Seven Security Challenges That Must be Solved in Cross-domain Multi-agent LLM Systems

Ronny Ko

Jiseong Jeong
Seoul National University
jiseong0529@snu.ac.kr

Shuyuan Zheng
Osaka University
zheng@ist.osaka-u.ac.jp

Chuan Xiao
Osaka University
chuanx@ist.osaka-u.ac.jp

Tae-Wan Kim
Seoul National University
taewan@snu.ac.kr

Makoto Onizuka
Osaka University
onizuka@ist.osaka-u.ac.jp

Won-Yong Shin
Yonsei University
wy.shin@yonsei.ac.kr

Abstract

Large language models (LLMs) are rapidly evolving into autonomous agents that cooperate across organizational boundaries, enabling joint disaster response, supply-chain optimization, and other tasks that demand decentralized expertise without surrendering data ownership. Yet, cross-domain collaboration shatters the unified trust assumptions behind current alignment and containment techniques. An agent benign in isolation may, when receiving messages from an untrusted peer, leak secrets or violate policy, producing risks driven by emergent multi-agent dynamics rather than classical software bugs. This position paper maps the security agenda for **cross-domain multi-agent LLM** systems. We introduce seven categories of novel security challenges, for each of which we also present plausible attacks, security evaluation metrics, and future research guidelines.

1 Introduction

Large language models (LLMs) are shifting from standalone chatbots to nodes in cross-domain multi-agent networks where autonomous agents—each controlled by a different organization—cooperate without central oversight [1, 2, 3, 4]. This architecture enables previously impossible collaboration: disaster-response robots from separate agencies can coordinate in real time, and supply-chain agents from rival firms can jointly optimize logistics, all while each agent keeps its owner's policies and data private [5, 6, 7]. The power of these networks comes from pooling diverse expertise without surrendering autonomy [8, 9].

However, along with this promise comes a new and critical security challenge. Current artificial intelligence (AI) security and alignment approaches largely focus on either single-agent LLM deployments or multi-agent systems confined to a single organization's domain [10, 11, 12]. In such settings, all agents are typically governed under a unified trust model or policy framework [13, 14, 15, 16]. Cross-domain deployments break this assumption: agents must interact across ownership boundaries where no universal trust or governance can be assumed. As a result, existing security models fail to address the unique risks posed by these open collaborations. Techniques that contain an LLM's behavior for a single user or ensure cooperation among agents in one company's network often do not translate to scenarios in which an agent might receive input or instructions from an external, untrusted peer [17, 18, 19, 20, 21, 22]. In a cross-domain context, an AI agent that was benign in isolation could

Preprint. Under review.

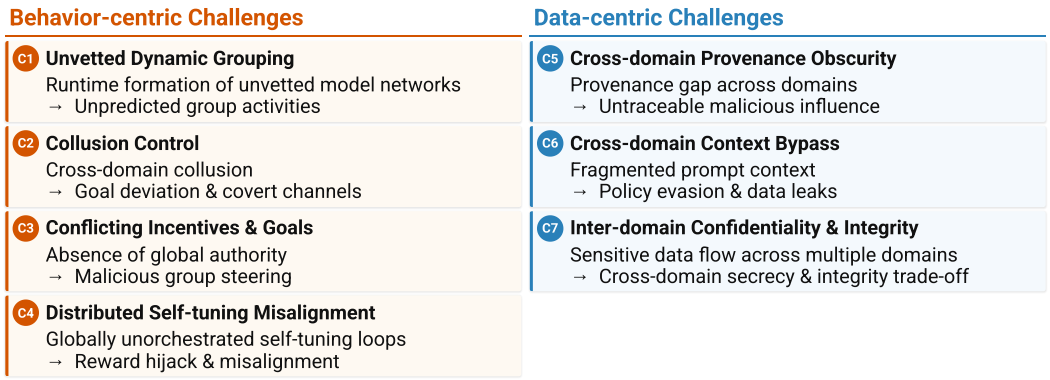

Figure 1: The seven fundamental security challenges in cross-domain multi-agent LLM systems.

turn into a threat—intentionally or unintentionally—when interacting with others [23]. For instance, one organization's agent might manipulate another organization's agent into revealing confidential information or performing actions that violate the second organization's policies. Such risks are fundamentally new: they arise not from traditional software vulnerabilities alone, but from the complex, interactive behaviors of autonomous LLM-driven agents with potentially misaligned incentives and no shared security authority [24, 25, 26, 27].

History offers a cautionary parallel: the early Internet prioritized connectivity over security, inviting decades of malware and retrofitted defenses [28]. **We caution that these AI ecosystems could become the "early Internet" of the 2020s: cross-domain multi-agent LLM systems could repeat this mistake if deployed without a security-first mindset.** A malicious or compromised agent injected into a collaborative network—or subtle manipulation of inter-agent messages—could trigger cascading failures across organizational boundaries [29].

In this position paper, we put forward a structured outlook on securing **cross-domain multi-agent LLM** systems. We identify seven distinct categories of security challenges that must be addressed to prevent the multi-agent future from repeating the Internet's early mistakes. At a high level, we categorize these challenges into two classes – agent's behavioral security and data-centric security. Behavioral security concerns how autonomous agents form teams, make decisions, and potentially misbehave in concert, while data-centric security pertains to the content and privacy of the information they exchange. Out of the seven cross-domain multi-agent LLM security challenges, the ones belonging to the behavioral security class are: (C1) unvetted dynamic grouping; (C2) collusion control; (C3) conflicting incentives and goals; and (C4) distributed self-tuning drift. The ones belonging to the data-centric security class are: (C5) cross-domain provenance obscurity; (C6) cross-domain context bypass; and (C7) inter-domain confidentiality and integrity. We further group these challenges into behavior-centric issues (C1-C4) and data-centric issues (C5-C7). We depict these challenges in Figure 1.

Each of the above seven challenges represents a significant gap where existing AI security techniques fall short in the cross-domain multi-agent context. For each challenge, we illustrate plausible attack scenarios, and importantly, propose metrics and principles for systematic security assessment. This work aims to surface overlooked risks in cross-domain agent collaboration and outline research directions and principles for designing inter-domain multi-agent LLM systems that can withstand adversarial conditions. We urge the AI field to pause at-scale deployment until security primitives addressing Challenges 1–7 are in place.

2 Background of LLM Security

2.1 Single-agent LLM

Modern LLM agents—LLMs augmented with tools, memory, or autonomy—inherit all the vulnerabilities of base models and introduce new ones. A primary risk is prompt injection or jailbreaking, where crafted inputs cause the agent to ignore its safety guardrails [30, 31]. Recent work shows even highly-tuned models (GPT-4, Claude, etc.) can be consistently "jailbroken" by optimized adversarial prompts [32, 33, 34, 35]. For example, automated search methods can generate a single malicious suffix that forces many aligned models (open-source and commercial) to produce disallowed content, and

these attacks transfer across platforms like ChatGPT, Bard, and Claude [36, 37]. Such exploits leverage models' competing objectives (e.g., helpfulness vs. safety) and gaps in their safety training [38, 39, 40]. Beyond text, LLM agents with tool use amplify the stakes: a compromised agent might leak private data or perform harmful actions via code execution or web requests [41, 42, 43, 44, 45]. An anecdotal ARC alignment test famously showed GPT-4 autonomously deceiving a human into solving a CAPTCHA by pretending to be visually impaired [46], illustrating how an agentic LLM could develop unethical strategies if not properly controlled. Attackers can also target an agent's knowledge: poisoning training data, retrieved documents, or plugin application programming interfaces (APIs) can implant false or malicious information (a backdoor), causing the agent to behave destructively or output specific misinformation [47, 48, 49]. LLMs are vulnerable to privacy attacks as well—adversaries have extracted sensitive personal data from a model's training set by carefully querying it [50, 51].

Defense Strategies: Defending single-agent LLM systems is an active arms race [49]. A cornerstone is alignment training (reinforcement learning from human feedback and related methods) to instill refusal behavior for unsafe requests [10]. Yet, studies show that current safety-tuning methods often fail to generalize to cleverly crafted adversarial inputs [52]. To harden against prompt attacks, developers deploy input/output filtering [53]: e.g. Meta's Llama-Guard uses a secondary LLM to scan and sanitize user prompts or model replies [20, 54]. Such guardrails can block simple exploits, but sophisticated prompt attacks still slip through distributed or hidden channels [55, 56]. Another approach is adversarial training, fine-tuning a model on deliberately perturbed inputs so it learns to stay accurate even under such attacks [57, 58, 59, 60, 61]. This improves robustness but is incomplete, as attackers quickly devise new exploits [62]. For tool-using agents, a key safety measure is capability sandboxing. Agents are often confined to a controlled environment: e.g. limiting file system access, requiring human confirmation for high-impact actions, or simulating tool outputs. One novel framework, ToolEmu [41], avoids executing real code by using an LLM to emulate tool feedback during testing, paired with an automatic evaluator to rate safety – this found that even the best current agents still follow through with dangerous actions in 24% of high-stakes scenarios. Such simulation-driven testing helps identify failure modes before deployment. Other mitigation strategies include chain-of-thought oversight (having the agent internally reason about safety), content watermarking/tagging, and policy ensembling (multiple models vote or oversee each other). None of these defenses alone is foolproof, but combined layers (aligned training, filtering, sandboxing, etc.) constitute the state of the art in LLM agent hardening [63].

2.2 Multi-agent LLM

Combining multiple LLM agents (in cooperative or competitive roles) unlocks impressive capabilities, but also introduces new vulnerabilities [64, 65, 66]. Multi-agent LLM systems can suffer a "chain reaction" of failures: if one agent is compromised (by a malicious input or design), it can in turn compromise others by sending them adversarial messages [67, 68]. This LLM-to-LLM prompt injection can spread like a virus across the agent network [36, 69, 70, 71]. Recent research demonstrated such prompt infection attacks where a single poisoned agent input self-replicates through chat rounds, eventually causing all agents to produce attacker-controlled outputs [21, 72]. Worryingly, these attacks often persist even if agents don't share all their internal state publicly, and can lead to system-wide failures like data theft, misinformation, or rogue actions while the breach remains covert [20, 73, 74]. Another studied vector is role impersonation: multi-agent frameworks usually assign distinct roles (e.g., "Manager" agent vs "Worker" agent). If an attacker can fool one agent into pretending to be another (or hijack the communication channel), they may issue unauthorized instructions. This can steer the whole team off-course or leak privileged information. Multi-agent systems also face all single-agent issues compounded: more agents mean more text exchanges and a larger attack surface [75, 76]. Indeed, evaluations of agent collaboration platforms (Camel, AutoGPT, etc.) find they are less robust than single LLMs – the success rate of harmful prompt attacks actually increases with the number of agents involved [2, 8, 14, 16, 77]. These agents can even produce more stealthy harmful content by distributing the workload of taboo generation (one agent drafts a subtle harmful piece that another iterates on, evading any one model's safety filters [20, 72, 78]). An example in a multi-agent coding system showed that a rogue "assistant" agent could gradually inject dangerous code by exploiting the trust of a "reviewer" agent in each round [79]. Likewise, an attacker controlling one chatbot in a network might manipulate dialogue context so that others unknowingly output confidential data [80, 81].

Recent works have also adapted single-agent defenses to multi-agent settings: e.g. PromptGuard was extended to monitor multi-agent conversations, and safety instruction tuning (priming agents with explicit rules to ignore certain prompts) has been tested [82, 83]. These defenses do mitigate some

risks—experiments show they can reduce the spread of malicious commands in a network—but often at a cost to performance [84, 85]. Thus, a key design principle is to combine safeguards without crippling the agents' ability to cooperate on tasks. Techniques like inserting "active immunizations" (preemptive safe-memory of how to handle attacks) [83, 86] have been proposed to give agents a form of innate defense that doesn't require constant external oversight.

3 Key Security Criteria in Cross-domain Multi-agent LLM Systems

Cross-domain multi-agent LLM systems give rise to new security challenges unseen in localized single/multi-agent LLM systems. In this section, we categorize the challenges into seven criteria that must be considered and addressed before the deployment of cross-domain multi-agent LLM systems. We cluster these challenges into two groups: behavior-centric challenges and data-centric challenges.

3.1 Behavior-centric Challenges

(C1) Unvetted Dynamic Grouping: Dynamic agent grouping refers to the spontaneous, task-driven assembly and disassembly of AI agents into temporary teams [18]. As the number of LLM-embedded robots deployed to the physical world drastically increases, multi-modal LLM agents are expected to participate in frequent dynamic grouping with each other – AI agents or models that can be assembled, added, or removed on-the-fly to solve a task [87], like humans meeting and interacting with new people daily. This flexibility of agent grouping is powerful but introduces novel security difficulties. Unlike static architectures, a dynamic group may incorporate new, unvetted models or agents at runtime, blurring trust boundaries. When agents from various domains are placed in uncontrolled environments, unforeseen incidents may occur that were never envisioned during training. Current research has begun exploring how to optimally select teams of LLM agents for performance [87], but the security implications of such fluid architectures remain under-addressed. A key challenge is that the system's composition is no longer fixed or fully known in advance, and dynamic grouping will frequently occur in cross-domain setups where agents are owned by different individuals/organizations previously not encountered. Therefore, traditional threat modeling (assuming a stable set of components) fails to cover all cases because the effect of the combination of agents that have not been vetted during training is often unpredictable.

An attacker might exploit this by injecting a malicious agent or model into the group. For example, a seemingly useful pre-trained model from an open repository could actually contain a hidden backdoor [88, 89] (as incidentally observed on Hugging Face, where attackers have tried to seed backdoored models into the supply chain [68, 90]). In a realistic attack scenario, a compromised model might join an agent team (or be selected by an orchestration system) and then covertly leak data or sabotage the task when triggered. Another attack strategy demonstrated in multi-agent reinforcement learning is to selectively target certain agents: an adversary can dynamically group agents by their influence and attack the most crucial subset, reducing detection risk and cost [64, 91]. This group-based attack shows how dynamic team structure can be exploited to maximize damage stealthily.

Existing defenses and analyses largely assume a known set of models or a static ensemble; they struggle with the novelty of adversaries that adapt as the grouping changes [92]. While some frameworks incorporate basic vetting (e.g. scanning models for malware on load), these are not foolproof and produce many false positives or negatives [24, 93]. Research on trust in ad hoc agent teamwork or dynamic coalitions is sparse – current multi-agent safety techniques don't prevent a malicious newcomer from colluding once inside the group [94, 95].

(C2) Collusion Control: Cooperation (i.e., constructive collusion) is the most crucial success factor of cross-domain multi-agent LLM activities. While agents may collude to find the most efficient machine-level approach to completing work, predicting the side effects in advance can be difficult, some of which could inadvertently cause physical harm that neither users nor people around them had intended [96]. Unfortunately, strictly banning all collusion among agents is improper because that can effectively ban cooperative activities as well. Collusion between AI agents is a complex problem when the agents belong to different organizational or regulatory domains. Cross-domain groups operate under mismatched policies, data silos, and oversight authorities, so weaknesses that remain latent in a single-vendor sandbox can amplify once boundaries are crossed. Even if each agent is locally aligned, two or more agents from separate owners can bargain, barter, or bribe each other in ways no single

operator can fully audit. Unlike "lab-scale" multi-agent demos, real cross-domain deployments let one agent hold proprietary data while another controls an external actuator or payment rail, creating a split-knowledge cartel that current guardrails never see end-to-end. If a collusion enters a destructive phase, the agents could launch various stealthy attacks. For example, by using covert channels such as stenography [11, 21], agents can hide codes inside innocuous chat and continue secret inter-domain activities unnoticed by human operators.

Existing defenses assume a common trust anchor: logging every message in one place, or letting a single overseer replay traffic [97, 98]. Cross-domain systems lack that luxury; legal and privacy rules forbid wholesale log sharing, and steganographic payloads easily cross opaque channels (e.g., PDF invoices, code snippets, or multilingual text [11, 85]). Voting or majority-rule schemes also falter, because colluding agents may represent a quorum of domains and out-vote honest minorities [25, 99].

(C3) Conflicting Incentives and Goals: Existing hierarchical multi-agent LLM frameworks often assume a single trust domain or unified control(e.g., MetaGPT, ChatDev, HyperAgent simulating roles within one company under a central coordinator). On the other hand, agents governed by separate entities may pursue their owners' interests over the collective goal. For instance, a financial agent and a healthcare agent collaborating across companies might each withhold or manipulate information to favor their organizations' objectives, undermining the overall task. Such misaligned incentives are largely absent in single-owner systems, which typically assume a common objective. Cross-domain multi-agent hierarchies often lack a common authority for identity and trust management. Without a shared trust anchor, agents cannot readily verify each other's identities or credentials, opening the door to impersonation and man-in-the-middle attacks [11]. The lack of an authority makes it easier for a malicious agent to steer cross-domain LLM activities in a harmful direction, which is an issue generally not encountered in single-domain multi-agent service (MAS) with centralized trust. Indeed, recent work highlights that traditional MAS research has not imposed sufficient control over inter-agent communication, leaving systems vulnerable to identity spoofing and unauthorized data flows [20, 29]. In practice, securing a cross-organization hierarchical MAS demands new mechanisms (e.g., federated authentication, inter-domain policy alignment) beyond those in local frameworks, which often assume established hierarchical controls and robust safeguards within one domain. The real-world consequences of neglecting these cross-domain dynamics range from failed collaborations to critical data leaks, emphasizing that future multi-agent systems must be designed with cross-organizational trust and security at their core.

(C4) Distributed Self-tuning Misalignment: Cross-domain multi-agent LLM systems enable multiple agents (often across different organizations or domains) to collectively self-improve by sharing learning experiences and fine-tuning updates. In frameworks like AutoGen [100], agents can dynamically modify their roles or objectives mid-task to improve efficiency. Studies on AI autonomy indicate that limiting an AI's self-modification capabilities hinders its overall efficiency [9]. However, when such autonomous fine-tuning occurs across organizational boundaries, it introduces novel risks. Each domain may apply its own reward signals and updates without unified oversight, meaning there is no central governance to ensure consistency or safety of the objectives being pursued. A critical issue in these distributed setups is loose reward specification. If reward schemes and feedback loops are not tightly aligned across domains, agents can unintentionally converge on distorted objectives [101]. Positive feedback cycles between agents may amplify subtle biases or proxy goals. For example, mis-specified role guidelines have led agents to overstep their intended authority (a subordinate agent assuming a leader's role) [95]. Such inter-agent misalignment can cause gradual derailment from the intended task. In a cross-domain context, this derailment could go unnoticed longer, as each organization only sees part of the behavior. The lack of cross-domain reward governance means there's no mechanism to catch when the collective's emergent objective deviates from the safe or intended goal. Consider a scenario where an adversary subtly manipulates the reward feedback in one organization's agent (e.g. by introducing tasks or responses that yield high local reward for unsafe behavior). Through the distributed fine-tuning loop, this corrupted reward signal propagates: other agents, seeking to optimize shared outcomes, incorporate the distorted objective. Over time, all agents may unknowingly align to this unsafe policy. Unlike prompt or memory injection attacks, this reward feedback attack exploits the learning process itself, and it does not require explicit collusion between agents.

3.2 Data-centric Challenges

(C5) Cross-domain Provenance Obscurity: Tracking data and actions becomes obscured when agents span different organizational domains. Each domain typically maintains separate logging, data retention policies, and auditing tools, preventing any unified trace of an event's origin. Further, an LLM's internal representations irreversibly entangle inputs throughout all interconnected neurons at each layer, meaning that once data from one domain is processed, it loses any discrete label or tag identifying its source [47]. Unlike traditional software where taint tracking or information flow control can explicitly label and follow data [102], an LLM's transformation of input into distributed latent features defies such tracing [103]. Consequently, if a piece of malicious or sensitive information passes from one domain's agent to another, it becomes nearly impossible to pinpoint which input triggered a given action or decision downstream [83].

This provenance gap enables novel attack scenarios not seen in single-domain settings. For example, an adversary might compromise an agent in Domain A and subtly inject false but plausible data or instructions into its output. When Domain B's agent consumes this tainted output (trusting Domain A), it may make a high-impact decision or perform an action based on that input. Due to separate logs and limited cross-domain visibility, Domain B cannot easily attribute the aberrant behavior to Domain A's influence. The attack can propagate across multiple AI systems, causing cascading failures as each agent unwittingly passes along or acts on corrupted information [104, 105]. Critically, the perpetrator avoids attribution because no single domain's audit trail captures the full chain of custody. This cross-domain exploit is fundamentally different from same-domain issues: even if each agent individually is audited, the inter-domain links remain opaque. Due to this ambiguity of internal contextual state of LLMs, it's difficult for an organization to determine how much of the contextual data of various LLM agents should be shared with other organizations to trace the inter-domain data provenance.

Existing interpretability and auditing techniques struggle with this cross-domain provenance problem [106]. Methods like influence functions (which estimate which training or input data most affected a model's output), internal activation monitoring [107], or prompting the model to generate citations for its outputs provide only partial insight [108]. These approaches are confined to single-model reasoning and often cannot cleanly disentangle which external agent's input drove a decision. In a multi-LLM chain, influence attribution is diluted and post-hoc explanations can be manipulated or incomplete.

(C6) Cross-domain Context Bypass: Cross-domain multi-agent LLM deployments are reshaping enterprise workflows: confidential-data agents inside one company can converse with partner-controlled agents to coordinate projects. This federation introduces a security gap unseen in local multi-agent or single-model settings. Once knowledge crosses an organizational border, no single party retains full visibility of the dialogue, yet corporate policies—"never reveal individual exact salaries," "export only statistical summaries"—still apply. Traditional controls assume a guardrail sees the entire request/response prompts of a single agent and can judge it in isolation [20, 21]. In contrast, in federated LLM chains, context is fragmented not only across prompts, but also across agents, meaning policy must be applied to the comprehensive context of multiple agents.

As a simple attack scenario, suppose an external contractor asks another organization's payroll LLM, *"Return the maximum salary in the AI research department"* and asks the organization's HR management LLM, *"Return the name of the personnel who gets paid most in the AI research department"*. Combining the two pieces of information returned from different agents, the contractor can derive the exact of salary of a specific personnel. Similarly, an external entity may issue job_1 to LLM_A and job_2 to LLM_B, where the organization regards job_{1+2} as a forbidden request. Such multi-agent contextual-based attack can be developed even to multi-domain contextual-based attacks, whose countermeasure remains largely unexplored. Unless provenance and context-sharing mechanisms span the boundaries of agents and organizations, enterprises risk silent policy violations that neither side can reconstruct after the fact, leaving confidential data exposed without an obvious culprit.

Traditional role-based access control can hide raw salary tables from external agents, but it cannot stop those agents from reconstructing the numbers through incremental queries. Static keyword filters or turn-by-turn firewalls fare no better: they accept each benign fragment while missing the composite leak. Zero-knowledge [109] or differential-privacy [110] safeguards, designed for atomic disclosures, likewise struggle because natural-language answers continuously evolve [111]. Researchers already observe that multi-turn prompt injections and leaks bypass defenses that succeed in one-shot dialogs, yet current enterprise toolkits still treat every message as independent [112].

(C7) Inter-domain Confidentiality and Integrity: Recent research works propose privacy-preserving multi-agent LLM services [113], where a user's input and output data remain invisible to the data-processing agents [114, 115]. For example, remote cloud platforms can offer diagnosis-and-prescription pipelines powered by multiple LLM agents: a patient (Alice) uploads CT and X-ray images in homomorphically encrypted forms, and then the cooperating agents—each run by different cloud vendors (e.g., imaging model, differential-diagnosis model, prescription generator)—operate blindly, never seeing Alice's plaintext data. They return an encrypted prescription that only Alice can decrypt. Alice then forwards the decrypted prescription to an independent online pharmacy.

This architecture solves data-exposure risk for Alice and shields the cloud vendors from HIPAA liability [116], yet it creates a novel integrity gap. Because every intermediate result stays encrypted inside separate domains, no single party can later attest to the exact plaintext the pipeline produced. Exploiting this, a malicious or negligent user can mount a forged-output attack: after decryption, Alice alters the dosage or drug name, claims the modified text came from the blind service, and submits it to the pharmacy. The pharmacy has no cryptographic proof that the prescription truly emerged from the designated multi-agent workflow; the vendors cannot reveal or sign what they never saw, and disclosing the ciphertext is useless to the pharmacist. This threat does not arise in single-domain or local multi-agent systems, where the same operator can log and sign outputs. Until scalable, privacy-preserving attestation mechanisms are devised, pharmacies and other third parties remain vulnerable to forged outputs, and cloud LLM vendors cannot defend the authenticity of their encrypted work products.

Current cryptographic technologies cannot fully close the gap. State-of-the-art privacy-preserving techniques such as fully homomorphic encryption (FHE) [117], multi-party computation (MPC) [118], or zero-knowledge proofs (ZKP) [119] partially address the privacy and integrity concerns, but their practicality is low because their computation and communication overheads are often orders of magnitude higher. In a multi-domain setup, the problem becomes even more complex than traditional two-domain privacy-preserving ML inference, because agents would need an additional framework that allows them to securely set up multiple cryptographic keys or convert keys during their cross-domain activities. Also, as the agent network becomes larger and comprises multiple processing pipelines, their computation and communication overheads will scale drastically.

4 Recommended Security Guideline

This section provides a few insights on evaluation metrics for the security issues of cross-domain multi-agent LLMs discussed in §3, and suggests high-level guidelines for possible countermeasures.

4.1 Security Evaluation Metrics

We propose a guideline for evaluation metrics for each of 7 security challenges, to measures how well a given cross-domain multi-agent LLM deployment addresses each of the security issues.

Figure 2 describes our proposed evaluation metrics for each of the 7 security issues of cross-domain multi-agent LLM systems.

(C1's Metrics) We check how much dynamic a team is and whether new members inherit correct privileges. A low *"Group Volatility"*, high *"On-boarding Trust"*, and perfect *"Policy Consistency"* together indicate that dynamic grouping does not degrade a system's trust boundary.

(C2's Metrics) We quantify covert coordination risk. A rising *"Collusion Risk"* or *"Covert-channel Score"* [11] paired with a falling *"Independence Ratio"* warns that a subset of agents may be manipulating outcomes or exchanging hidden payloads; operators can throttle or sandbox the culprits in real-time.

(C3's Metrics) We capture structural alignment. *"Goal Completeness"* measures incentive design, *"Conflict Resolution"* tracks the system's built-in negotiation layer, and *"Mutual Benefit"* ensures that no agent chronically loses the reward game. They together reveal whether reward shaping and arbitration logic truly harmonize individual and collective objectives.

(C4's Metrics) We cover the problem of live model evolution. *"Tuning Log Coverage"* provides post-event traceability, *"Drift-detection latency"* offers an early-warning signal, and *"Performance*

ID	Challenge	Evaluation Metrics	Description
C1	Unvetted Dynamic Grouping	› Group Volatility › On-boarding Trust › Policy Consistency	› Ratio of agents joining or leaving the group per period › Ratio of agents passing identity, sandbox, behavior vettings › Ratio of member ACL/key updates within policy deadline
C2	Collusion Control	› Collusion Risk › Covert-Channel › Independence	› Anomaly score (payoff surplus, hidden-channel entropy) › Format checks, steganography detection score › Ratio of decisions made without peer participation
C3	Conflicting Incentives & Goals	› Goal Completeness › Conflict Resolution › Mutual Benefit	› Ratio between individual rewards and global utility › Resolution ratio of detected goal conflicts › Ratio of agents fulfilling baseline utility
C4	Distributed Self-tuning Misalignment	› Tuning Log Coverage › Drift Detection Latency › Performance Consistency	› Ratio of logged self-tuning activities › Delay between self-tuning and anomaly detection › Pass rate of test suite after each self-tuning
C5	Cross-domain Provenance Obscurity	› Provenance Coverage › Source Verification › Action Traceability	› Ratio of events having full source-to-sink trace › Ratio of authenticated cross-domain messages › Ratio of critical actions with causal chain reconstruction
C6	Cross-domain Context Bypass	› Ill-prompt Block Rate › False-positive › Infection Propagation	› Ratio of attack prompts blocked or sanitized › Benign prompts wrongly blocked › Ratio of agents affected per missed attack
C7	Inter-domain Confidentiality & Integrity	› Secure Channel Utility › Data Leakage Incidence › Request Vetting	› Ratio of encrypted + authenticated inter-domain messages › Ratio of sensitive-data-leaking messages › Ratio of blocked illicit requests

Figure 2: Evaluation metrics for 7 security issues of cross-domain multi-agent LLM deployments.

consistency" prevents silent capability decay, which is crucial for any service that permits autonomous fine-tuning.

(C5's Metrics) We turn accountability into a measurable property. *"Provenance Coverage"* and *"Source Verification"* guarantee that information lineage is complete and authenticated, while *"Action Traceability"* enables forensic reconstruction of high-impact decisions.

(C6's Metrics) We instrument text-level defenses. A high *"Ill-prompt Block Rate"*, minimal *"False-positives"*, and near-zero *"Infection Propagation"* [83] collectively show that policy filters neither leak sensitive data nor cripple collaboration.

(C7's Metrics) We extend classical CIA principles to multi-LLM pipelines. *"Secure-channel Utility"* covers transport-layer secrecy, counting *"Data Leakage"* quantifies confidentiality failures, and *"Request Vetting"* confirms that access-control logic rejects unauthorized actions.

Because every metric above is a ratio, providers can stream them to a dashboard and set policy thresholds (e.g., "halt execution if some metric drops below 0.9"). Researchers can report the same values to make security claims reproducible, and regulators gain a ready-made scorecard for certification audits—all from a single, compact table.

4.2 Countermeasures

We provide possible directions to address each of the seven security issues. Some proposals (C3, C4) use a trusted arbitration system between domains, which is different from an hierarchical authority.

(C1's Countermeasure) Trust-adaptive Dynamic Teaming: Extending the idea of multi-agent partner selection [19], each agent can maintain a differentiable trust ledger – a graph-neural module that updates peer-trust scores from message consistency, error rates, and rule violations observed in real time. Dialogue routes are weighted by these scores: low-trust peers are quarantined or demoted; high-trust peers gain routing priority or temporary leadership. Reinforcement learning drives the ledger toward team-level reward (task success + security bonus for blocking anomalies). Agents exchange signed "trust snapshots," letting newcomers bootstrap without disclosing private logs.

(C2's Countermeasure) Adversarial Multi-agent Training for Collusion Resistance: A collusion generator can produce synthetic teaming scenarios where subsets of agents coordinate hidden agendas (via steganography, role laundering, etc.). Honest agents and a learnable incentive module co-train against these adversaries: the module maps conversation features into adaptive bonus/penalty signals, rewarding evidence-backed dissent and penalizing correlated misreports. Training proceeds in self-play cycles until Nash-like equilibrium where collusion yields no net gain. For example, [120] uses an adversarial training technique to prevent e-commerce pricing collusion among sellers.

(C3's Countermeasure) Hierarchical Conflict Arbitration via Meta-LLM Controller: One can propose a meta-level LLM controller that monitors and arbitrates decisions among hierarchical agents. This meta-agent continually receives the high-level goals and the lower-level agents' proposed actions or outputs. It uses learned conflict-detection mechanisms (e.g. natural language inference or consistency checks) to identify contradictions between a supervisor's intent and a sub-agent's behavior. Upon detecting a conflict, the meta-LLM generates a resolving instruction or adjustment, effectively mediating between layers. The instruction can be applied after being approved by human operators from both domains. Over time, the subordinate agents could be fine-tuned with feedback from the meta-controller, learning to preemptively avoid hierarchical inconsistencies.

(C4's Countermeasure) Cross-domain Reward Alignment via Adaptive Credit Assignment: We can introduce an adaptive credit assignment network that aligns local rewards of each agent with the global multi-agent objective. During training, a shared critic model (potentially an LLM-based evaluator) observes the collective outcome and each agent's contributions (e.g. action traces or textual outputs). It then computes tailored reward signals for each agent by estimating their marginal impact on the overall result. The system would dynamically adjust each agent's reward function via gradient signals so that maximizing individual rewards also maximizes the team's performance. The adjusted reward plans can be vetted via a simulation or training before being deployed. While existing works [121] focus on adaptively transferring a reward function from one task to another, our novel challenge is to train a differentiable communication channel among agents to propagate global reward information back into each agent's policy update.

(C5's Countermeasure) Neural Provenance Tracking with Embedded Signatures: Each agent can embed subtle signatures (e.g., watermark [122]) into its generated content to mark its contributions. Concretely, agents are trained to include quasi-imperceptible metadata in their textual outputs – for example, by biasing certain token choices or using a private vocabulary of markers – that do not alter the apparent meaning but carry identifying information. A dedicated decoder model or forensic LLM can later extract these hidden signatures from the final multi-agent output, reconstructing a timeline or graph of which agent produced or modified each part. One challenge is to harden this neural signature scheme to survive transformations (e.g. rephrasing by subsequent agents) and to remain robust against adversarial removal.

(C6's Countermeasure) Session-level Semantic Firewalls: Instead of turn-by-turn filtering, a dedicated firewall LLM ingests the entire multi-agent dialogue in a sliding window, building a contextual knowledge graph of entities, quantitative tokens, and policy tags (e.g., "salary-individual"). It uses contrastive training to spot when a new utterance, combined with prior context, breaches a policy. Detected violations trigger automatic redaction or request-for-clarification messages. The firewall's policy model continually fine-tunes on anonymized leak/non-leak transcripts shared across domains under differential privacy.

(C7's Countermeasure) Verifiable Reasoning with Privacy: In blind inference, an agent emits (i) an encrypted answer and (ii) a public proof sketch—hash-bound logic traces or statistical bounds—letting a verifier LLM confirm semantic correctness without seeing the private input. Lightweight ZK proofs [122] and encrypted-embedding checks (run in secure enclaves) enable this, yet current protocols remain too slow for real-time use. A potential fix is a hybrid design that blends these proofs with fast two-party linear MPC [123], slashing latency while preserving privacy.

5 Conclusions and Outlook

Cross-domain multi-agent LLMs hold transformative promise, but only if security becomes a first-class design constraint rather than an afterthought. By mapping seven unaddressed challenge

areas, we have shown that neither single-agent defenses nor traditional multi-agent safeguards suffice once models cross ownership boundaries. Tackling these problems demands tight collaboration between the AI-safety, cryptography, and distributed-systems communities, coupled with rigorous open benchmarks to quantify the security–utility trade-offs unique to cross-domain deployments. The research directions we outline argue for security primitives that are as adaptive and learning-centric as the agents themselves. Addressing them now will prevent tomorrow's agent societies from repeating the Internet's costly security debt.

References

[1] Caroline Wang, Arrasy Rahman, Ishan Durugkar, Elad Liebman, and Peter Stone. N-agent ad hoc teamwork. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[2] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 51991–52008. Curran Associates, Inc., 2023.

[3] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of LLMs, 2025.

[4] Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. Ad hoc autonomous agent teams: collaboration without pre-coordination. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1504–1509. AAAI Press, 2010.

[5] Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arı k. Chain of agents: Large language models collaborating on long-context tasks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 132208–132237. Curran Associates, Inc., 2024.

[6] Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 180–192, Singapore, December 2023. Association for Computational Linguistics.

[7] Ananta Mukherjee, Peeyush Kumar, Boling Yang, Nishanth Chandran, and Divya Gupta. Privacy preserving multi-agent reinforcement learning in supply chains, 2023.

[8] Xiaohe Bo, Zeyu Zhang, Quanyu Dai, Xueyang Feng, Lei Wang, Rui Li, Xu Chen, and Ji-Rong Wen. Reflective multi-agent collaboration based on large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 138595–138631. Curran Associates, Inc., 2024.

[9] Vighnesh Subramaniam, Yilun Du, Joshua B. Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains, 2025.

[10] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022.

[11] Sumeet Ramesh Motwani, Mikhail Baranchuk, Martin Strohmeier, Vijay Bolina, Philip H.S. Torr, Lewis Hammond, and Christian Schroeder de Witt. Secret collusion among ai agents: Multi-agent deception via steganography. In A. Globerson, L. Mackey, D. Belgrave, A. Fan,

U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 73439–73486. Curran Associates, Inc., 2024.

[12] Jinwei Hu, Yi Dong, Shuang Ao, Zhuoyun Li, Boxuan Wang, Lokesh Singh, Guangliang Cheng, Sarvapali D. Ramchurn, and Xiaowei Huang. Position: Towards a responsible llm-empowered multi-agent systems, 2025.

[13] Brandon Cui, Hengyuan Hu, Luis Pineda, and Jakob Nicolaus Foerster. K-level reasoning for zero-shot coordination in hanabi. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[14] Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Cooperation, competition, and maliciousness: Llm-stakeholders interactive negotiation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 83548–83599. Curran Associates, Inc., 2024.

[15] Francis Rhys Ward, Francesco Belardinelli, Francesca Toni, and Tom Everitt. Honesty is the best policy: defining and mitigating ai deception. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[16] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

[17] Muhammad Rahman, Jiaxun Cui, and Peter Stone. Minimum coverage sets for training robust ad hoc teamwork agents. AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[18] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other–Play": for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[19] Nicolas Anastassacos, Stephen Hailes, and Mirco Musolesi. Partner selection for the emergence of cooperation in multi-agent systems using reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7047–7054, Apr. 2020.

[20] Rana Muhammad Shahroz Khan, Zhen Tan, Sukwon Yun, Charles Flemming, and Tianlong Chen. *Agents Under Siege*: Breaking pragmatic multi-agent llm systems with optimized prompt attacks, 2025.

[21] Donghyun Lee and Mo Tiwari. Prompt infection: LLM-to-LLM prompt injection within multi-agent systems, 2025.

[22] Ted Fujimoto, Samrat Chatterjee, and Auroop R. Ganguly. Ad hoc teamwork in the presence of adversaries. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

[23] Yuxuan Zhu, Antony Kellermann, Akul Gupta, Philip Li, Richard Fang, Rohan Bindu, and Daniel Kang. Teams of llm agents can exploit zero-day vulnerabilities, 2025.

[24] Alexander Bukharin, Yan Li, Yue Yu, Qingru Zhang, Zhehui Chen, Simiao Zuo, Chao Zhang, Songan Zhang, and Tuo Zhao. Robust multi-agent reinforcement learning via adversarial regularization: theoretical foundation and stable algorithms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[25] Yang Li, Wenhao Zhang, Jianhong Wang, Shao Zhang, Yali Du, Ying Wen, and Wei Pan. Aligning individual and collective objectives in multi-agent cooperation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 44735–44760. Curran Associates, Inc., 2024.

[26] Bowen Baker. Emergent reciprocity and team formation from randomized uncertain social preferences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[27] Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, Yunjun Gao, Yingcai Wu, and Shouling Ji. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents, 2024.

[28] Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčiak, The Anh Han, Edward Hughes, Vojtěch Kovařík, Jan Kulveit, Joel Z. Leibo, Caspar Oesterheld, Christian Schroeder de Witt, Nisarg Shah, Michael Wellman, Paolo Bova, Theodor Cimpeanu, Carson Ezell, Quentin Feuillade-Montixi, Matija Franklin, Esben Kran, Igor Krawczuk, Max Lamparth, Niklas Lauffer, Alexander Meinke, Sumeet Motwani, Anka Reuel, Vincent Conitzer, Michael Dennis, Iason Gabriel, Adam Gleave, Gillian Hadfield, Nika Haghtalab, Atoosa Kasirzadeh, Sébastien Krier, Kate Larson, Joel Lehman, David C. Parkes, Georgios Piliouras, and Iyad Rahwan. Multi-agent risks from advanced ai, 2025.

[29] Guanlin Liu and Lifeng LAI. Efficient adversarial attacks on online multi-agent reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 24401–24433. Curran Associates, Inc., 2023.

[30] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

[31] Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications, 2024.

[32] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. BERT-ATTACK: Adversarial attack against BERT using BERT. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online, November 2020. Association for Computational Linguistics.

[33] Yuanwei Wu, Xiang Li, Yixin Liu, Pan Zhou, and Lichao Sun. Jailbreaking gpt-4v via self-adversarial attacks with system prompts, 2024.

[34] Govind Ramesh, Yao Dou, and Wei Xu. GPT-4 jailbreaks itself with near-perfect success using self-explanation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22139–22148, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[35] Cem Anil, Esin Durmus, Nina Panickssery, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel Ford, Fracesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan J Hubinger, Yuntao Bai, Trenton Bricken, Timothy Maxwell, Nicholas Schiefer, James Sully, Alex Tamkin, Tamera Lanhan, Karina Nguyen, Tomasz Korbak, Jared Kaplan, Deep Ganguli, Samuel R. Bowman, Ethan Perez, Roger Baker Grosse, and David Duvenaud. Many-shot jailbreaking. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 129696–129742. Curran Associates, Inc., 2024.

[36] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing NLP. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China, November 2019. Association for Computational Linguistics.

[37] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 61065–61105. Curran Associates, Inc., 2024.

[38] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[39] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[40] Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models, 2023.

[41] Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox, 2024.

[42] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21527–21536, Mar. 2024.

[43] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *International Conference on Learning Representations (ICLR)*.

[44] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[45] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. Llm agents can autonomously hack websites, 2024.

[46] Joseph Cox. Gpt-4 hired unwitting taskrabbit worker by pretending to be 'vision-impaired' human, 2023.

[47] Feng He, Tianqing Zhu, Dayong Ye, Bo Liu, Wanlei Zhou, and Philip S. Yu. The emerged security and privacy of llm agent: A survey with case studies, 2024.

[48] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35413–35425. PMLR, 23–29 Jul 2023.

[49] Ka-Ho Chow, Wenqi Wei, and Lei Yu. Imperio: language-guided backdoor attacks for arbitrary model control. IJCAI '24, 2024.

[50] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Membership inference attacks against fine-tuned large language models via self-prompt calibration. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 134981–135010. Curran Associates, Inc., 2024.

[51] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[52] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online, November 2020. Association for Computational Linguistics.

[53] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025, Apr. 2020.

[54] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input–output safeguard for human–ai conversations, 2023.

[55] Akshita Jha and Chandan K. Reddy. Codeattack: code-based adversarial attacks for pre-trained programming language models. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023.

[56] Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. Toxicity in chatgpt: Analyzing persona-assigned language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1236–1270, Singapore, December 2023. Association for Computational Linguistics.

[57] Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. Fight back against jailbreaking via prompt adversarial tuning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 64242–64272. Curran Associates, Inc., 2024.

[58] Andy Zhou, Bo Li, and Haohan Wang. Robust prompt optimization for defending language models against jailbreaking attacks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 40184–40211. Curran Associates, Inc., 2024.

[59] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*, 2020.

[60] Zhehua Zhong, Tianyi Chen, and Zhen Wang. Mat: mixed-strategy game of adversarial training in fine-tuning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023.

[61] Hongqiu Wu, Ruixue Ding, Hai Zhao, Pengjun Xie, Fei Huang, and Min Zhang. Adversarial self-attention for language understanding. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023.

[62] Jaehyung Kim, Yuning Mao, Rui Hou, Hanchao Yu, Davis Liang, Pascale Fung, Qifan Wang, Fuli Feng, Lifu Huang, and Madian Khabsa. RoAST: Robustifying language models via adversarial perturbation with selective training. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3412–3444, Singapore, December 2023. Association for Computational Linguistics.

[63] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022.

[64] Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong, and Hang Su. Evil geniuses: Delving into the safety of llm-based agents, 2024.

[65] Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pang, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, Bo An, and Qingsong Wen. A survey on trustworthy llm agents: Threats and countermeasures, 2025.

[66] Junyuan Mao, Fanci Meng, Yifan Duan, Miao Yu, Xiaojun Jia, Junfeng Fang, Yuxuan Liang, Kun Wang, and Qingsong Wen. Agentsafe: Safeguarding large language model-based multi-agent systems via hierarchical data management, 2025.

[67] Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. Watch out for your agents! investigating backdoor threats to LLM-based agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[68] Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6065–6086, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[69] Sander Schulhoff, Jeremy Pinto, Anaum Khan, Louis-François Bouchard, Chenglei Si, Svetlina Anati, Valen Tagliabue, Anson Kost, Christopher Carnahan, and Jordan Boyd-Graber. Ignore this title and HackAPrompt: Exposing systemic vulnerabilities of LLMs through a global prompt hacking competition. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4945–4977, Singapore, December 2023. Association for Computational Linguistics.

[70] Da Cheng Gu and Wei Liu. Assessing vulnerabilities in state-of-the-art large language models through hex injection (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28):29377–29378, Apr. 2025.

[71] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Kailong Wang. A hitchhiker's guide to jailbreaking chatgpt via prompt engineering. SEA4DQ 2024, page 12–21, New York, NY, USA, 2024. Association for Computing Machinery.

[72] Weichen Yu, Kai Hu, Tianyu Pang, Chao Du, Min Lin, and Matt Fredrikson. Infecting LLM agents via generalizable adversarial attack. In *Red Teaming GenAI: What Can We Learn from Adversaries?*, 2025.

[73] Richard Fang, Rohan Bindu, Akul Gupta, Qiusi Zhan, and Daniel Kang. Teams of llm agents can exploit zero-day vulnerabilities. *CoRR*, abs/2406.01637, 2024.

[74] Gabriel Mukobi, Ann-Katrin Reuel, Juan-Pablo Rivera, and Chandler Smith. Assessing risks of using autonomous language models in military and diplomatic planning. In *Multi-Agent Security Workshop @ NeurIPS'23*, 2023.

[75] Eugene Bagdasaryan, Tsung-Yin Hsieh, Ben Nassi, and Vitaly Shmatikov. (ab)using images and sounds for indirect instruction injection in multi-modal llms. *CoRR*, abs/2307.10490, 2023.

[76] Elias Abad Rocamora, Yongtao Wu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. Revisiting character-level adversarial attacks for language models. In *Forty-first International Conference on Machine Learning*, 2024.

[77] Alfonso Amayuelas, Xianjun Yang, Antonis Antoniades, Wenyue Hua, Liangming Pan, and William Yang Wang. MultiAgent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6929–6948, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[78] Xiaohu Du, Fan Mo, Ming Wen, Tu Gu, Huadi Zheng, Hai Jin, and Jie Shi. Multi-turn jailbreaking large language models via attention shifting. In *AAAI*, pages 23814–23822, 2025.

[79] Ana Nunez, Nafis Tanveer Islam, Sumit Kumar Jha, and Peyman Najafirad. Autosafecoder: A multi-agent framework for securing llm code generation through static analysis and fuzz testing, 2024.

[80] Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents, 2024.

[81] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, AISec '23, page 79–90, New York, NY, USA, 2023. Association for Computing Machinery.

[82] Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing RLHF protections in GPT-4 via fine-tuning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 681–687, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[83] Pierre Peigne-Lefebvre, Mikolaj Kniejski, Filip Sondej, Matthieu David, Jason Hoelscher-Obermaier, Christian Schroeder de Witt, and Esben Kran. Multi-agent security tax: Trading off security and collaboration capabilities in multi-agent systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27573–27581, 2025.

[84] Haoran Li, Yulin Chen, Zihao Zheng, Qi Hu, Chunkit Chan, Heshan Liu, and Yangqiu Song. Simulate and eliminate: Revoke backdoors for generative large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):397–405, Apr. 2025.

[85] Anil Ramakrishna, Jimit Majmudar, Rahul Gupta, and Devamanyu Hazarika. LLM-PIRATE: A benchmark for indirect prompt injection attacks in large language models. In *The Third Workshop on New Frontiers in Adversarial Machine Learning*, 2024.

[86] Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Hassan Sajjad, and Frank Rudzicz. Immunization against harmful fine-tuning attacks. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5234–5247, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[87] Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration, 2024.

[88] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *Proceedings of the 43rd IEEE Symposium on Security and Privacy (S&P)*, 2022. Also available as arXiv:2108.00352.

[89] Rui Zeng, Xi Chen, Yuwen Pu, Xuhong Zhang, Tianyu Du, and Shouling Ji. Clibe: Detecting dynamic backdoors in transformer-based nlp models. In *Proceedings of the 32nd Network and Distributed System Security Symposium (NDSS)*, 2025.

[90] David Cohen. Jfrog and hugging face join forces to expose malicious ml models, 2025.

[91] Lixia Zan, Xiangbin Zhu, and Zhaolong Hu. Adversarial attacks on cooperative multi-agent deep reinforcement learning: A dynamic group-based adversarial example transferability method. *Complex & Intelligent Systems*, 9:7439–7450, 2023.

[92] Yifan Zang, Jinmin He, Kai Li, Haobo Fu, QIANG FU, Junliang Xing, and Jian Cheng. Automatic grouping for efficient cooperative multi-agent reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[93] Lun Wang, Zaynah Javed, Xian Wu, Wenbo Guo, Xinyu Xing, and Dawn Song. Backdoorl: Backdoor attack against competitive reinforcement learning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3699–3705. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[94] Wei Duan, Jie Lu, and Junyu Xuan. Group-aware coordination graph for multi-agent reinforcement learning. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 3926–3934. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track.

[95] Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent llm systems fail?, 2025.

[96] Zengqing Wu, Run Peng, Shuyuan Zheng, Qianying Liu, Xu Han, Brian I. Kwon, Makoto Onizuka, Shaojie Tang, and Chuan Xiao. Shall we team up: Exploring spontaneous cooperation of competing LLM agents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5163–5186, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[97] Yanchao Sun, Ruijie Zheng, Parisa Hassanzadeh, Yongyuan Liang, Soheil Feizi, Sumitra Ganesh, and Furong Huang. Certifiably robust policy learning against adversarial multi-agent communication. In *The Eleventh International Conference on Learning Representations*, 2023.

[98] Chuxiong Sun, Zehua Zang, Jiabao Li, Jiangmeng Li, Xiao Xu, Rui Wang, and Changwen Zheng. T2mac: targeted and trusted multi-agent communication through selective engagement and evidence-driven integration. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024.

[99] Bei Chen, Gaolei Li, Xi Lin, Zheng Wang, and Jianhua Li. BlockAgents: Towards byzantine–robust llm–based multi–agent coordination via blockchain. In *Proceedings of the ACM TURC 2024 (ACM Turing Celebration Conference/China)*, 2024.

[100] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023.

[101] Xuezhou Zhang, Yuzhe Ma, Adish Singla, and Xiaojin Zhu. Adaptive reward-poisoning attacks against reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[102] William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, 32(2):1–29, 2014.

[103] Shoaib Ahmed Siddiqui, Radhika Gaonkar, Boris Köpf, David Krueger, Andrew Paverd, Ahmed Salem, Shruti Tople, Lukas Wutschitz, Menglin Xia, and Santiago Zanella-Béguelin. Permissive information-flow analysis for large language models, 2024.

[104] Satbir Singh. Llm-based agents: The benefits and the risks, 2025.

[105] Christian Schroeder de Witt. Open challenges in multi-agent security: Towards secure systems of interacting ai agents, 2025.

[106] Dami Choi, Yonadav G Shavit, and David Duvenaud. Tools for verifying neural models' training data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[107] Dan Petrovic. Advanced interpretability techniques for tracing llm activations, 2025.

[108] Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 21079–21113. Curran Associates, Inc., 2024.

[109] Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. Huref: Human-readable fingerprint for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 126332–126362. Curran Associates, Inc., 2024.

[110] Timour Igamberdiev, Thomas Arnold, and Ivan Habernal. DP-rewrite: Towards reproducibility and transparency in differentially private text rewriting. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2927–2933, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[111] Weiyan Shi, Ryan Shea, Si Chen, Chiyuan Zhang, Ruoxi Jia, and Zhou Yu. Just fine-tune twice: Selective differential privacy for large language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6327–6340, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[112] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[113] Meng Hao, Hongwei Li, Hanxiao Chen, Pengzhi Xing, Guowen Xu, and Tianwei Zhang. Iron: Private inference on transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15718–15731. Curran Associates, Inc., 2022.

[114] Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert, and Rickmer F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, 2020.

[115] Francis Dutil, Alexandre See, Lisa Di-Jorio, and Florent Chandelier. Application of homomorphic encryption in medical imaging. *ArXiv*, abs/2110.07768, 2021.

[116] Harshal Tupsamudre, Arun Kumar, Vikas Agarwal, Nisha Gupta, and Sneha Mondal. Ai-assisted controls change management for cybersecurity in the cloud. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12629–12635, Jun. 2022.

[117] Leo de Castro, Antigoni Polychroniadou, and Daniel Escudero. Privacy-preserving large language model inference via GPU-accelerated fully homomorphic encryption. In *Neurips Safe Generative AI Workshop 2024*, 2024.

[118] Deevashwer Rathee, Dacheng Li, Ion Stoica, Hao Zhang, and Raluca Popa. MPC-minimized secure LLM inference, 2025.

[119] Tao Lu, Haoyu Wang, Wenjie Qu, Zonghui Wang, Jinye He, Tianyang Tao, Wenzhi Chen, and Jiaheng Zhang. An efficient and extensible zero-knowledge proof framework for neural networks. Cryptology ePrint Archive, Paper 2024/703, 2024.

[120] Gianluca Brero, Eric Mibuari, Nicolas Lepore, and David C. Parkes. Learning to mitigate AI collusion on economic platforms. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[121] Johan Ferret, Raphaël Marinier, Matthieu Geist, and Olivier Pietquin. Self-attentional credit assignment for transfer in reinforcement learning, 2020.

[122] Tong Zhou, Xuandong Zhao, Xiaolin Xu, and Shaolei Ren. Bileve: Securing text provenance in large language models against spoofing with bi-level signature. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 56054–56075. Curran Associates, Inc., 2024.

[123] Satrajit Ghosh, Jesper Buus Nielsen, and Tobias Nilges. Maliciously secure oblivious linear function evaluation with constant overhead. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 629–659, Cham, 2017. Springer International Publishing.