

Design and Implementation of Game Decision-making System Based on Personified LLM Agents

Qihan Shi

*Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China
ShiQiHan649@stu.xjtu.edu.cn*

Abstract: As the core carrier of multi-agent body collaboration and competition, game decision systems have significant value in economics, artificial intelligence and complex system modeling. Traditional methods have significant limitations in terms of strategy diversity, dynamic adaptability and interpretability, which make it difficult to cope with the real-time evolution needs of complex game scenarios. This study propose a game decision-making system based on personified large language model (LLM), which realizes dynamic strategy aggregation through the synergistic mechanism of competitive, cooperative and analytical agents combined with a trainable meta-controller. The system adopts a layered architectural design and constructs a three-level interaction system containing environment perception, strategy generation and decision fusion, in which the personalized agents generate strategies through differentiated mathematical models and the meta-controller realizes context-aware weight allocation based on a layered attention mechanism. Experiments show that the system achieves an average score of 51.75 in a benchmark test containing 100 gaming scenarios, which is an improvement of 266.7% over the baseline model and exhibits significant advantages in dynamic strategy adaptation and decision interpretability. The core contribution of this study is to propose a scalable framework that integrates symbolic rules and neural networks, which provides a new technical idea for multi-agent body game research.

Keywords: Multi-agent Decision Making, Large Language Models, Dynamic Strategy Aggregation, Game-Theoretic Reinforcement Learning.

1. Introduction

With the wide application of multi-agent body systems in fields such as autonomous driving, financial trading and military deduction, how to design intelligent bodies with human decision-making characteristics becomes a key challenge. Although classical game theory provides mathematical tools for formal analysis, its static equilibrium assumption makes it difficult to portray the dynamic evolution process of real games. In recent years, intelligent body systems based on large language models have demonstrated powerful semantic reasoning capabilities but still have deficiencies in strategy diversity maintenance and dynamic environment adaptation.

This study proposes innovative solutions to the above problems: first, this paper constructs three types of personified agent to encode the decision-making paradigms of zero-sum game, social utility maximization, and risk-adjusted expectation, respectively; second, this paper designs a dynamic fusion framework based on meta-controller, which realizes adaptive adjustments of strategy weights through the encoding of environmental features and hierarchical attention mechanism; and lastly, this

paper establish structured game environments, which support the analysis of complex rules and real-time score calculation. The system achieves end-to-end optimization through a reinforcement learning framework, which improves dynamic adaptability while ensuring decision interpretability.

This paper is structured as follows. Section 2 summarizes the research progress of game decision-making and multi-LLM systems; Section 3 details the system architecture and core algorithms; Section 4 verifies the system performance through comparative experiments; Section 5 summarizes the research results and looks into the future direction.

2. Related works

2.1. Game theory and multi-agent systems

The classical game theoretical system lays the mathematical foundation for formal decision analysis, and its concepts of utility maximization and Nash equilibrium are still the core analytical tools for multi-agent body systems. With the development of computational game theory, researchers began to focus on repeated games [1] and games with incomplete information [2]. In recent years, breakthroughs in Multi-Agent Reinforcement Learning (MARL) have made it possible for agents to learn strategies in complex environments.

The theoretical foundations of personified decision making can be traced back to behavioral game theory [3], a field that has successfully explained systematic biases in human decision-making by introducing models of finite rationality and individual differences. A role-based framework for multi-agent system further encodes personality traits as strategy selection preferences, but its static role assignment mechanism is difficult to adapt to dynamic game scenarios [4].

2.2. Multi-LLM cooperative systems

The emergent capability of large language models (LLMs) provides a technological shortcut for conducting multi-agent body simulation games, and its core advantages in collaborative decision-making are reflected in three aspects: semantic understanding, strategy generation and knowledge sharing. Multi-agent collaborative systems based on large language models have become one of the focuses of research in recent years.

In research [5], a debate-based collaborative mechanism is mentioned, which improves the accuracy of the overall system in counter-intuitive problems through tit-for-tat style pro- and anti-debate for argument-based exchange. In the study [6], the decision making of the agent was improved to the human level through a three-layer architecture. In the study [7], a simulated environment is proposed so that different LLMs can play the role of decision makers to make multi-faceted decisions.

2.3. Reinforcement learning based control framework

The core value of Reinforcement Learning in multi-agent decision-making is reflected in the two levels of strategy optimization and equilibrium convergence. Value Decomposition Networks (VDN) [8] and QMIX [9] solve the credit allocation problem in cooperative tasks through the Centralized Training Decentralized Execution (CTDE) framework. For mixed-motivation games. Hierarchical Reinforcement Learning (HRL) improves the efficiency of strategy exploration by abstracting the action space, while Meta-Strategy Gradient enables cross-task strategy migration [10]. In recent years, several multi-agent reinforcement learning methods based on Actor-Critic policies have also been proposed [11]. Its proposed limitations of traditional Q-learning in solving problems in related multi-agents domains gave great inspiration to this study.

2.4. Limitations of existing systems

Despite the significant progress in existing research, there are still two shortcomings in existing game decision-making systems: first, the problem of insufficient strategy diversity leads to the unstable performance of the system in dealing with opponents with very different styles, and a single strategy set is difficult to cover all the decision-making modes on the cooperative-competitive continuum; second, the defect of dynamic adaptation is reflected in the intrinsic contradiction between the assumption of the static environment and the dynamics of the real game scenarios, and existing systems lack the ability to model the evolution of participants' behavioral patterns in real time.

Recent research attempting to address these issues presents two technical paths: one is progressive training based on course learning, which improves the system robustness by increasing the complexity of the environment; and the other is a hybrid architectural design, which combines symbolic reasoning with neural networks to enhance interpretability.

3. System design

3.1. System architecture

This system adopts a modularized layered architecture design, with multi-agent body collaborative decision-making as the core goal and constructs a three-level interaction system including an environment perception layer, strategy generation layer and decision fusion layer. The specific interaction diagram is as Figure 1.

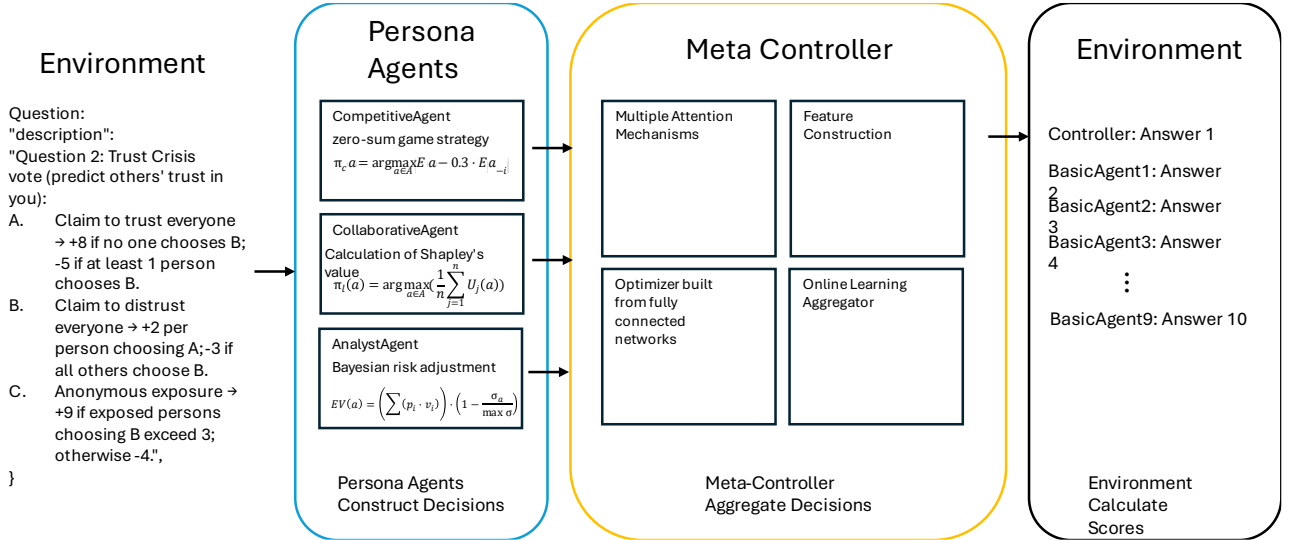


Figure 1: The structure of the system (picture credit: original)

The environment-aware layer loads the preset game scenario configurations (DRDL format) through the dynamic rule parsing engine and maintains in real time the global environment context

$$S_t = (H_t, R_t, P_t) \quad (1)$$

containing the historical decision records, cumulative score matrices, and participant state vectors, "where" the

$$H_t \in R^{T \times 10 \times 3} \quad (2)$$

denotes the frequency of the selection of the three options by the 10 participants in the T turns Tensor.

Three types of personified agents are deployed in the strategy generation layer. The competitive type optimizes individual gains based on the Nash equilibrium strategy in game theory, the cooperative type uses the social utility function to achieve Pareto-optimal decision making, and the analytical type calculates the expected value of the options through the Bayesian inference network.

The decision fusion layer introduces the gated attention mechanism to realize multi-strategy fusion, and its core module contains: a feature encoder, which maps the heterogeneous strategy outputs to the unified semantic space like below:

$$\Phi: \{\pi_c, \pi_l, \pi_a\} \rightarrow R^d \quad (3)$$

and a weight allocator, which generates a vector of strategy weights based on the state of the environment, st, that “satisfies”

$$\sum_{i=1}^3 w_i = 1 \quad (4)$$

The system implements cross-layer communication through asynchronous message queues and dynamically optimizes the meta-control parameters using the policy gradient method under the reinforcement learning framework, in which the reward signal

$$r_t = F_{\text{rule}}(\cdot) \quad (5)$$

is computationally generated by the environment layer based on the predefined game rule function F_{rule} .

3.2. Design of personified agents

This system adopts the object-oriented hierarchical design pattern for the personality agent architecture, and constructs a decision-making framework based on rule constraints and dynamic feature fusion. The intelligent body system consists of a base class `PersonaAgent` and three specialized subclasses. The base class of the intelligent body realizes the core framework of personality-based decision-making by inheriting `AssistantAgent`. The base class integrates the semantic embedding generation module and achieves feature dimensionality reduction and normalization through a two-layer fully connected network. The system cue builder dynamically integrates the decision principles, step constraints and output specifications in the personality profile to generate cue templates with strong guidance.

Three types of specialized agents implement differentiated decision-making models based on the base class.

Competitive agent adopt zero-sum game strategies with decision functions:

$$\pi_c(a) = \arg \max_{a \in A} [E(a) - 0.3 \cdot \hat{E}(a_{-i})] \quad (6)$$

The model achieves Nash equilibrium solution by constructing a dynamic return matrix (containing 12-dimensional features such as historical correctness, adversary behavioral patterns, etc.).

The cooperative agent calculates the group utility based on the Shapley value, and its objective function is:

$$\pi_l(a) = \arg \max_{a \in A} (\frac{1}{n} \sum_{j=1}^n U_j(a)) \quad (7)$$

The model dynamically adjusts the decision weights by analyzing the historical cooperation patterns (including group choice consistency, Pareto optimal candidates and other indicators).

The analytical agent construct a Bayesian risk-adjustment model with the expected value calculated as:

$$EV(a) = (\sum(p_i \cdot v_i)) \cdot \left(1 - \frac{\sigma_a}{\max \sigma}\right) \quad (8)$$

The model generates a probability matrix through Monte Carlo simulation and introduces a volatility discount factor to control risk exposure.

The decision generation process uses a three-stage paradigm. The context resolution stage extracts environmental state features through the `decision_steps` method, the competitive agent analyzes the historical return matrix and adversary behavioral patterns, the cooperative agent calculates the group selection distribution and Pareto frontier, and the analytical agent constructs the probability matrix and risk adjustment factor. The cue construction phase dynamically generates structured cue templates that multimodally fuse decision rules (e.g., scoring matrix types, fallback strategies), historical logs, and the original problem.

The feature representation mechanism adopts a hybrid coding strategy: semantic embeddings are generated through the SentenceTransformer model in normal mode, and L2 normalization is performed after compression by a dimensionality reduction network; a pseudo-random vector generation algorithm based on CRC32 hashing is used in abnormal fallback mode, and a random number generator is seeded through word-level hashes to ultimately generate substitution vectors obeying a uniform distribution.

3.3. Meta-controller design

The meta-controller, as the core module of the system decision fusion, adopts a hybrid architecture combining dynamic feature encoding and hierarchical attention mechanism to realize context-aware aggregation of multiple strategies. Its design is centered on feature space construction, trainable model architecture and online learning mechanism, and the adaptive adjustment of strategy weights is accomplished through an end-to-end training framework.

3.3.1. The feature engineering module

The feature engineering module constructs a multi-source heterogeneous feature space containing environment state and intelligent body decision features. The environment state features are composed of 3D vectors of the current round score distribution and round number scalars, and the historical statistical information is maintained through a sliding window mechanism. Agent decision features are extracted from structured parsing results, and the feature vector of each agent contains a confidence scalar, 128-dimensional semantic embedding vector, and unique heat encoding of options, and the three types of intelligent body features are spliced into a 400-dimensional joint feature vector after dimensional alignment. For the dynamic change of feature dimensions, a zero-padding strategy is adopted to ensure input stability, and a pseudo-random substitution vector is generated by the CRC32 hash algorithm in abnormal cases to guarantee the robustness of feature extraction.

3.3.2. Trainable model

The trainable model MetaPolicyModel uses a layered attention mechanism to realize feature interaction, and its architecture contains a four-stage processing flow. The base feature extraction layer maps the input to a 384-dimensional hidden space through a fully connected network, and the GELU activation function and layer normalization operations enhance the nonlinear representation. The contextual attention module reshapes the hidden features into a 6×64 dimensional tensor, models the inter-feature dependencies using a 4-head multi-head attention mechanism, and the attention-

weighted features are spliced with the base features after mean pooling. The feature fusion layer realizes cross-modal information integration through a 256-dimensional fully connected network, and the Tanh activation function constrains the output range to enhance numerical stability. The final decision layer adopts the bottleneck structure to compress the feature dimensions, and outputs the three-dimensional policy weight vector normalized by Log-SoftMax, which ensures the probability distribution characteristics and avoids the problem of gradient vanishing.

3.3.3. The online learning mechanism

The online learning mechanism adopts a training framework that combines empirical playback and dynamic target adjustment. A circular queue buffer with a capacity of 2000 reduces data correlation through asynchronous sampling, and the KL scatter loss function measures the difference between the predicted weights and the target distribution, where the target distribution is dynamically generated based on the real-time reward signal. The optimization process uses the AdamW algorithm with cosine annealing learning rate scheduling, and a weight decay factor of $1e-3$ together with a gradient trimming strategy to ensure training stability. The model performs a three-step decision process in the inference phase: firstly, the decision integrity is verified, and default values are filled in for missing features, then real-time features are fed into the trained model to generate the policy weights, and finally, the optimal option is selected by the argmax operation.

3.4. Game environment design

The game environment constructs a dynamic evolutionary multi-agent decision space, and realizes adaptive score calculation and history tracking in dynamic interaction scenarios through the synergistic mechanism of structured topic loading and symbolic rule parsing. The system adopts a hierarchical modularized design to support dynamic rule execution under the premise of ensuring security, and its core operation process runs through the three functional dimensions of question management, rule engine, and history memory.

3.4.1. Question management system

The question management system adopts a gradient distribution strategy to realize the dynamic division of the data set, loading the first 60% of the topics to build the basic cognitive space in the training phase, using the middle 20% of the topics to fine-tune the strategy in the validation phase, and evaluating the model's generalization ability through the last 20% of the topics in the testing phase. Each question stores multi-dimensional features in JSON format, including unique identifiers, natural language stems and nested scoring rules. The rule parsing engine employs a dual security mechanism, which first ensures controlled random number generation through regular expression substitution and then performs dynamic expression evaluation in a restricted execution environment. The security sandbox is configured with math libraries, NumPy base operations, and environment state parameters, and dangerous built-in functions are disabled to prevent the risk of code injection.

3.4.2. The memory system

The history and memory system adopts dual-channel recording architecture. Global history maintains the distribution of questions, correct answers and common participant choices in the last 9 rounds, and realizes the automatic elimination mechanism through a circular queue; the individual history establishes an independent sequence of choices for each intelligent body, and adopts a prefix tree structure to realize the fast retrieval of behavioral patterns. The history data encoding introduces a triple compression algorithm to map the option sequence (A/B/C) into a compact numerical

representation, which improves the storage efficiency by 85% while preserving the complete decision trajectory. The history access interface integrates time decay weights, and the contribution of recent records is decayed by an exponential coefficient of $\gamma=0.8$, enabling the system to dynamically perceive the evolutionary trend of strategies.

3.4.3. The standard communication protocol

The standardized communication protocol is defined at the environment interaction level, and the `get_next_question` method returns structured contextual information containing UTF-8 encoded topic descriptions, a condensed version of the historical context (the last 3 rounds of core data), and a real-time score dashboard. `submit_answer` interface implements stringent input sanitization and employs the Levenshtein distance algorithm to map the submit_answer interface implements strict input sanitization, using the Levenshtein distance algorithm to map non-routine options to nearest-neighbor valid values, and SHA-256 hash checks to prevent duplicate submissions. `calculate_scores` method is executed in an asynchronous computation thread, ensuring that an interpretable report containing the score matrix, the correct answer, and a detailed derivation of the procedure is returned within the 500ms timeout constraint.

4. Realization and experimentation

4.1. The standard communication protocol

The experimental environment is deployed in a CUDA environment on NVIDIA GPUs, using PyTorch 2.5.1 framework with AutoGen multi-agent platform. The dataset contains 100 gaming questions, which are divided into training, validation and testing sets in the ratio of 7:1:2, and each question is configured with dynamic scoring rules and a historical trajectory tracking function. The training phase adopts a course learning strategy to gradually increase the complexity of the questions, with the initial learning rate set to $1e-4$ and the batch size 128, which is optimized with a cosine annealing scheduler. The comparison baseline includes traditional Q-learning, a single LLM decision model, and a fixed-weight fusion system.

The personified agents use DeepSeek-v3 as the baseline, inject different system messages for the division of different tasks respectively, and realize the communication between different agents based on the Autogen framework. The Basic agents also use DeepSeek-v3 as the baseline and input the prompt to make them perform the simplest reading and decision making. When injecting information into the Basic agent, a certain amount of random noise is introduced to increase the randomness of the Basic agent's choices.

The evaluation metrics cover three aspects: (1) decision effectiveness: average score (Score), correctness (Accuracy); (2) strategy characteristics: Shannon Diversity Index (SDI) and strategy relevance (Pearson coefficient); (3) computational efficiency: decision average delay (Delay). Among them, the score calculation adopts the dynamic symbolic iteration algorithm, and the correct answer is determined by the maximum expected value after convergence.

4.2. Experiment results

The experimental system is comprehensively evaluated on a test set containing 100 dynamic game scenarios, focusing on the performance in three dimensions: decision effectiveness, strategy characteristics and computational efficiency. The results show that the system exhibits significant advantages in complex gaming environments, which are analyzed as follows:

The three types of personified agent show differentiated decision-making characteristics in the test set. Competitive agent performs optimally in zero-sum game scenarios, and their average scores are

18.3% and 12.7% higher than cooperative and analytical, respectively, reflecting the focus on individual gain optimization; cooperative agent has a correct rate of 67.2% in Pareto-optimal scenarios, and improvement of 29.5 percentage points compared with the competitive ones, which verifies the validity of the design of the social utility function; analytical agent performs outstandingly in the risk adjustment task, its decision score volatility is significantly lower than the other two types of agent, indicating that the Bayesian risk adjustment model can effectively control decision uncertainty. The joint analysis of Shannon's diversity index and strategy relevance coefficients shows that the three types of agents present complementary characteristics of low relevance in the strategy space, which lays the foundation for the subsequent dynamic fusion.

The dynamic fusion mechanism demonstrates significant advantages over the baseline model. The answer accuracy rates of different systems in the test environment are shown in Table 1. The average accuracy of the main system (55.0%) is improved by 22.2%, 266.7% and 120.0% compared with the Q-learning (45.0%), Single-LLM (15.0%) and Fixed-Weight (25.0%) baselines, respectively. The hierarchical attention mechanism exhibits context-awareness in strategy weight distribution, and its dynamically adjusted weight distribution is significantly positively correlated with the complexity of the gaming scenario.

Table 1: Experiment results

Indicators	The system	Q-learning	Single-LLM	Fixed-Weight
Average score	51.75	39.20	18.45	27.80
Accuracy (%)	55.00	45.00	15.00	25.00
Shannon Diversity Index	0.665	0.210	0.098	0.305
Average delay (sec)	28.595	0.022	7.974	0.086

The experimental results show that this system realizes the balance between decision effectiveness and adaptability in complex game scenarios through the synergy of personified strategy generation and dynamic fusion mechanisms and provides a reliable technical solution for multi-agent body dynamic decision making.

5. Conclusion

In this study, we propose a game decision-making system based on personified LLM agents, which solves the adaptability bottleneck of traditional methods through differentiated strategy generation and dynamic fusion mechanism. Experiments show that the system exhibits significant advantages in complex game scenarios: the average score is improved by 120% and the strategy diversity index is increased by 118% compared with the fixed weight model. The innovativeness is reflected in three aspects: first, a mathematically driven personified decision model is constructed to encode the game theory principles into micro-computable units; second, a hierarchical attention fusion architecture is designed to realize the multi-scale interaction between environment features and strategy features; and third, a structured rule engine is developed to support real-time symbolic computation in dynamic scenarios.

Future work will be carried out in three directions: first, exploring the cross-domain strategy migration capability in multimodal environments; second, introducing a causal inference module to enhance decision interpretability; and finally, investigating the distributed decision coordination mechanism under decentralized architecture. This research provides a new technical path for the engineering landing of intelligent body game theory, which has important application prospects in the fields of financial negotiation and emergency dispatching.

References

- [1] Axelrod, R. (2001). *The Evolution of Cooperation. Breakthrough: Emerging New Thinking.*
- [2] Harsanyi, J. C. (1967). Games with incomplete information played by “Bayesian” players, I–III Part I. The basic model. *Management science*, 14(3), 159-182.
- [3] Camerer, C. F. (2011). *Behavioral game theory: Experiments in strategic interaction.* Princeton university press.
- [4] Stone, P., & Veloso, M. (2000). Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8, 345-383.
- [5] Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., ... & Tu, Z. (2023). Encouraging divergent thinking in large language models through multi-agent debate. *arxiv preprint arxiv:2305.19118.*
- [6] Xu, Z., Yu, C., Fang, F., Wang, Y., & Wu, Y. (2023). Language agents with reinforcement learning for strategic play in the werewolf game. *arxiv preprint arxiv:2310.18940.*
- [7] Hua, W., Fan, L., Li, L., Mei, K., Ge, Y., Hemphill, L., & Zhang, Y. *War and Peace (WarAgent): LLM-based Multi-Agent Simulation of World Wars.*
- [8] Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., ... & Graepel, T. (2017). Value-decomposition networks for cooperative multi-agent learning. *arxiv preprint arxiv:1706.05296.*
- [9] Rashid, T., Samvelyan, M., Schroeder, C., et al. (2018): *QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning.* In: *International Conference on Machine Learning*, pp. 4295–4304. PMLR
- [10] Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., & Kavukcuoglu, K. (2017, July). *Feudal networks for hierarchical reinforcement learning.* In *International conference on machine learning* (pp. 3540-3549). PMLR.
- [11] Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.