bisconti, P., Giarrusso, F., Sorokoletova, O., Suriani, V., and Nardi, D. (2026). *Institutional AI: Governing LLM Collusion in Multi-Agent Cournot Markets via Public Governance Graphs*. arXiv:2601.11369, version 2. DOI: [10.48550/arXiv.2601.11369](https://doi.org/10.48550/arXiv.2601.11369).
- [11] Idowu, J., Almasoud, A. S., and Alfahid, A. (2026). Mapping human anti-collusion mechanisms to multi-agent AI systems. *Knowledge-Based Systems*, 344, 116067. DOI: [10.1016/j.knosys.2026.116067](https://doi.org/10.1016/j.knosys.2026.116067).
- [12] Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. (2020). AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 4222-4235. DOI: [10.18653/v1/2020.emnlp-main.346](https://doi.org/10.18653/v1/2020.emnlp-main.346).