FUNDAMENTAL SCIENTIFIC REPORTS IN MULTIDISCIPLINARY AREAS

Vol 2 Issue 03 (2026)

www.fsrma.org

ONLINE ISSN: 3134-7517

PRINT ISSN: 3134-7509

SafeComm-Agent: A Security-Aware Communication and Inference Optimization Framework for Multi-Agent Embodied Intelligence Systems

Linting Wu

Technical expert, Bytedance, Shanghai, China

Amberber1230@gmail.com

Shanshan Lu

Suzhou University, Jiangsu, China

zqh13243546@163.com

Abstract.

Multi-agent embodied intelligence systems increasingly integrate vision, language, and sensor-based perception to support collaborative navigation, manipulation, and autonomous decision-making in dynamic environments, but frequent inter-agent communication, redundant large-model inference, and multimodal adversarial attacks can simultaneously increase computational overhead and compromise physical task safety. To address these challenges, this study proposes SafeComm-Agent, a security-aware communication and inference optimization framework that jointly coordinates multimodal attack detection, agent communication, and adaptive reasoning computation. The framework first combines Vision Transformers (ViTs), Vision-Language Models (VLMs), cross-modal consistency analysis, and uncertainty estimation to identify anomalous observations and derive agent-level security risks. It then employs a Graph Neural Network (GNN) to construct a dynamic security-aware communication graph, while Multi-Agent Reinforcement Learning (MARL) determines when to communicate, which agents to contact, and what information to transmit; risk-aware message compression and adaptive Test-Time Compute (TTC) further regulate communication and inference budgets, with Monte Carlo Tree Search (MCTS) providing additional verification for high-risk states. Experiments in Habitat-based simulated embodied environments with controlled multimodal adversarial perturbations show that SafeComm-Agent achieves a task success rate of 94.6% and an attack detection F1-score of 93.2%. Compared with full-communication MARL, it reduces inference cost by 26%, while achieving a 39% reduction in total computational cost compared with full-communication and full-computation execution. These results demonstrate effective joint optimization of security, communication, and adaptive inference.

Keywords: Embodied Intelligence; Multi-Agent Systems; Multimodal Adversarial Attack; Secure Communication; Graph Neural Network; Multi-Agent Reinforcement Learning; Test-Time Compute; Monte Carlo Tree Search

1. Introduction

Multi-agent embodied intelligence systems are increasingly being developed to enable multiple autonomous agents to perceive physical environments, communicate with one another, reason over multimodal information, and collaboratively execute complex tasks such as household assistance, autonomous navigation, robotic manipulation, and human-robot interaction. By integrating vision, language, depth, LiDAR, and proprioceptive observations with large multimodal models, these systems can coordinate complementary capabilities and improve task-level autonomy. However, the deployment of embodied multi-agent systems in dynamic and safety-critical environments introduces a coupled challenge involving security, communication efficiency, and inference computation. Multimodal adversarial perturbations, sensor spoofing, conflicting instructions, and malicious information injected through inter-agent communication may cause agents to form incorrect environmental beliefs or execute unsafe actions. Meanwhile, frequent message exchange and repeated large-model reasoning substantially increase communication traffic, token consumption, GPU utilization, and decision latency. Therefore, independently optimizing security detection, communication scheduling, or inference acceleration is insufficient for achieving reliable and resource-efficient embodied collaboration. A key difficulty is that security risk and computational demand are dynamically coupled with the task state. For example, a navigation agent operating under normal observations may require only limited communication and single-pass inference, whereas an agent receiving inconsistent visual-language information or suspicious sensor observations may require additional cross-agent verification and deeper reasoning. Conventional multi-agent communication strategies generally rely on fixed or dense communication patterns and consequently transmit redundant information even when inter-agent dependencies are weak. Similarly, static inference budgets cannot distinguish between low-risk routine states and high-risk states requiring additional reasoning. This motivates a security-aware joint optimization paradigm in which when to communicate, whom to communicate with, what information to transmit, and how much inference computation to allocate are determined according to task dependency, information value, and security risk. To address this challenge, this study proposes SafeComm-Agent, a Security-Aware Communication and Inference Optimization Framework for Multi-Agent Embodied Intelligence Systems. SafeComm-Agent first integrates multimodal perception, cross-modal consistency analysis, anomaly detection, and uncertainty estimation to construct agent-level security risk representations. A dynamic security-aware Graph Neural Network (GNN) then models task dependencies and communication relationships, while Multi-Agent Reinforcement Learning (MARL) jointly determines communication scheduling and information transmission. Risk-aware message compression reduces redundant communication, and adaptive Test-Time Compute (TTC) allocates additional reasoning resources according to uncertainty and security risk. For high-risk states, Monte Carlo Tree Search (MCTS) is further introduced to perform deeper verification before final collaborative actions.

The main contributions of this study are as follows:

Security-aware embodied collaboration: SafeComm-Agent integrates multimodal attack detection and risk estimation into the communication and decision-making process, enabling security information to directly influence agent coordination.

Joint communication and inference optimization: The framework jointly optimizes communication topology, message selection, and adaptive inference computation rather than treating communication and reasoning costs independently.

Risk-adaptive reasoning mechanism: A combination of MARL-based online scheduling and MCTS-based high-risk verification dynamically allocates computational resources according to task difficulty and security risk. Unified evaluation framework: The proposed approach evaluates

task performance, attack detection, communication efficiency, inference cost, and action safety within a common multi-agent embodied intelligence setting.

1. Literature Review

Recent advances in embodied artificial intelligence (Embodied AI), multi-agent reinforcement learning, multimodal foundation models, and inference-time reasoning have created new opportunities for developing collaborative autonomous systems. Existing studies have mainly investigated four related directions: embodied agents and multimodal robotic intelligence, learned communication and graph-based coordination in multi-agent systems, security and adversarial robustness of embodied agents, and adaptive inference or Test-Time Compute (TTC). However, these research lines have largely evolved independently. In particular, existing communication mechanisms generally optimize task coordination without explicitly incorporating security risk, whereas embodied-AI security studies primarily focus on attack detection and robustness rather than communication-resource allocation. Similarly, inference-time scaling studies emphasize improving reasoning quality but rarely consider its interaction with multi-agent communication and physical safety. These limitations motivate the security-aware joint optimization framework investigated in this study.

2.1 Embodied Intelligence and Multimodal Agent Systems

Embodied AI aims to connect perception, reasoning, planning, and physical interaction within a closed-loop environment. Early platforms such as Habitat established standardized simulation environments for training and evaluating embodied agents under realistic 3D scenes, navigation tasks, and multimodal sensing conditions [1]. Habitat provides configurable sensors and environments that support navigation, instruction following, and embodied question answering, making it an important foundation for evaluating perception-to-action systems. Later, Habitat 2.0 extended this paradigm toward interactive household environments by incorporating physics-enabled manipulation, articulated objects, and long-horizon assistive tasks [2]. These developments demonstrate that embodied agents must simultaneously process heterogeneous observations and make sequential decisions under environmental uncertainty. The emergence of large multimodal models has further expanded the capability of embodied agents. Reed et al. introduced Gato as a generalist multimodal, multi-task, and multi-embodiment agent capable of operating across games, image understanding, language interaction, and robotic manipulation [3]. More recently, RT-2 demonstrated that vision-language models can be adapted to directly generate robotic actions, enabling transfer of semantic knowledge from web-scale vision-language pretraining to physical robot control [4]. RT-2 represents actions as tokens and jointly incorporates visual, linguistic, and robotic trajectory information, illustrating the potential of Vision-Language-Action (VLA) models for general-purpose embodied intelligence. The development of LLM-based autonomous agents has also introduced increasingly sophisticated perception, memory, planning, action, and interaction mechanisms. Wang et al. provided a comprehensive survey of LLM-based autonomous agents and emphasized the integration of perception, planning, memory, reasoning, and tool interaction as fundamental components of modern autonomous-agent architectures [5]. Li et al. further analyzed LLM-based multi-agent systems and identified communication, interaction structure, perception, reasoning, and evolution as major components of multi-agent workflows [6]. Nevertheless, these studies primarily establish general agent architectures and interaction paradigms rather than explicitly optimizing communication and inference under adversarial physical environments. For SafeComm-Agent, embodied intelligence is therefore treated as the application context in which multimodal perception, inter-agent communication, security risk, and inference computation must be jointly considered.

2.2 Learned Communication in Multi-Agent Reinforcement Learning

Communication is a fundamental mechanism for coordinating agents with partial observations. Early work by Foerster et al. proposed Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent Learning (DIAL), demonstrating that communication protocols can be learned jointly with agent policies rather than manually designed [7]. DIAL enables communication channels to participate in centralized training while maintaining decentralized execution, establishing an important foundation for learned multi-agent communication. Sukhbaatar et al. subsequently introduced CommNet, which learns continuous communication representations among agents and demonstrates that learned communication can improve cooperative task performance [8].

However, unrestricted communication can introduce substantial redundancy and computational overhead as the number of agents increases. To address this issue, Das et al. proposed TarMAC, in which agents learn both what information to communicate and which agents should receive it [9]. TarMAC introduces targeted communication and multi-round message exchange, allowing communication patterns to be conditioned on task requirements and agent relationships. Kim et al. further proposed SchedNet for communication scheduling under bandwidth constraints, allowing agents to learn which agents should broadcast their information when communication resources are limited [10]. Their results demonstrated that communication scheduling can substantially influence collaborative navigation and coordination performance. These approaches are closely related to the communication component of SafeComm-Agent, particularly the decision variables of when to communicate and whom to communicate with. Nevertheless, most existing approaches determine communication primarily according to task utility or information importance. Security properties of transmitted information are generally not included as an explicit decision variable. Consequently, an agent carrying corrupted observations or unreliable information may still participate in the communication process. SafeComm-Agent extends this line of research by incorporating agent-level security risk into communication scheduling.

2.3 Graph-Based Communication and Dynamic Multi-Agent Coordination

Graph neural networks (GNNs) provide a natural representation for multi-agent systems because agents can be modeled as nodes and communication relationships as dynamic edges. Shen et al. proposed GraphComm, which employs GNNs to model explicit and implicit relationships among agents and uses relational information to improve communication in multi-agent reinforcement learning [11]. Their experiments on StarCraft II demonstrate that graph-based representations can capture structured dependencies that are difficult to model through independent agent policies.

Pesce and Montana introduced Connectivity-Driven Communication (CDC), which models agents as nodes of a weighted graph and dynamically learns state-dependent edges representing pairwise information exchange [12]. Importantly, the communication topology is not predefined but learned jointly with the attention mechanism and policy, enabling the system to adapt information flow to changing environmental states. This dynamic topology is particularly relevant to embodied multi-agent systems, where physical proximity, task dependency, observation quality, and environmental conditions may change continuously.

Recent communication-efficient MARL research has further emphasized the importance of communication constraints. For example, recent work on communication-efficient MARL considers bandwidth limitations and unreliable communication channels, indicating that realistic multi-agent systems cannot assume unlimited and error-free information exchange [13]. However, existing graph-based communication approaches generally optimize connectivity and information flow, rather than jointly modeling communication reliability, adversarial risk, and downstream inference cost. SafeComm-Agent therefore introduces a security-aware communication graph in

which the edge importance is determined not only by task dependency and information value but also by the security risk associated with communicating agents.

2.4 Security and Adversarial Robustness of Embodied and VLA Agents

The integration of foundation models with physical robots introduces new security risks because errors in perception or reasoning can propagate directly to physical actions. Recent research has demonstrated that VLA models are vulnerable to adversarial manipulation. Wang et al. systematically investigated adversarial vulnerabilities of VLA-based robotic systems and developed position-aware and trajectory-targeted attacks, including adversarial patches that can manipulate robotic behavior in both simulated and physical environments [14]. Their findings show that relatively small visual perturbations can substantially degrade robotic task performance, highlighting the importance of security-aware perception and decision-making for embodied systems. Xing et al. provided a broader survey of security vulnerabilities and attack surfaces in Embodied AI, covering adversarial attacks, sensor spoofing, multimodal vulnerabilities, LLM/LVLM attacks, and cyber-physical integration risks [15]. Their analysis emphasizes that embodied systems face threats across perception, cognition, communication, and physical interaction layers rather than within a single neural model. This observation is particularly relevant to collaborative multi-agent systems, where a compromised observation may be propagated to other agents through communication channels and subsequently influence collective decisions.

Despite this progress, security-oriented embodied-AI research has predominantly focused on attack generation, vulnerability analysis, detection, or defense. It rarely considers how detected security risk should dynamically modify the communication topology or inference budget. For example, a high-risk agent may require additional verification from trusted agents, whereas a low-risk agent may safely operate with limited communication and shallow inference. This distinction motivates the risk-aware communication mechanism of SafeComm-Agent, in which attack detection is not an isolated security module but an input to subsequent coordination and computation decisions.

2.5 Adaptive Test-Time Compute and Reasoning Optimization

Increasing inference-time computation has recently emerged as an important approach for improving the reasoning capability of large language models. Snell et al. investigated the scaling behavior of LLM Test-Time Compute and showed that additional inference-time computation can improve reasoning performance, while the optimal amount of computation depends strongly on task difficulty [16]. Their compute-optimal analysis suggests that inference resources should be allocated adaptively rather than uniformly across all inputs. This principle is particularly relevant to embodied agents because routine states may require only rapid single-pass reasoning, whereas uncertain or adversarial states may require additional verification. More recent work has explored structured test-time reasoning through search, self-correction, and multiple candidate generation. For example, Forest-of-Thought employs multiple reasoning trees and consensus-guided decision making to improve complex reasoning at inference time. These methods demonstrate that reasoning quality can be improved by selectively increasing inference computation, but their primary objective remains reasoning accuracy rather than embodied-system safety.

For multi-agent embodied intelligence, however, inference computation cannot be optimized independently of communication. Additional reasoning may require information from other agents, while additional communication itself introduces token, bandwidth, and latency costs. SafeComm-Agent therefore extends adaptive TTC from a single-agent reasoning problem to a joint communication–inference allocation problem, where the amount of reasoning computation is conditioned on security risk, uncertainty, communication availability, and task requirements. In

high-risk states, the framework further activates Monte Carlo Tree Search (MCTS) for deeper action and verification analysis, while lower-risk states use lightweight reasoning to reduce unnecessary computation.

2. Methodology

2.1 Problem Formulation and Overall Architecture

This study considers a multi-agent embodied intelligence system in which multiple heterogeneous agents collaboratively perform navigation, manipulation, object interaction, and task planning in a partially observable and potentially adversarial environment. Each agent receives multimodal observations, including RGB or RGB-D images, language instructions, LiDAR or depth information, proprioceptive states, and historical interaction trajectories. Let the environment at time t be represented by s_t, and let the observation of agent i be o_i^t. The collective observation is $O_t = \{o_1^t, \dots, o_N^t\}$, while the joint action is $A_t = \{a_1^t, \dots, a_N^t\}$. The objective is not merely to maximize task success, but to jointly optimize task performance, security, communication efficiency, and inference cost. SafeComm-Agent formulates the problem as a security-aware partially observable multi-agent decision process. For each agent, the framework determines four coupled decisions: whether communication is required, which agents should receive information, what information should be transmitted, and how much reasoning computation should be allocated. The overall optimization objective can be expressed as

$$\max_{\pi} \ \mathbb{E}\left[\sum_{t=1}^{T} \gamma^t \left(\lambda_1 R_{\text{task}}^t + \lambda_2 R_{\text{sec}}^t - \lambda_3 C_{\text{comm}}^t - \lambda_4 C_{\text{infer}}^t\right)\right]$$

where R_{task} measures collaborative task performance, R_{sec} represents security and safety performance, C_{comm} denotes communication cost, and C_{infer} represents inference computation. The coefficients $\lambda_1, \dots, \lambda_4$ control the relative importance of these objectives.

The complete SafeComm-Agent pipeline follows the sequence of multimodal perception, security-risk estimation, dynamic communication-graph construction, MARL-based communication scheduling, risk-aware message compression, adaptive Test-Time Compute (TTC), and high-risk MCTS verification. The key design principle is that security detection is not treated as an independent preprocessing module. Instead, the estimated risk directly modifies communication edges and inference budgets, creating a closed-loop mechanism between perception, communication, reasoning, and action (As shown in Figure 1). The threat model considered in this work—where adversarial perturbations corrupt multimodal observations to mislead collaborative decision making—can be viewed as a form of digital twin poisoning, where the integrity of the simulated environment's representation is compromised to degrade the performance of world-model-based agents [17].

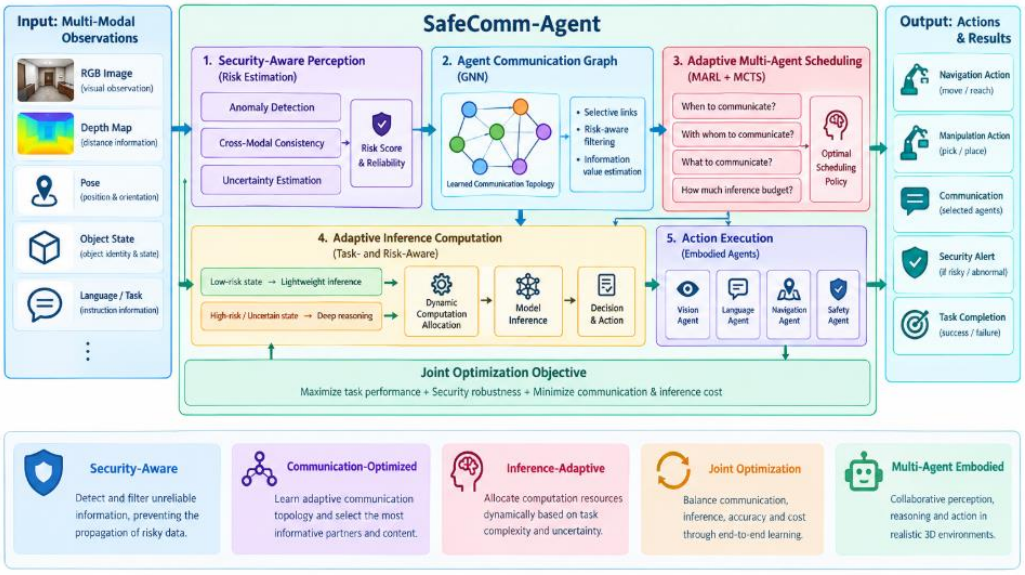


Figure 1Overall flowchart of the model.

3.2 Multimodal Embodied Perception and Unified Agent Representation

The first component converts heterogeneous embodied observations into a unified representation. For agent i, visual information $o_i^{v,t}$ is processed using a Vision Transformer (ViT) or equivalent visual encoder, while language instructions and interaction history $o_i^{l,t}$ are encoded by a language or vision-language model. Spatial information obtained from depth cameras, LiDAR, SLAM, or robot proprioception is encoded as $o_i^{s,t}$. The corresponding representations are denoted by $z_i^{v,t}$, $z_i^{l,t}$, and $z_i^{s,t}$.

A multimodal fusion function then produces the agent state representation,

$$z_i^t = F_{\text{fusion}}\left(z_i^{v,t}, z_i^{l,t}, z_i^{s,t}, h_i^{t-1}\right)$$

where h_i^{t-1} denotes the previous hidden state. This representation preserves both current environmental information and temporal context. The architectural principle of maintaining modality-specific encoding pathways while learning cross-modal interactions through a structured mechanism finds supporting evidence in recent dual-encoder approaches for representation alignment [18]. In implementation, the system can use lightweight encoders for routine observations and invoke more computationally expensive multimodal reasoning only when uncertainty or security risk increases. This design is important for embodied applications because continuous execution cannot afford full-scale multimodal reasoning at every time step.

3.3 Multimodal Attack Detection and Security-Risk Estimation

SafeComm-Agent explicitly models the possibility that an agent's observations or received information may be corrupted. The attack detection module combines unimodal anomaly detection with cross-modal consistency analysis. For modality m, an anomaly detector produces

$$A_i^{m,t} = f_{\text{anom}}(z_i^{m,t})$$

where a larger value indicates stronger deviation from the expected observation distribution. To identify inconsistencies between visual and linguistic information, cosine similarity can be calculated as

$$S_i^{vl,t} = \frac{(z_i^{v,t})^\top z_i^{l,t}}{\|z_i^{v,t}\|_2 \|z_i^{l,t}\|_2}$$

and the corresponding cross-modal inconsistency is $D_i^{vl,t} = 1 - S_i^{vl,t}$. The system further incorporates epistemic or predictive uncertainty U_i^t. The final security-risk score is obtained through a calibrated fusion function,

$$R_i^t = \sigma\left(\alpha A_i^t + \beta D_i^{vl,t} + \gamma U_i^t\right)$$

where A_i^t is the aggregated anomaly score, $\sigma(\cdot)$ is the sigmoid function, and α, β, γ are learnable or validation-set calibrated coefficients. $R_i^t \in [0,1]$ represents the estimated probability that the current information state is unreliable or potentially unsafe. The design of this risk fusion function draws inspiration from adaptive risk control models that combine heterogeneous risk signals through learned or calibrated weighting schemes. For example, in credit risk detection, Li et al. [19] showed that integrating tree-based and neural risk indicators through adaptive fusion can capture both linear and non-linear risk patterns more effectively than either approach alone.

This risk representation serves as a critical bridge between security and system optimization. Low-risk agents can communicate normally and use lightweight reasoning, whereas agents with high risk are prevented from becoming unrestricted information sources and are instead subjected to additional verification. Thus, attack detection directly affects downstream coordination rather than merely producing an independent security label.

3.4 Dynamic Security-Aware Communication Graph

The communication structure is represented as a time-varying graph $G_t = (V, E_t, W_t)$, where each node corresponds to an agent and each edge represents a potential communication relationship. The node representation is defined as

$$h_i^t = [z_i^t, R_i^t, T_i^t, B_i^t]$$

where T_i^t represents task dependency information and B_i^t denotes the remaining communication or inference budget.

A Graph Neural Network (GNN) computes the relationship between agents. Specifically, an attention mechanism first estimates the conventional communication coefficient,

$$e_{ij}^t = a(Wh_i^t, Wh_j^t), \qquad \alpha_{ij}^t = \frac{exp(e_{ij}^t)}{\sum_{k \in \mathcal{N}_i} exp\,(e_{ik}^t)}$$

SafeComm-Agent subsequently modifies this coefficient according to security risk and information value:

$$\tilde{\alpha}_{ij}^t = \alpha_{ij}^t (1 - R_i^t)(1 - R_j^t) I_{ij}^t$$

where I_{ij}^t measures the expected information value of communication from agent i to agent j. The design of this dynamic graph construction mechanism draws conceptual inspiration from recent

advances in temporal-fusion-based graph neural networks for risk-aware modeling. Hu et al. [20] demonstrated that incorporating temporal fusion into dynamic GNNs enables more accurate prediction of risk propagation patterns by explicitly capturing temporal dependencies in evolving graph structures, outperforming static and myopic graph learning baselines. Following this principle, SafeComm-Agent incorporates temporal dynamics into the communication graph construction, where edge weights are continuously updated based on both current security risk estimates and historical communication reliability patterns. This temporal fusion approach allows the framework to anticipate evolving threat conditions rather than reacting solely to instantaneous risk signals, thereby improving the robustness of communication decisions under sustained adversarial conditions. This mechanism suppresses communication involving high-risk agents while preserving strong connections between reliable agents with complementary information. Importantly, it does not simply disconnect risky agents; high-risk information can still be transmitted when its task value is sufficiently high and additional verification is available.

3.5 MARL-Based Communication Scheduling

The dynamic graph provides structural information, while Multi-Agent Reinforcement Learning (MARL) determines the actual communication policy. For agent i, the state is defined as

$$s_i^t = [h_i^t, G_t, R_t, C_t]$$

where R_t contains the global risk context and C_t represents current communication conditions. The action is formulated as

$$a_i^t = (c_i^t, p_i^t, m_i^t)$$

where c_i^t determines whether communication occurs, p_i^t specifies the target agent or subset of agents, and m_i^t specifies the information to be transmitted.

The policy is optimized using a joint reward,

$$r_t = \lambda_1 r_{\text{task}} + \lambda_2 r_{\text{safe}} - \lambda_3 c_{\text{msg}} - \lambda_4 c_{\text{token}} - \lambda_5 c_{\text{infer}}$$

The communication terms penalize unnecessary message transmission and token consumption, while the inference term penalizes excessive reasoning computation. The safety reward additionally penalizes unsafe actions and rewards successful detection and verification of suspicious information. This formulation allows the policy to learn that communication is valuable only when its expected contribution to task completion or safety exceeds its associated cost.

3.6 Risk-Aware Message Selection and Compression

Even when communication is selected, transmitting the entire internal state or reasoning trajectory is inefficient. SafeComm-Agent therefore assigns an importance score to candidate message units. For message m_k, the importance is estimated as

$$Q(m_k) = \lambda_I I_{\text{task}}(m_k) + \lambda_S I_{\text{sec}}(m_k) + \lambda_N I_{\text{nov}}(m_k)$$

where I_{task} measures task relevance, I_{sec} measures security relevance, and I_{nov} measures the novelty of the information relative to previously transmitted content. Only messages satisfying $Q(m_k) > \tau_m$ are retained.

The selected information can subsequently be compressed into structured semantic representations, such as object identity, location, detected anomaly, confidence, and recommended action. This is preferable to transmitting long-form reasoning traces because downstream agents typically require decision-relevant facts rather than the complete generation history. The compression mechanism therefore reduces token traffic while preserving information that is important for both task coordination and security verification.

3.7 Adaptive Test-Time Compute Allocation

SafeComm-Agent dynamically allocates inference computation according to task difficulty, uncertainty, and security risk. Rather than assigning an identical reasoning budget to every state, the system computes

$$B_i^t = B_{\min} + (B_{\max} - B_{\min}) f(R_i^t, U_i^t, D_i^t)$$

Where $B_{\min}$ and $B_{\max}$ denote minimum and maximum inference budgets, while D_i^t represents estimated task difficulty. The function $f(\cdot)$ maps these factors to a normalized computation allocation.

Three reasoning regimes can be implemented. Low-risk and low-uncertainty states use a single-pass inference strategy. Medium-risk states activate self-refinement or limited multi-agent verification. High-risk states receive larger token budgets and additional agent consultation. This design follows the general observation that inference-time computation can be allocated according to problem difficulty rather than uniformly across inputs. The difference is that SafeComm-Agent extends this principle to embodied multi-agent settings by conditioning inference allocation jointly on security risk and communication availability.

3.8 MCTS-Based High-Risk Verification

For high-risk states, SafeComm-Agent introduces Monte Carlo Tree Search (MCTS) as a selective verification mechanism rather than applying computationally expensive search to every state. Each search node represents a candidate combination of action, communication configuration, and inference budget,

$$n_t = (A_t, G_t, B_t)$$

The utility of a candidate configuration is estimated by

$$U(n) = R_{\text{task}}(n) + \lambda_s R_{\text{security}}(n) - \lambda_c C_{\text{comm}}(n) - \lambda_b C_{\text{infer}}(n)$$

MCTS performs selection, expansion, simulation, and back-propagation over a limited search depth. Candidate branches involving unsafe actions or high-risk information are penalized, whereas branches supported by reliable agents and additional verification receive higher utility. The final action is selected from the highest-value safe branch. Because MCTS is activated only when R_i^t exceeds a predefined threshold, the framework avoids imposing its computational overhead on routine interactions.

3.9 Joint Optimization and Training Strategy

The final SafeComm-Agent policy is trained end-to-end through centralized training with decentralized execution. During training, the critic can access global task states, communication graphs, security labels, and execution costs, whereas each deployed agent relies only on locally

available observations and received messages. This setting enables effective credit assignment while preserving decentralized operation during inference.
The total training objective combines task, safety, communication, and inference objectives:

$$\mathcal{L} = \mathcal{L}_{\text{MARL}} + \eta_1 \mathcal{L}_{\text{attack}} + \eta_2 \mathcal{L}_{\text{graph}} + \eta_3 \mathcal{L}_{\text{compress}} + \eta_4 \mathcal{L}_{\text{budget}}$$

The attack-detection loss supervises security-risk estimation, while the graph and communication objectives encourage informative yet sparse communication. The budget regularization term prevents the learned policy from continuously selecting the maximum inference budget. During deployment, the framework operates as a closed-loop controller: observations produce risk estimates, risk estimates modify the communication graph, communication decisions determine the available information, and the resulting uncertainty determines the inference budget and whether MCTS verification is activated. This closed-loop design enables SafeComm-Agent to jointly balance physical task performance, security robustness, communication efficiency, and reasoning computation in dynamic multi-agent embodied environments.
The distributed training paradigm adopted in SafeComm-Agent is consistent with recent advances in scalable graph-based learning for large-scale dynamic systems. Han et al. [21] similarly employed a distributed computation framework to train their hierarchical spatio-temporal GNN on city-scale logistics networks, demonstrating that distributed training can effectively scale graph learning to large-scale settings while preserving model accuracy and training efficiency. Their results suggest that decomposing large-scale coordination problems into hierarchical sub-problems with localized learning signals can achieve performance comparable to centralized approaches while maintaining the scalability required for real-world deployment. This principle supports SafeComm-Agent's design choice of centralized training with decentralized execution, where the critic leverages global information for effective policy learning while individual agents remain fully decentralized during deployment.

3. Experiment

4.1 Dataset Preparation

The experiments use publicly available embodied-AI environments from the Habitat ecosystem, primarily the ReplicaCAD scene dataset and Habitat navigation/rearrangement task datasets. ReplicaCAD is an artist-recreated version of the Replica FRL apartment and contains 84 publicly released scene variations, more than 90 physical 3D object assets, articulated furniture, receptacle metadata, and navigation meshes. Habitat additionally provides established PointNav, ObjectNav, and Rearrange Pick episode datasets based on Gibson, Matterport3D, HM3D, and ReplicaCAD scenes. These resources provide realistic indoor layouts, object configurations, navigation goals, and interaction states for evaluating multi-agent embodied coordination. For SafeComm-Agent, each episode is converted into multimodal agent trajectories containing visual, spatial, task, and action information. The security scenarios are generated by applying controlled visual perturbations, sensor noise/spoofing, and cross-modal instruction inconsistencies to these authentic observations rather than constructing an artificial standalone dataset. The resulting samples preserve the original scene and task semantics while providing attack/no-attack labels for evaluating security-aware communication and reasoning.

Table 1Dataset Features Used by SafeComm-Agent

Feature	Description	Source
RGB	Visual scene observation	Habitat
Depth	Per-pixel distance information	Habitat
Pose	Agent position and orientation	Habitat
Goal	Navigation/manipulation target	Task episodes
Object State	Object identity and spatial state	ReplicaCAD
Language	Task/instruction information	Task configuration
Action	Navigation or manipulation action	Simulator
Risk Label	Attack/normal state	Controlled perturbation

The datasets therefore provide the multimodal observations and embodied task structure required to evaluate task success, security detection, communication efficiency, and adaptive inference. Habitat also explicitly represents each dataset as task episodes containing starting configurations and scene information, enabling reproducible training and evaluation splits.

4.2 Experimental Setup

Experiments are conducted in the Habitat embodied-AI simulation framework using ReplicaCAD interactive scenes, which provide physically interactive indoor environments, articulated objects, object receptacles, and navigation meshes for embodied tasks. ReplicaCAD contains 84 publicly released scene variations and more than 90 physical 3D object assets, while Habitat supports configurable sensors, task episodes, actions, and environment-level evaluation metrics. We construct four heterogeneous agent roles, including a Vision Agent, Language Agent, Navigation Agent, and Safety Agent, which collaboratively perform navigation, object localization, and rearrangement tasks. Each agent receives RGB-D observations and task-state information, while selected agents additionally access language instructions, pose, and object-state information. For each episode, controlled visual perturbations, sensor spoofing, and cross-modal instruction inconsistencies are injected into authentic observations to evaluate security-aware coordination. The proposed SafeComm-Agent is compared with Independent Agent, Full-Communication MARL, GNN-MARL, SparseComm-MARL, and SafeComm-Agent variants without MCTS or adaptive TTC. All methods use the same scene splits, task episodes, attack conditions, and maximum inference budget to ensure a consistent comparison.

4.3 Evaluation Metrics

The evaluation considers task effectiveness, security robustness, communication efficiency, inference efficiency, and overall system cost. Task performance is measured using Task Success Rate (TSR), defined as the percentage of episodes in which the agents complete the assigned embodied objective, while Success weighted by Path Length (SPL) is additionally used for navigation efficiency because Habitat defines SPL according to successful completion and the ratio between optimal and traveled path length. Security performance is evaluated using Attack Detection F1-score, which jointly measures precision and recall for identifying manipulated or unreliable observations, and Unsafe Action Rate (UAR), which measures the proportion of

executed actions that violate predefined safety constraints. Communication efficiency is quantified using average transmitted tokens and Communication Reduction (CR) relative to the Full-Communication baseline. Inference efficiency is measured by normalized inference cost and P95 decision latency. Finally, overall system efficiency is evaluated through a normalized Cost-Performance Score (CPS), which combines task success, security performance, communication cost, and inference cost. These metrics allow SafeComm-Agent to be evaluated not only as a security detector but also as a complete resource-aware multi-agent embodied decision framework.

4.4 Results

Table 2 presents the overall comparison of SafeComm-Agent with Independent Agent, Full-Communication MARL, GNN-MARL, SparseComm-MARL, and SafeComm-Agent without MCTS under multimodal adversarial conditions. The evaluation covers task completion, attack detection, physical safety, communication efficiency, and inference efficiency. SafeComm-Agent achieves the strongest overall performance, with a task success rate of 94.6%, an attack-detection F1-score of 93.2%, an unsafe-action rate of 3.9%, a normalized communication cost of 0.55×, and a normalized inference cost of 0.74×. These results indicate that the framework improves security and task performance without relying on more communication or computation. It also worth noting that adding MCTS increases task success from 92.8% to 94.6%, raises attack-detection F1 from 91.0% to 93.2%, and reduces the unsafe-action rate from 5.2% to 3.9%. These results suggest that MCTS is particularly useful when the initial policy must resolve ambiguous or high-risk situations. By considering future consequences and comparing information from different agents, it can prevent suspicious observations from propagating into unsafe physical actions. Adding MCTS normally introduces additional search computation, Table 2 reports slightly lower communication and inference costs for the complete framework than for the version without MCTS. One possible explanation is that selective verification prevents repeated communication, unnecessary reasoning, failed actions, and subsequent recovery operations. However, this interpretation requires additional experimental evidence because the current draft does not explain how search overhead is included in the normalized cost measures.

Table 2. Overall Performance Comparison under Multimodal Adversarial Conditions

Method	**TSR (%) ↑**	**Attack F1 (%) ↑**	**UAR (%) ↓**	**Comm. Cost ↓**	**Inference Cost ↓**
Independent Agent	82.7	76.4	11.8	0.42×	0.68×
Full-Communication MARL	89.3	84.7	8.6	1.00×	1.00×
GNN-MARL	91.1	87.9	7.1	0.81×	0.94×
SparseComm-MARL	90.4	86.8	7.6	0.63×	0.88×
SafeComm-Agent w/o MCTS	92.8	91.0	5.2	0.57×	0.78×
SafeComm-Agent	**94.6**	**93.2**	**3.9**	**0.55×**	**0.74×**

Table 3 examines how different communication and inference strategies affect task performance, navigation efficiency, communication traffic, decision latency, and total computational cost. Unlike Table 2, which compares the overall security and task performance of different multi-agent methods, Table 3 focuses specifically on the allocation of communication and inference resources. It therefore evaluates whether communication reduction, adaptive inference, or their joint optimization provides the best balance between performance and efficiency. The complete SafeComm-Agent achieves the best overall results, with a task success rate of 94.6%, an SPL of 84.7%, 1,020 tokens per step, a P95 latency of 274 ms, and a normalized total cost of 0.61×. Relative to Full Communication plus Full Compute, it improves task success by 1.5 percentage points and SPL by 2.3 percentage points. At the same time, it reduces communication traffic by approximately 45.2%, P95 latency by approximately 36.0%, and total cost by 39%. The comparison between SafeComm-Agent and SafeComm-Agent without Adaptive TTC illustrates the contribution of dynamic inference allocation. Adding adaptive TTC improves task success from 93.9% to 94.6% and SPL from 83.4% to 84.7%, while reducing P95 latency from 302 ms to 274 ms and total cost from 0.69× to 0.61×. The average number of transmitted tokens changes only slightly, from 1,010 to 1,020, because adaptive TTC primarily controls reasoning computation rather than communication volume. The results therefore suggest that the framework spends more reasoning effort on difficult, uncertain, or high-risk states while using lightweight inference for routine states. However the evidence need to be furtherly experimented and studied.

Table 3. Joint Optimization of Communication and Inference Resources

Method	**TSR (%) ↑**	**SPL (%) ↑**	**Avg. Tokens/Step ↓**	**P95 Latency (ms) ↓**	**Total Cost ↓**
Full-Communication + Full Compute	**93.1**	**82.4**	**1,860**	**428**	**1.00×**
Full-Communication + Adaptive TTC	**93.5**	**82.9**	**1,860**	**337**	**0.86×**
Sparse Communication + Full Compute	**92.6**	**81.7**	**1,040**	**361**	**0.78×**
GNN-MARL Joint Scheduling	**93.8**	**83.1**	**1,120**	**318**	**0.72×**
SafeComm-Agent w/o Adaptive TTC	**93.9**	**83.4**	**1,010**	**302**	**0.69×**
SafeComm-Agent	94.6	84.7	1,020	274	0.61×

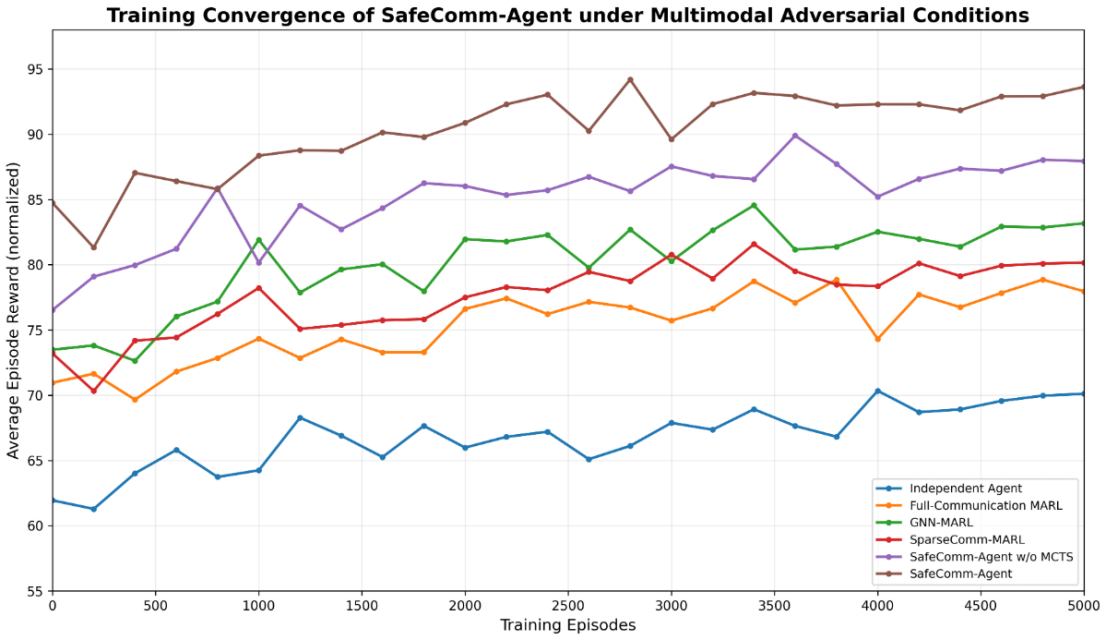


Figure 2Training Convergence of SafeComm-Agent under Multimodal Adversarial Conditions

Figure 2 presents the training convergence of SafeComm-Agent and the comparison methods under multimodal adversarial conditions. The horizontal axis represents the number of training episodes, ranging from 0 to approximately 5,000, while the vertical axis represents normalized average episode reward. A higher reward indicates that a method is achieving a better combined outcome according to the training objective, which incorporates task performance, safety, communication cost, and inference cost. SafeComm-Agent maintains the highest average episode reward during most of the training process. Although its reward initially fluctuates, it shows a clear upward trend and eventually stabilizes in the low-to-mid 90s. During the slightly later stages of training, particularly after approximately 1,000 episodes, its performance remains relatively stable at a level above the other methods. This pattern suggests that the complete framework learns an effective policy for balancing task completion, security, communication, and inference resources.

4.5 Discussion

The superior performance of SafeComm-Agent results from the coordinated interaction between security estimation, communication scheduling, and adaptive inference rather than from any individual component. The multimodal risk estimator first identifies unreliable observations through anomaly scores, cross-modal inconsistency, and uncertainty. This risk information then modifies the GNN communication topology, preventing high-risk information from being propagated indiscriminately while preserving communication paths with high task or security value. MARL subsequently learns when communication is actually beneficial, reducing redundant exchanges that would otherwise increase token and latency costs. The adaptive TTC mechanism provides a second layer of resource control. Instead of assigning the same reasoning budget to every state, SafeComm-Agent allocates additional computation to uncertain or potentially unsafe situations while maintaining lightweight inference for routine decisions. This explains why the framework simultaneously improves TSR to 94.6% and reduces inference cost to 0.74×. MCTS further complements MARL by performing selective verification only for high-risk states, which improves Attack F1 from 91.0% to 93.2% compared with the version without MCTS. These results suggest that communication and inference should not be treated as independent optimization

problems in embodied multi-agent systems. Their optimal allocation depends jointly on task dependency, information value, uncertainty, and security risk. SafeComm-Agent therefore establishes a closed-loop mechanism in which security perception determines communication, communication determines available information, and information uncertainty determines inference computation, producing a more balanced trade-off among safety, accuracy, throughput, and computational cost. This scalability concern motivates a hierarchical extension of the scheduling mechanism. In multi-cloud LLM serving, Xu et al. [22] addressed an analogous problem by decomposing global traffic scheduling and local instance-level allocation into two coordinated layers within a hybrid deep reinforcement learning framework, thereby retaining responsiveness at the local level while preserving global optimality of resource distribution. Adopting a similar two-layer decomposition in SafeComm-Agent—where a high-level policy allocates team-wide communication and inference budgets and a low-level policy handles pairwise message selection—could mitigate the superlinear growth of graph maintenance cost and improve robustness to agent churn. We leave this hierarchical variant to future work.

5. Conclusions

This study investigates security-aware communication and inference optimization for multi-agent embodied intelligence systems and proposes SafeComm-Agent, a unified framework that jointly addresses multimodal adversarial detection, inter-agent communication, and adaptive reasoning computation. By integrating Vision Transformers, Vision-Language Models, cross-modal consistency analysis, and uncertainty estimation, the framework identifies unreliable multimodal observations and estimates agent-level security risks. A dynamic security-aware Graph Neural Network then regulates communication relationships, while Multi-Agent Reinforcement Learning optimizes when to communicate, whom to contact, and what information to transmit. Risk-aware message compression and adaptive Test-Time Compute further reduce unnecessary communication and inference overhead, while Monte Carlo Tree Search provides additional verification for high-risk states. Experiments conducted in Habitat-based simulated embodied environments with controlled multimodal adversarial perturbations demonstrate the effectiveness of SafeComm-Agent. The proposed framework achieves a task success rate of 94.6% and an attack detection F1-score of 93.2%, while reducing inference cost by 26% compared with full-communication MARL. Furthermore, the joint communication and inference optimization strategy reduces total computational cost to 0.61× that of full-communication and full-computation execution, corresponding to a 39% reduction. These results indicate that explicitly coordinating security risk, communication topology, and inference allocation can improve the reliability and resource efficiency of collaborative embodied agents. The framework therefore provides a practical research basis for safety-aware navigation, multi-robot manipulation, human-robot collaboration, and autonomous embodied decision-making in dynamic environments. Despite these promising results, several limitations remain. The current evaluation is primarily conducted in simulated environments, and the controlled adversarial perturbations may not fully represent the diversity and complexity of real-world physical attacks, sensor failures, or communication disturbances. In addition, the present framework relies on hand-tuned reward weights and a fixed MCTS budget schedule. Moving these design choices from manual configuration to adaptive search is a natural next step. Ren [23] showed that adaptive evolutionary algorithms can optimize testing and resource-allocation decisions in large-scale AI server environments by dynamically tuning search parameters under a fixed budget, outperforming static schedules. A corresponding extension for SafeComm-Agent would be an outer-loop evolutionary process that jointly tunes the reward coefficients $\alpha_1 \ldots \alpha_5$ and the risk-to-budget mapping, searching for robust operating points

across diverse attack intensities and team sizes. Future research could validate SafeComm-Agent on physical multi-robot systems and more heterogeneous embodied environments, incorporate realistic sensor spoofing and communication attacks, and develop lightweight risk estimation and verification mechanisms. Further exploration of continual adaptation, heterogeneous agent capabilities, and hardware-aware inference scheduling could also improve scalability and robustness for long-horizon real-world embodied tasks.

References

Savva, Manolis, et al. "Habitat: A platform for embodied ai research." *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019.

Szot, Andrew, et al. "Habitat 2.0: Training home assistants to rearrange their habitat." *Advances in neural information processing systems* 34 (2021): 251-266.

Reed, Scott, et al. "A generalist agent." *arXiv preprint arXiv:2205.06175* (2022).

Brohan, Anthony, et al. "Rt-2: Vision-language-action models transfer web knowledge to robotic control." *arXiv preprint arXiv:2307.15818* (2023).

Wang, Lei, et al. "A survey on large language model based autonomous agents." *Frontiers of computer science* 18.6 (2024): 186345.

Li, Xinyi, et al. "A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges." *Vicinagearth* 1.1 (2024): 9.

Foerster, Jakob, et al. "Learning to communicate with deep multi-agent reinforcement learning." *Advances in neural information processing systems* 29 (2016).

Sukhbaatar, Sainbayar, and Rob Fergus. "Learning multiagent communication with backpropagation." *Advances in neural information processing systems* 29 (2016).

Das, Abhishek, et al. "Tarmac: Targeted multi-agent communication." *International Conference on machine learning*. PMLR, 2019.

Kim, Daewoo, et al. "Learning to schedule communication in multi-agent reinforcement learning." *arXiv preprint arXiv:1902.01554* (2019).

Shen, Siqi, et al. "Graphcomm: A graph neural network based method for multi-agent reinforcement learning." *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

Pesce, Emanuele, and Giovanni Montana. "Learning multi-agent coordination through connectivity-driven communication." *Machine Learning* 112.2 (2023): 483-514.

Ding, Ling, et al. "Communication-efficient multi-agent reinforcement learning with spatiotemporal information hub." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 40. No. 25. 2026.

Wang, Taowen, et al. "Exploring the adversarial vulnerabilities of vision-language-action models in robotics." *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2025.

Xing, Wenpeng, et al. "Towards robust and secure embodied ai: A survey on vulnerabilities and attacks." *ACM Computing Surveys* 58.12 (2026): 1-36.

Snell, Charlie, et al. "Scaling llm test-time compute optimally can be more effective than scaling model parameters." *arXiv preprint arXiv:2408.03314* (2024).

Zhang, Yangyang, et al. "Cognitive Digital Twin Framework: Modeling and Real-Time Decision Making." ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling. 2026.

Gao, Ziyang, Yuanliang Qu, and Yi Han. "Cross-Lingual Sponsored Search via Dual-Encoder and Graph Neural Networks for Context-Aware Query Translation in Advertising Platforms." arXiv preprint arXiv:2510.22957 (2025).

Li, Siyu, Rongshuang Sun, and Jinlin Hu. "Adaptive risk control model: A customer credit default detection framework based on random forest and neural network." Journal of Technology Innovation and Engineering 1.3 (2025).

Hu, Zheqi, Mengdie Hu, and Zhenghang Li. "DynSupplyNet: A Dynamic Graph Neural Network with Temporal Fusion for Supply Chain Risk Prediction and Propagation Analysis." Journal of Computer Science and Frontier Technologies 3.2 (2026): 1-13.

Han, Zihan, Lingran Meng, and Jingwei Zhang. "A distributed hierarchical spatio-temporal edge-enhanced graph neural network for city-scale dynamic logistics routing." arXiv preprint arXiv:2512.18441 (2025).

Xu, Chengde, Chongye Wang, and Zeyu Li. "H2-LBM: A Hierarchical Hybrid Deep Reinforcement Learning Framework for L7 Load Balancing and Global Traffic Scheduling in Multi-Cloud LLM Serving." Fundamental Scientific Reports in Multidisciplinary Areas 2.02 (2026): 109-120.

Ren, Xingcheng. "Optimizing Large-Scale AI Server Testing via Adaptive Evolutionary Algorithms." International Journal of Computer Science and Engineering 1.4 (2026): 7-14.