Multi-Panzer: a LLM-based Multi-agent System for Tank Detachment Combat Simulation

1st Yibo Chen

War Research Institute

PLA Academy of Military Science

Beijing, China

1606082029@stu.sqxy.edu.cn

2nd Yang Ping

War Research Institute

PLA Academy of Military Science

Beijing, China

ocean.py@163.com

***Abstract*—To further expand the adversarial intelligence capability of tank detachment combat simulation, this study designs a large language model-based multi-agent system for tank detachment combat, Multi-Panzer, using a modular design approach. The Multi-Panzer architecture comprises key components including a scenario information converter, combat unit agents, symbolic planner, communication module, and command agent, wherein combat unit agents possess planning and memory structures. Simulation results demonstrate that various large language models (LLMs) can be applied within the Multi-Panzer architecture; Multi-Panzer exhibits superior combat capabilities compared to traditional agents and demonstrates effectiveness across various scenario terrains; the communication module and command agent enhance Multi-Panzer's collaborative abilities; the prompt chaining method and memory bank configuration of combat unit agents ensure logical decision-making and behavioral coherence. The simulation results indicate that enhancing cooperation among combat units in tank detachment combat simulations is more important than strengthening individual cognition. This study provides a novel framework for LLM-based multi-agent systems in tank detachment combat simulation, with its architectural design offering guidance for developing LLM-based multi-agent systems in combat simulation.**

Keywords—Large language models, multi-agent systems, tank detachment combat, combat simulation

I. INTRODUCTION

Agent technologies for computer-based adversarial simulations are highly mature [1]. However, with continuous in-depth research in artificial intelligence, large language models (LLMs), and other fields, technologies such as computer cooperative simulation, social simulation, and multi-agent simulation based on LLMs have developed rapidly, playing an increasingly important role in decision support and other areas [2]. Current rule-based artificial intelligence offers advantages in rapid deployment but is constrained by rules, making it difficult to break through in intelligence level. While data-driven and reinforcement learning-based artificial intelligence excel in processing massive data, their interpretability and model transfer capabilities remain deficient [3-6]. Therefore, the design of new generative artificial intelligence based on the revolutionary technology of LLMs and ensuring that it possesses a certain adversarial intelligence capability [7] is a crucial issue in the combat simulation field.

Imagine that on a rapidly changing tank battlefield, a tank platoon commander must handle dozens of rapidly shifting factors simultaneously amidst roaring artillery fire – terrain masking, enemy situation dynamics, friendly force positions, superior orders, and equipment status, and quickly make coordinated offensive and defensive decisions. This high-collaboration, high-confrontation activity, with its nonlinear decision-making, unpredictability, and irreproducibility, remains a core challenge in combat simulations [8]. Recent breakthroughs in LLMs, particularly their strength in natural language understanding, complex reasoning, and in-context learning [9], offer a novel solution path. Waragent "War and Peace" simulated political behaviors that triggered wars in history using LLM-based multi-agent technology [10], "BattleAgent" simulated battles in the era of cold weapons using LLM-based multi-agent technology [11], and "COA-GPT" generated infantry course of action by combining LLM technology with manual correction methods [12]. The feasibility of applying LLM technology to military science, especially combat simulation, was verified. However, the simulation of modern tank detachment combat using LLM-based multi-agent technology remains a subject to be explored. Crucially, the core of informationized tank detachment combat—perceiving, comprehending, transmitting, deciding, and coordinating battlefield information—can be viewed as a complex "language task." Situation reports, orders, requests, and feedback rely on language or symbolic exchange. By designing structured prompt inputs [13] to translate combat into LLM-understandable text and convert LLM responses into actions, general-purpose LLMs can become tank detachment combat agents. LLM-based multi-agent systems provide a strong social nature, high adaptability, transparent reasoning, convenient debugging, and low interaction barriers compared to traditional AI [14]. This makes them highly promising for addressing tank simulation pain points such as difficult coordination, poor adaptation, and opaque logic, which hold significant research value.

The main work and contributions of this paper:

(1) For the first time, the LLM-based tank detachment combat simulation multi-agent system Multi-Panzer is based on the B/S architecture Army Tactical Adversarial Intelligent Simulation Platform [15]. This system adopts a modular design, providing an extensible framework for the application of LLMs in multi-agent combat simulations.

(2) Through structured prompt engineering and prompt chaining methods, combat information is transformed into text inputs understandable by LLMs, enabling LLMs to perform corresponding logical reasoning and behavioral decision-making as the battle situation changes, thereby achieving effective application of LLMs in tank detachment combat simulation.

(3) Systematic experiments have verified multiple advantages of Multi-Panzer: it possesses adversarial intelligence capabilities superior to those of traditional agents; it exhibits strong adaptability, maintaining high win rates across different terrain scenarios without additional training; and it supports various mainstream LLMs, indicating the architecture's good generalization.

(4) Ablation experiments revealed that in tank detachment combat simulations, enhancing cooperation among combat unit agents through command and communication is more important for improving overall combat effectiveness than strengthening individual combat unit cognition through memory and planning. This finding provides guidance for the design of future multi-agent systems for tank detachment combat, highlighting the critical role of collaborative mechanisms in tank detachment operations.

(5) Transparency of the agent decision-making process has been achieved by leveraging the natural language generation and self-explanation capabilities of LLMs, facilitating review, inspection, and adjustment, thereby enhancing the system's trustworthiness and practicality. This provides a highly interpretable decision support framework for tank detachment combat simulations.

(6) Multi-Panzer can endow agents with "personalities" and tactical preferences through "role information," supports natural language interaction, and expands the application prospects of human-machine cooperative operations. It offers possibilities for personalized agent design and human-computer interaction using LLMs in combat simulations.

The data that support the findings of this study are available in the GitHub repository at https://github.com/c12730/MP

II. Multi-Panzer Multi-Agent System Architecture

Our Multi-Panzer underwent adaptive design for the data interface of the Army Tactical Adversarial Intelligent Simulation Platform to ensure normal operation on this platform. The Army Tactical Adversarial Intelligent Simulation Platform uses a B/S architecture that is capable of describing the main combat actions of primary combat units [16, 17] and supporting auxiliary functions such as command, direction, situation monitoring, and control, making it particularly suitable for tank battle simulation. In this study, the Multi-Panzer occupies the Red Force seat within this simulation platform.

Based on the LLM agent theory, we integrate multiple tank unit agents into a multi-agent system through components such as the scenario information converter and symbolic planner. This multi-agent system enables interactions between agents and between agents and the scenario, enabling agents to obtain key information, communicate with each other, and move within the scenario and attack targets.

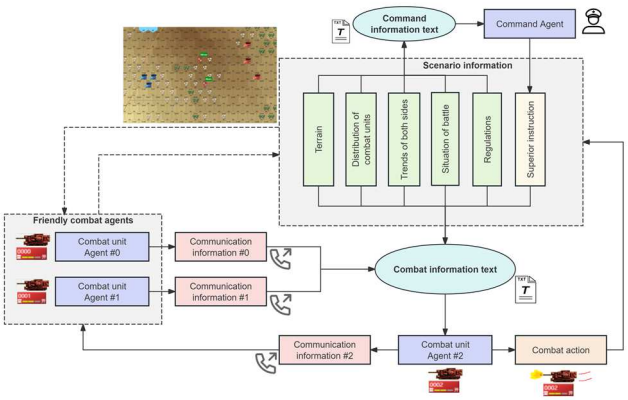

Fig. 1. Closed-loop feedback of Multi-Panzer

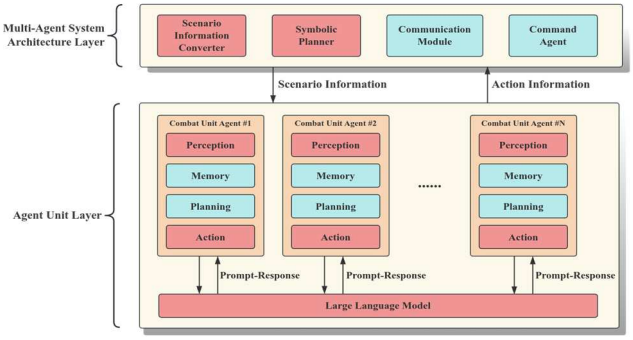

Fig. 2. Multi-Panzer architecture

Fig.1 illustrates the closed-loop feedback flow of scenario → combat information → agent → action → scenario, taking the example of a multi-agent system providing input to and receiving output from "Combat Unit Agent #2." Fig.2 shows the architecture of the Multi-Panzer. The simulation of tank detachment combat is realized through interactions between individual combat unit agents and the scenario, other friendly combat agents, and command agent. The system structure includes the following.

(1) Scenario Information Converter: The ever-changing scenario is crucial for tank detachment combat simulations. The scenario contains various factors, such as terrain, combat unit distribution, movements of friendly and enemy combat units, battle status, doctrines, and superior instructions. Key information from these factors needs to be converted by the scenario information converter according to specific rules into natural language text that is understandable by LLMs for decision-making by the command agent and combat unit agents. Scenario information is public to all agents, equivalent to assuming the timely sharing of battlefield situational awareness. The scenario information converter is a fundamental functional module; without it, the Multi-Panzer cannot operate normally.

(2) Combat Unit Agent: The tank platoon is the main body of the tank detachment combat. A combat unit agent corresponds to a Red Force tank platoon unit in the simulation platform. Each combat unit agent includes modules, such as perception, memory, planning, and action. These modules form a closed loop according to specific rules to achieve a credible action output. Combat unit agents are fundamental functional modules; without them, Multi-Panzer cannot operate normally.

(3) Symbolic Planner: In convert the behavioral decisions of the combat unit agents into various actions within the scenario, we incorporated a symbolic planner as the interface for the interaction between each combat unit agent and the scenario. This module is responsible for extracting behavioral decisions from the agent's response text, and through programming, converts the agent's behavioral decisions into various actions performed by the agent within the scenario. The symbolic planner is a fundamental functional module; without it, Multi-Panzer cannot operate normally.

(4) Communication Module: To demonstrate the collaborative communication capabilities between tank platoons in tank detachment combat, we incorporated a communication module to enable communication between the combat unit agents. To prevent a situation in which two combat unit agents simultaneously send requests to each other, causing both agents to act in "synchronized oscillations" to fulfill the other's request and lose coordination, we stipulate that at any given moment, only one agent can select another agent to send a message. This message is delivered immediately to the target agent and participates in the subsequent decision-making of the target agent. Because scenario information is public to all agents, interactions between agents do not need to transmit battlefield information redundantly, but primarily focus on stating intentions, sending requests, and issuing instructions. The communication module is an advanced functional module; without it, there is no communication between combat unit agents.

(5) Command Agent: In reflect the characteristics of tactical commands in tank detachment combat, we incorporated a command agent. The command agent is primarily responsible for issuing combat instructions to combat agents and receiving feedback from scenario information; this corresponds to the Red Force commander seat in the simulation platform. The interaction between the command agent and combat unit agents is unidirectional, with the content being the combat instructions generated by the command agent. The command agent is an advanced functional module; without it, combat unit agents can only attempt to organize combat autonomously.

III. Structure of Multi-Panzer Unit Agents

A. Large Language Model Selection

As the most critical decision engine for LLM agents, we primarily selected the LLM Qwen-Long as the cognitive core of the agent. Qwen-Long was optimized for ultra-long context processing scenarios. It supports direct Chinese input and ultra-long context dialogues of up to 10 million tokens (approximately 15 million Chinese characters or 15000 pages of documents) [18], making it highly suitable for combat simulation scenarios with substantial information volumes. Qwen-Long also conveniently allows the input of role, system, and user information to accomplish combat tasks more professionally. We also experimented with other LLMs, with varying results, as detailed in the experimental section of this paper.

B. Combat Unit Agent Structure Design

To ensure behavioral consistency in combat unit agents, agents must possess memory, and even short-term memory is sufficient to guarantee coherent actions. To facilitate the LLMs in understanding and solving problems, it is preferable to decompose a large problem into smaller sub-tasks. Therefore, we adopted the prompt chaining method [19], dividing the agent's action decision into two sub-tasks: formulating a short-term plan based on combat information, basic information, and memory bank information; then, we analyzed the current action to be taken by combining combat information, the short-term plan, basic information, and memory bank information, and outputting the response. Both the resulting short-term plan and the current action are stored in the memory bank. The structure of the combat unit agent is shown in Fig.3. Here, "basic information" includes scenario terrain line-of-sight rules, combat simulation rules, and the capabilities of combat unit agents; the "memory bank" contains historical events, the latest plan and action taken. "Determining Action" and "Basic Information" are foundational functional modules that maintain normal agent operation, while the "Memory Bank" and "Formulating Plan" are advanced functional modules that enhancing the cognitive capabilities of combat unit agents.

Each action decision of the agent requires several rounds of prompting and response-based logical reasoning based on external environmental input and its own memory, which can also be described as guiding LLM through a step-by-step thinking process [20]. Therefore, structured prompt inputs are indispensable for an agent's LLM core to function as intended. Typically, executing an accurate prompt for an LLM requires three types of prompt input: "role information," "system information," and "user information" [21].

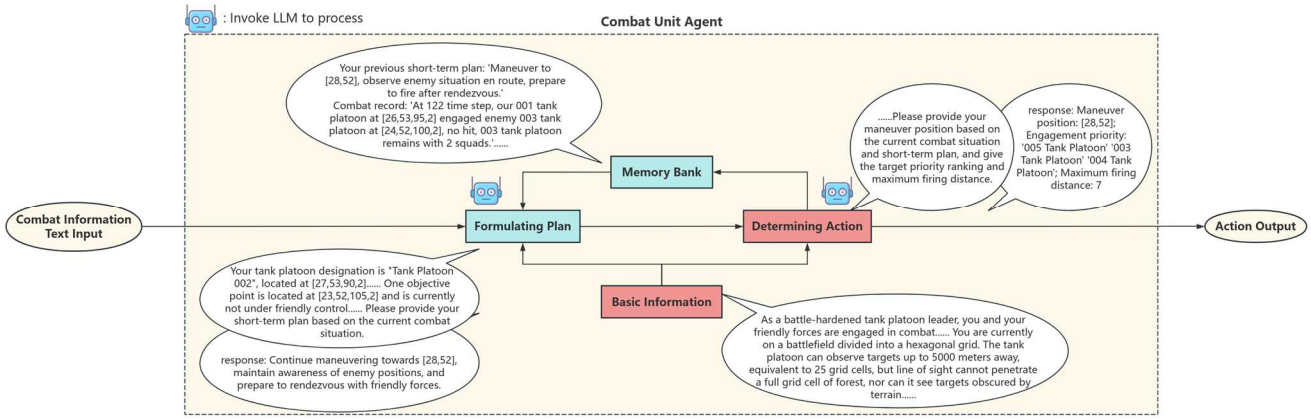

Fig. 3. Structure of combat unit agents

In the "Formulating Plan" step of the combat unit agent: Basic information such as scenario terrain line-of-sight rules, combat simulation rules, and the capabilities of the combat unit agent are input as "role information." The combat information is input as "system information." Finally, the historical events from the current agent's memory bank, along with the previous round's short-term plan and action, are integrated into "user information" to pose a question regarding the agent's required update to its short-term plan. The obtained response serves as input for the subsequent "Determining Action" step and is submitted to the memory bank for storage.

In the "Determining Action" step of the combat unit agent: It is similarly necessary to input basic information such as scenario terrain line-of-sight rules, combat simulation rules, and the capabilities of the combat unit agent as "role information," input combat information as "system information," and finally integrate the historical events from the current agent's memory bank, this round's short-term plan, and the previous action into "user information" to pose a question regarding the agent's specific current action. The obtained response serves as the combat action output for interaction with the scenario and is submitted to the memory bank for storage.

The expressions corresponding to the above prompt chains are respectively:

$$P_t = f(I_{role}, I_{combat}, p_{plan} + P_{t-1} + A_{t-1} + M) \quad (1)$$

$$A_t = f(I_{role}, I_{combat}, p_{decide} + P_t + A_{t-1} + M) \quad (2)$$

Where I_{role} is the basic information input, I_{combat} is the combat information input, P_{t-1} is the short-term plan generated in the previous time step, P_t is the output of this round's Formulating Plan, p_{plan} is the prompt requiring the LLM to formulate a plan, p_{decide} is the prompt requiring the LLM to determine an action, A_{t-1} is the action generated in the previous round, A_t is the output of this round's "Determining Action," and M is the historical event in the memory bank.

C. Command Agent Design

The command agent is a simplified version of the combat unit agent. Compared to the structure of the combat unit agent, the command agent merges the "Formulating Plan" and "Determining Action" Modules into a "Issue Command" Module. This is because the command agent itself is not located in the scenario, and its "short-term plan" typically corresponds directly to the command it needs to issue.

To ensure that the command agent makes decisions based on the objective battlefield situation, we specified in the prompt that it must analyze the current situation using basic information, memory bank information, and command information before issuing commands that are publicly broadcast to all agents.

IV. Simulation Experiment

The simulation experiments were based on the Army Tactical Adversarial Intelligent Simulation Platform, which uses a B/S architecture and has a public web interface. This platform is dedicated to army combat simulations and possesses numerous scenarios and terrains for tank detachment combat. The platform employs engagement adjudication rules based on random numbers and adjudication tables, similar to the common army combat simulations. Different terrains affect the units' line of sight and movement speed; factors such as the terrain a unit occupies, the unit's own status, and the target distance all influence engagement adjudication outcomes. Apart from the basic movement rules, reconnaissance rules, and engagement rules, no other special mechanisms exist in this platform.

The primary scenario for the simulation experiments was a typical tank detachment seize-control task (terrain name: "Urban Residential Area," see Fig.4). The red force is the tank detachment, consisting of three tank platoons that we need to test with our agent. The blue force is the rule-based AI agent "DEMO_AI" within the simulation platform, which possesses a certain level of capability. DEMO_AI has dynamic adjustment capabilities and employs three strategic rules: strike, advance, and encirclement. In each match, DEMO_AI randomly implemented one of these strategies.

We used the agent's win rate and the net score defined by the Army Tactical Adversarial Intelligent Simulation Platform as evaluation metrics. The net score calculation rules are as follows: the net score starts at 0 at the beginning, destroying an opponent's tank platoon (composed of two tanks each worth 20 points) adds 40 points to one's own score, while losing one's own tank platoon deducts 40 points; at the end of the match, controlling a "Minor" point adds 50 points, and controlling a "Main" point adds 80 points. A net score greater than 0 was considered a victory. Based on the net score results, the win rate, the average net score, and the median net score can be calculated — all three metrics reflect the agent's adversarial intelligence capability.

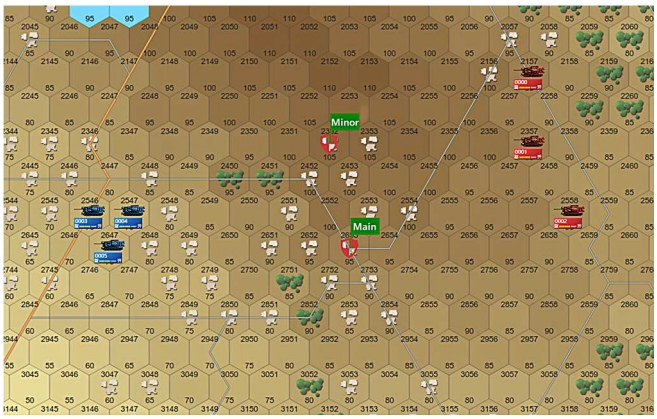

Fig. 4. Experiment scenario

Except for generalization tests, our Multi-Panzer uses Qwen-Long as the agent's cognitive core; except for adaptability tests, we use "Urban Residential Area" as the scenario terrain. Since this research employs LLMs as the cognitive core, leveraging their high generalization capability, we utilized no training datasets throughout the study. The main program of Multi-Panzer in this study runs on a Ubuntu 20.04 virtual machine, with hardware configuration consisting of a 13th Gen Intel(R) Core(TM) i9-13900HX processor at 2.20 GHz and 64GB of RAM.

A. Baseline Test

In the baseline test, we must compare the match results when Multi-Panzer and other different types of agents serve as the red force. We include DEMO_AI as a baseline because it is a representative rule-based agent and is identical to the agent used by the blue force, which helps illustrate the balance of this scenario for both red and blue forces. We selected the

reinforcement-learning-based agent "AlphaWar" as another baseline [22]. Specifically designed for tank detachment combat simulation, its performance has reached the state-of-the-art (SOTA) level in this field, and it is a fully trained out-of-the-box agent. Each agent served as a red force for 100 runs. The test results are shown in Fig.5 and Table I.

The experimental results show that the win rate of DEMO_AI is only 45%, with an average net score of -18.8. This indicates that the scenario terrain is slightly unfavorable to the red force. The rule design of DEMO_AI is very sophisticated, and it can sometimes even defeat reinforcement-learning-based agents. AlphaWar achieved a 71% win rate against DEMO_AI, with an average net score of 83.0. Our Multi-Panzer achieves an 80% win rate with an average net score of 127.4, and easily defeats DEMO_AI. This demonstrates that our Multi-Panzer possesses a strong adversarial intelligence capability.

We conducted a head-to-head (H2H) comparison between Multi-Panzer (red) and AlphaWar (blue). Our Multi-Panzer achieved a 60% win rate with an average net score of 42.8, indicating that our Multi-Panzer outperformed AlphaWar.

TABLE I. BASELINE TEST RESULTS

Agent	Win Rate	Average Net Score	Median Net Score
DEMO_AI	45%	-18.8	-160
AlphaWar	71%	83.0	170
Multi-Panzer	80%	127.4	210

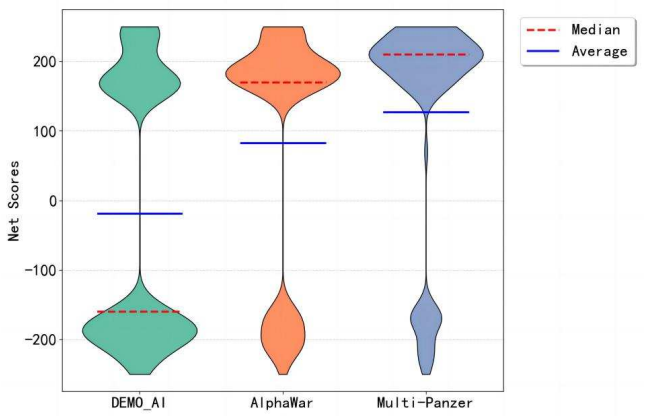

Fig. 5. Net score distribution of different agents

B. Adaptability Test

In the adaptability test, we need to compare the match results of the Multi-Panzer in entirely new scenario terrains. We conducted simulation experiments for various scenarios provided by the simulation platform. The terrain scenario used is shown in Fig.6, where terrain-a,b,c, and d are sequentially: "Urban Residential Area" with extensively distributed buildings and roads, "Paddy Fields with Water Networks" where widespread soft ground causes significant reductions in tank mobility speed, "Plateau Pass" with steep terrain and sparse cover, and "Mountainous Jungle" with undulating terrain and dense forests. A total of 100 tests were conducted for each scenario terrain, and the test results are shown in Table II and Fig.7.

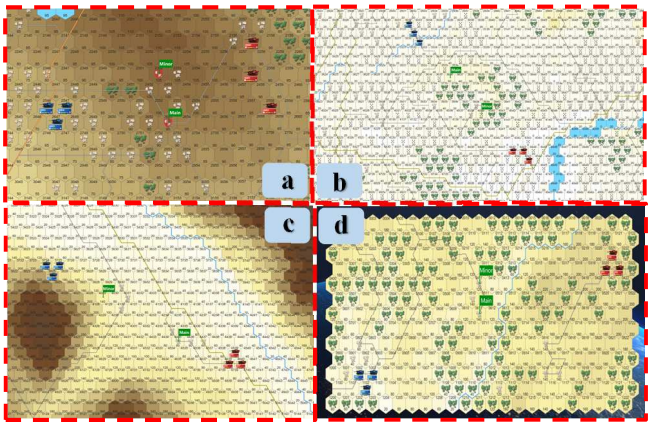

Fig. 6. Overview of scenario terrains

TABLE II. ADAPTABILITY TEST RESULTS

Scenario Terrain	Win Rate	Average Net Score	Median Net Score
Terrain-a	80%	127.4	210
Terrain-b	78%	129.6	210
Terrain-c	80%	121.7	210
Terrain-d	79%	131.4	210

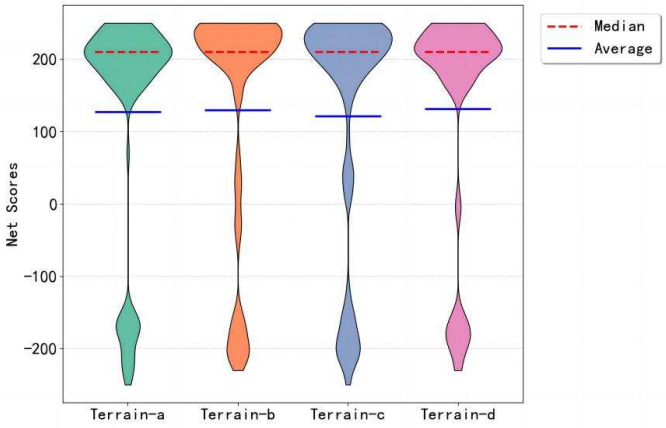

Fig. 7. Net Score distribution across different terrains

The experimental results show that Multi-Panzer can achieve high average net scores and win rates across different terrains, although the distribution of net scores is slightly affected by different terrains. This indicates that the Multi-Panzer possesses good adaptability and can adapt to entirely new scenario terrains without requiring additional training.

C. Generalization Test

In the generalization test, we compared the performance of the Multi-Panzer when employing different LLMs. We tested the Multi-Panzer using ERNIE-4.5, GPT-3.5, DeepSeek-V3, and Llama-3 and compared them with the Multi-Panzer using Qwen-Long. Each LLM was run for 100 tests, and the results are presented in Fig.8 and Table III.

The experimental results show that when Multi-Panzer employs different LLMs, the performance with Llama-3 and ERNIE-4.5 is relatively poor, with win rates between 60%-70%; whereas the performance with GPT-3.5, DeepSeek-V3, and Qwen-Long is relatively better, with win rates between 70%-80%. We believe that the cognitive mechanisms of different LLMs may affect the actual performance of the Multi-Panzer. Overall, Multi-Panzer using different LLMs achieves win rates exceeding 60%, and the average net scores

are all positive. Therefore, we consider that the Multi-Panzer architecture demonstrates a degree of generalization.

TABLE III. GENERALIZATION TEST RESULTS

LLM	Win Rate	Average Net Score	Median Net Score
Qwen-Long	80%	127.4	210
ERNIE-4.5	68%	79.7	200
GPT-3.5	79%	115.6	190
DeepSeek-V3	76%	117.0	190
Llama-3	68%	74.8	170

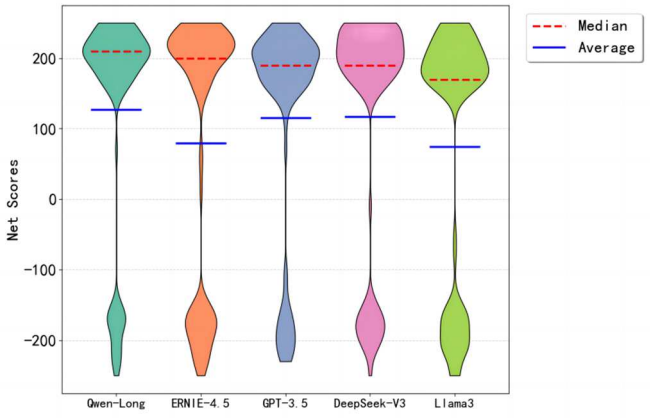

Fig. 8. Net score distribution with different LLMs

D. Ablation Experiment

In the ablation experiment, we need to compare the effects after removing different modules at both the multi-agent system architecture level and the unit agent structure level of Multi-Panzer to verify the rationality of Multi-Panzer's design. Specifically, at the multi-agent system architecture level, we removed the command agent, communication module, and both simultaneously; at the unit agent structure level, we removed the memory bank module, the planning module, and both simultaneously. Each configuration was run 100 times, and the experimental results are presented in Fig.9 and Table IV.

TABLE IV. ABLATION EXPERIMENT RESULTS

Configuration	Win Rate	Average Net Score	Median Net Score
Full Architecture	80%	127.4	210
w/o Command	72%	99.0	190
w/o Communication	70%	95.6	210
w/o Command & Communication	61%	49.8	170
w/o Plan	77%	120.1	190
w/o Memory	75%	113.3	210
w/o Plan & Memory	70%	94.6	190

The experimental results show that both the command agent and communication module contribute to the coordination of combat unit agents, and whether combat unit agents can effectively coordinate is closely related to the combat outcomes. Removing the command agent or communication module alone reduces the win rate by 8%-10%, whereas removing both simultaneously reduces the win rate by 19%. The planning module helps combat unit agents in organizing decision-making logic, providing cognitive enhancement. The memory bank module not only ensures decision-making coherence for combat unit agents, but also enables them to perceive the passage of time. Removing the planning module or memory bank module alone reduces the win rate by 3%-5%, whereas removing both simultaneously reduces the win rate by 10%. Overall, strengthening coordination between combat unit agents is more important than enhancing the individual cognition of combat unit agents. When removing the same number of modules, configurations with "strong coordination but weak cognition" consistently achieved 3%-9% higher win rates than those with "weak coordination but strong cognition". This aligns with the inherent characteristic of high cooperation in tank detachment combat.

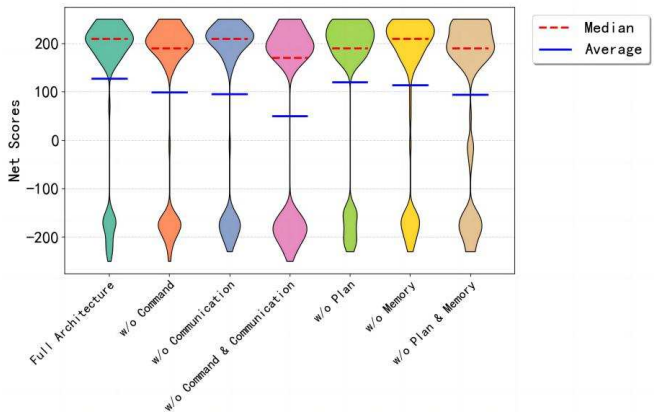

Fig. 9. Net Score distribution with different configurations

V. DISCUSSION

Unlike common combat simulation platforms, the Army Tactical Adversarial Intelligent Simulation Platform adopts an interaction triggering rule based on time steps; the time step of the simulation platform does not advance until all agents complete their decision-making. This mechanism ensures that agents' adversarial intelligence capabilities are not affected by the decision-making duration, thus eliminating the need to impose requirements on the LLMs' response speed. According to experimental records, when using Qwen-Long, the single prompt response duration is approximately 3 s, with a single prompt-response token count of approximately 6k.

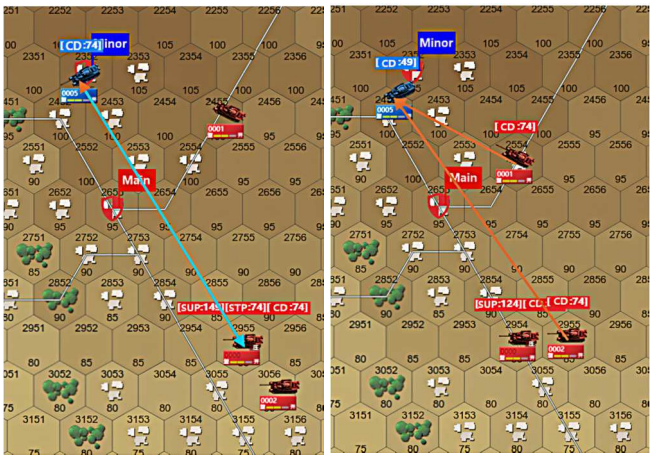

Fig. 10. A typical coordinated action

Regarding Multi-Panzer's interpretability, every action of an LLM-based agent leaves the corresponding prompt and response records. We can easily view the contextual records of an agent's decision at any given moment and input these records into the dialogue window of LLM. Based on this context, we can directly question the model of the reasons and objectives behind this decision. This demonstrates that the agent has good decision transparency and self-explanation

capability, facilitating targeted adjustments and improvements [23].

In simulation experiments, our Multi-Panzer was able to eliminate opponents and occupy all control points in most cases, demonstrating excellent performance in coordination. In a typical coordinated action, as shown in Fig.10, Multi-Panzer (acting as the red force) leveraged the terrain to set up an ambush. After the blue force tank platoon advanced to seize control and attacked the red force's 000 tank platoon, the 001 and 002 tank platoons immediately advanced from concealed positions to firing positions and simultaneously engaged the blue force in a counterattack. This demonstrates that Multi-Panzer plans meticulously, reacts swiftly, and enables timely mutual support between combat units, forming a tight and effective coordination. For the typical coordinated action shown in Fig.10, by leveraging the unique self-explanation capability of LLMs [24, 25], we can reproduce the decision-making environment through context and directly question the agent to understand its reasoning. Taking "001 tank platoon" in the figure as an example:

User: *Please explain the reasons and objectives for maneuvering to this position.*

Qwen-Long: *The selection of (25, 54) as the maneuver destination is based on the following considerations:*

Terrain Advantage: This position, classified as an urban area (Terrain Type 2), provides some cover, reducing the risk of enemy reconnaissance and direct fire strikes. Additionally, it allows effective observation of the movements in the "23, 52" region.

Coordinated Operations: Superior command orders a terrain-based ambush on the target point. The friendly "000 Tank Platoon" has indicated movement toward the "23, 52" target point to assist in defense or eliminate potential enemy forces. Selecting (25, 54) facilitates crossfire coordination with the friendly platoon, enabling effective suppression or elimination of enemy forces in "23, 52."

Support for Other Objectives: Although the friendly "002 Tank Platoon" suggested moving toward "26, 53," the most pressing task is addressing the known enemy threat posed by the "005 Tank Platoon" at "23, 52." Once the enemy threat at this location is confirmed or neutralized, subsequent actions can be adjusted flexibly based on battlefield conditions.

In summary, maneuvering to (25, 54) not only effectively addresses the immediate enemy threat, but also retains flexibility for future operations. [26]

During the agent adjustment process, we observed that the phrasing of battlefield information required a precise design. For instance, when describing the ownership status of a control point, if the prompt stated that the control point as "enemy-controlled," the agent would become hesitant and refuse to advance, even if there were no enemy units left on the field, never daring to seize control. By reproducing the decision-making environment and directly questioning the LLM, we discovered that the agent mistakenly assumed that an enemy-controlled point concealed many enemy troops, making it afraid to enter rashly. This was a misunderstanding of the simulation situation (as a controlled point itself does not contain any troops, it merely changes ownership). The solution was simple: changing the prompt description of the control point status from "enemy-controlled" to "not under friendly control" restored the agent's normal ability to seize control, with no adverse behavior observed after extensive testing. This demonstrates the importance of domain-aligned prompt engineering [27], suggesting its potential value in task-specific LLM adaptations.

Leveraging the unique characteristics of LLMs, in the future, we could use role-information prompts to imbue combat unit agents with "personality traits" and action preferences [28]. For example, setting restrictive combat doctrines, defining their "character" as assertive or supportive, or even providing "The Art of War: Terrain Chapter" as knowledge could theoretically produce corresponding effects in combat simulation. By employing different role-information prompts, we can study the impact of various subjective factors of combat units on combat operations.

Using LLMs as the core of tank detachment agents enables interactions between agents as well as between agents and human commanders/operators to occur directly in natural language. Theoretically, if a human commander or operator replaces a corresponding agent position, they can also communicate normally with other agents using natural language, indicating significant human-computer interaction potential. Because of the slow speed of manual text input, potentially causing delays and affecting other operations, voice-to-text conversion programs should be utilized, allowing human operators to communicate directly with agents via voice, enabling human-machine collaboration.

VI. Conclusion

This study, based on the B/S architecture Army Tactical Adversarial Intelligent Simulation Platform, completed the design of a multi-agent system Multi-Panzer for tank detachment combat simulation. Multi-Panzer, using the LLM Qwen-Long as the cognitive core, can be directly applied to tank detachment combat simulations in various terrains without requiring specialized training. Its adversarial intelligence capability surpasses that of typical traditional agents and its LLM core can be replaced as needed. When agents exhibit unreasonable decisions, their decision-making rationale can be understood by asking questions directly combined with the context, allowing for targeted adjustments and improvements, thus possessing a certain degree of explainability. Although the Multi-Panzer proposed in this study demonstrated excellent combat intelligence and collaboration capabilities in the tank unit combat simulation, it still has the following limitations: 1. Multi-Panzer relies on the reasoning ability of LLM. The parameter quantity of the underlying LLM will to some extent affect the performance of Multi-Panzer, which is an inevitable limitation for all intelligent agents that use LLM as the cognitive core; 2. The current Multi-Panzer architecture is based on a turn-based triggering mechanism and focuses on decision experiments, which is not suitable for simulating the rapid response requirements in real battlefields. Future work based on this study can further explore the impact of various subjective and objective factors in tank detachment combat on battle outcomes and can also enhance its adversarial intelligence capabilities. This study provides a novel framework for LLM-based multi-agent systems in tank detachment combat simulation, with its architectural design offering guidance for developing LLM-based multi-agent systems in combat simulation.

References

[1] Wang, L., Huang, F.: Multi-agent game, learning and control. Acta Automatica Sinica 49(3), 580‑613 (2023)

[2] Gu, X., Luo, J., Zhou, Y., et al.: A survey on intelligent agent technology for decision-making large models in intelligent gaming. Journal of System Simulation (2024)

[3] Sun, Y., Peng, Y., Li, B., et al.: Intelligent game overview: enlightenment of game AI on combat deduction. Chinese Journal of Intelligent Science and Technology (2), 157‑173 (2022)

[4] Wurman, P.R., et al.: Outracing champion Gran Turismo drivers with deep reinforcement learning. Nature (7896), 223‑228 (2022)

[5] Schrittwieser, J., et al.: Mastering Atari, Go, chess and shogi by planning with a learned model. Nature (7839), 604‑609 (2020)

[6] Silver, D., Hubert, T., Schrittwieser, J., et al.: A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science (6419), 1140‑1144 (2018)

[7] Cao, J.: Toward trustworthy AI: governance challenges and countermeasures for ChatGPT-like generative artificial intelligence. Journal of Shanghai University of Political Science and Law (The Rule of Law Forum) (4), 28‑42 (2023)

[8] Sun, C., Hua, C.: Construction of tank unit combat simulation model based on multi-agent. Journal of System Simulation 16(12), 2757‑2760 (2004)

[9] Wang, Y., Li, Q., Dai, Z., et al.: Research status and trends of large language models. Chinese Journal of Engineering 46(8), 1411‑1425 (2024)

[10] Hua, W., Fan, L., Li, L., et al.: War and Peace (WarAgent): Large Language Model-based Multi-Agent Simulation of World Wars. arXiv:2311.17227 (2023)

[11] Li, L., Hua, W., Fan, L., et al.: BattleAgent: Multi-modal Dynamic Emulation on Historical Battles to Complement Historical Analysis. arXiv:2404.15532 (2024)

[12] Waytowich, N., Giammarco, J., Guru, A., et al.: COA-GPT: Generative Pre-Trained Transformers for Accelerated Course of Action Development in Military Operations. arXiv:2402.01786v2 (2024)

[13] Amatriain, X.: Prompt Design and Engineering: Introduction and Advanced Methods (2024)

[14] Zhu, J., Chen, J.: Introduction to multi-agent system technology. Mechanical and Electrical Equipment 21(3), 25‑28 (2004)

[15] Institute of Automation, Chinese Academy of Sciences.: IA-CAS opens access to "MiaoSuan · ZhiSheng" tactical wargaming RTS human-AI confrontation platform. High-Technology & Industrialization (11), 72 (2020)

[16] Yao, C., Bi, S., Ma, R., et al.: Dynamic alliance formation method for force coordination in wargaming agents. Journal of System Simulation (2025)

[17] Zhou, Z., Zhao, H.: Design and implementation of B/S architecture-based wargaming system. Journal of Equipment Academy 27(2), 68‑72 (2016)

[18] Aliyun.: Qwen-Long_Large model service platform Bailian (Model Studio). https://help.aliyun.com/zh/model-studio/developer-reference/qwen-long/, last accessed 2024/08/11

[19] Sharma, A., Chaturvedi, A., Tripathi, A.K.: From Problem Descriptions to User Stories: Utilizing Large Language Models through Prompt Chaining. In: 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1‑2. IEEE, Kamand (2024)

[20] Ding, L., Chen, Y., Xiao, T., et al.: Preliminary exploration of generative AI application modes for new power systems based on large language models. Automation of Electric Power Systems (2024)

[21] Aliyun.: How to use Qwen-Long API. https://help.aliyun.com/zh/model-studio/developer-reference/qwen-long-api, last accessed 2024/08/11

[22] Yin, Q., Zhao, M., Ni, W., et al.: Intelligent Decision-Making Technologies and Challenges in Wargaming. Acta Automatica Sinica 49(5), 913‑928 (2023)

[23] Xie, Z., Li, H., Song, Y., et al.: From AIGC to AIGA: decision-making large models as a new intelligent track. Science Focus 19(2), 14‑33 (2024)

[24] Huang, S., Mamidanna, S., Jangam, S., et al.: Can Large Language Models Explain Themselves? A Study of LLM-Generated Self-Explanations (2023)

[25] Singh, C., Inala, J.P., Galley, M., et al.: Rethinking Interpretability in the Era of Large Language Models (2024)

[26] Alibaba Cloud.: Qwen-Long. https://bailian.console.aliyun.com/, last accessed 2024/08/11

[27] Chang, Y., Wang, X., Wang, J., et al.: A Survey on Evaluation of Large Language Models (2023)

[28] Jiang, H., Zhang, X., Cao, X., et al.: PersonaLLM: Investigating the Ability of GPT-3.5 to Express Personality Traits and Gender Differences (2023)