FC-AIT2026: Team Description Paper

Masaki Nishio, Ryota Sugamoto, Atsushi Ueda, Naoto Hashimoto, and Nobuhiro Ito[0000−0002−8062−9989]

Department of Information Science, Aichi Institute of Technology, Japan
fcait@maslab.aitech.ac.jp

Abstract. Team FC-AIT focuses on game analysis and team modeling for Soccer Simulation 2D (SS2D). This paper describes the practical application of the Expected Possession Value (EPV) model, a game analysis approach, to pass decision-making. Prior work has demonstrated that EPV-based pass selection improves ball possession rate and pass success rate compared to conventional heuristic methods [1]. However, inference with the EPV model is computationally expensive, making end-to-end EPV-based action selection difficult under the 100 ms per-cycle constraint in SS2D. To address this, we evaluated two acceleration techniques: (i) deploying the model via ONNX Runtime in C++, and (ii) asynchronous inference [2]. The results show that the introduction of asynchronous processing enabled decision-making within 100 ms. However, the ball possession rate did not reach the level achieved by the conventional method. The transition from full-state information to partial observation likely introduced noise into the model. This noise led to the underestimation of Pass EPV, thereby reducing the overall pass selection frequency. Future work will quantitatively verify the effects of observation noise and establish a decision-making framework in a competitive environment where all actions are evaluated equally. Moreover, integrating this approach into Team FC-AIT reduced the average goals conceded against Helios2025 from 6.7 to 4.4, demonstrating improved defensive skill compared to the previous version.

1 Introduction

1.1 Team Overview

Team FC-AIT has participated in the RoboCup JapanOpen twice. RoboCup2026 will be our first participation in the World Championship. The agent development framework is based on "HELIOS Base", which was released as open source by Team HELIOS [3]. Building on this, we extend the framework by integrating an EPV-based pass evaluation model with a real-time inference pipeline (ONNX Runtime and asynchronous processing) to handle partial observability.

We research cooperative strategies in multi-agent systems using Soccer Simulation 2D (hereafter, SS2D), focusing on behavioral analysis of players and groups. Current efforts center on game analysis and team modeling.

As part of the game analysis research, we introduce a decision-making method based on Expected Possession Value (hereafter, EPV). EPV is a metric that quantifies the expected value of future scoring based on match conditions such as player and ball positions. Originally developed for real-world sports analytics, EPV has been shown to be effective in SS2D by Amano et al. and Ban et al. [4, 1]. In particular, EPV-based methods substantially improve both ball possession and pass success rate compared to conventional heuristic approaches. Based on these findings, we adopt an EPV-based pass evaluation model; furthermore, we redesign our inference-and-decision pipeline to make this learning-based approach practical under the strict 100 ms constraint of SS2D.

1.2 Comparison of Evaluation Methods in Action Sequence Search

Action Sequence Search is an important technique for action selection in SS2D [5]. Our approach enables agents to select actions that contribute to future scoring. To this end, the evaluation function used in Action Sequence Search is improved using EPV. In contrast, evaluation function designs by other teams can be broadly classified into three categories:

1. Experience and Analysis
   - **ITAndroids2021**: A parameter tuning method based on Particle Swarm Optimization [6]
   - **MT2022**: Improvements informed by insights from log analysis [7]
2. Probabilistic and Approximate Methods
   - **Alice2021**: Construction incorporating uncertainty via Monte Carlo Tree Search [8]
3. Machine Learning
   - **EMPEROR2023**: Improvement using perceptrons and entropy [9]
   - **HELIOS2023**: A ranking method based on learning-to-rank with gradient boosting trees [10]

Improving the evaluation function via EPV can be viewed as an extension of the machine learning–based approach, since the EPV model itself is constructed through machine learning. However, the EPV model incurs a high computational cost. We therefore address this limitation.

2 Practical Application of the Expected Possession Value Model

2.1 Details and Challenges of Expected Possession Value

To gain an advantage in SS2D matches, each player must perceive the dynamically changing match situation and select appropriate actions (hereafter, decision-making) accordingly. Expected Possession Value (hereafter, EPV) serves as a metric for quantitatively evaluating such decision-making.

EPV defines the period from the start to the end of ball possession as a possession. It estimates, in real time, the expected value of the score (e.g., 0, 2, or 3 points, depending on the sport) obtainable at the end of the possession [11].

Following Fernandez et al. [12], a possession is defined as the continuous sequence of play from a given time point until the next goal is scored or a fixed duration elapses. This interval may include transitions of ball control to the opposing team. Player actions are classified into three categories: pass, dribble, and shoot. The expected value of each action's scoring outcome is weighted by its selection probability, and the resulting values, normalized to the range $[-1, 1]$, are summed to constitute the overall expected value of the possession.

In a prior study, Amano et al. extracted features from SS2D log data and constructed an EPV model [4]. Furthermore, by visually analyzing pass course tendencies based on the model's output, they demonstrated the effectiveness of EPV for SS2D match analysis. Ban et al. subsequently focused on the "pass" component of EPV and attempted to apply it to decision-making [1]. As a result, the ball possession rate and pass success rate were substantially improved compared to conventional heuristic methods, demonstrating EPV's effectiveness in reducing the risk of conceding goals. Nevertheless, a tactical limitation became apparent: evaluating only Pass EPV does not improve scoring ability. Additionally, a practical implementation challenge remained—the high computational cost made it difficult to complete processing within a single cycle (100 ms).

This section therefore addresses the computational cost problem to leverage the defensive advantages of improved ball possession rate and pass success rate in the competition environment. Specifically, two techniques are implemented to achieve real-time processing: porting the model to C++ and making the inference asynchronous. Subsequently, an evaluation is conducted to determine whether the defensive advantages of Pass EPV—namely, ball possession rate and pass success rate—are maintained even with the accelerated model.

2.2 Decision-Making Using Pass EPV

The Pass EPV model used in this study takes the positions and velocities of the ball and all agents as input, and outputs the Pass EPV for each grid point on a grid-discretized field, representing the expected value when the ball is moved to that point. The expected value is normalized to $[-1, 1]$.

The model training process is illustrated in Fig. 1.

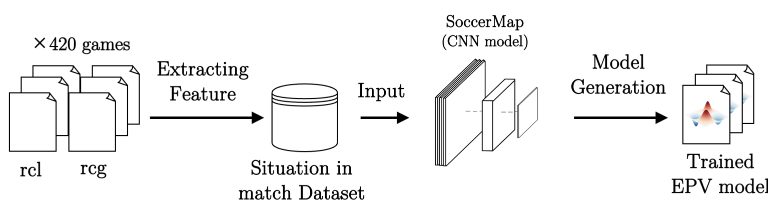

Fig. 1. Trained EPV model (example)

The EPV model is trained using log data (rcl and rcg files) obtained from 420 past matches. We construct a dataset linking match situations to action outcomes from these logs and train the SoccerMap deep learning architecture accordingly [13]. This process yields a model that takes the positions and velocities of the ball and all agents in any given situation as input and outputs the expected value at each point on the field.

An overview of decision-making using the trained model is presented in Fig. 2.

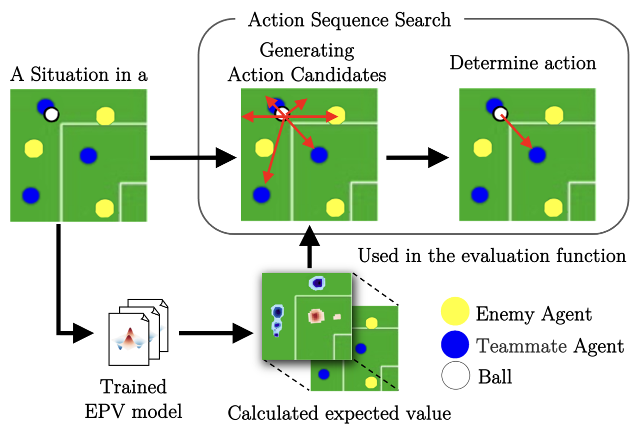

Fig. 2. Overview of pass decision-making using EPV

The agent feeds the current match situation (e.g., positions of agents and the ball) into the trained model to compute the Pass EPV distribution (heat map) across the field. Simultaneously, Action Sequence Search—the decision-making algorithm—generates multiple action candidates. For each pass candidate among them, the Pass EPV at the target location is used as the evaluation function, thereby selecting the pass destination with the highest expected value. In the study by Ban et al., agents using this Pass EPV model as the evaluation function were observed to preferentially select passes that avoid ball loss. Compared to the base team, ball possession rate improved by 25.9 percentage points and pass success rate by 18.35 percentage points. The present study focuses on this defensive characteristic.

2.3 Challenges and Methods for Real-Time Inference

Challenges and Objectives

A primary limitation identified by Ban et al. is the high computational cost of real-time EPV inference. In SS2D, one cycle lasts 100 ms, within which all communication, reasoning, and action determination must be completed.

Accordingly, this study aims to achieve EPV-based decision-making within a single cycle by redesigning the reasoning and action determination workflow. The target threshold is set at 72.8 ms, derived by subtracting the maximum communication processing time (27.1 ms), measured in a preliminary experiment (30 matches) using HELIOS_BASE V18, from the per-cycle limit of 100 ms.

Methods for Reducing Inference Time

To enable decision-making under real-time constraints, two methods were sequentially evaluated in this study. Although model quantization was considered, preliminary experiments revealed its dependency on CPU architecture. Therefore, it was excluded from this study to ensure operational stability in the competition environment.

1. **Execution speed improvement at the language level**: The trained model (PyTorch) was exported in ONNX format, and inference was executed from C++ using ONNX Runtime [2]. This C++ implementation improves compatibility with our main framework (C++) and enhances execution speed.
2. **Asynchronous inference processing**: The EPV inference process was configured to run on a separate thread, independent of the agent's decision-making process (main thread). In addition, a frequency control mechanism was incorporated to execute and update inference once every three cycles. This mechanism allowed the system to accommodate the computationally intensive inference process. Under sequential processing, inference causes delays in action determination at each cycle (100 ms). Asynchronous processing avoids this issue while reducing the load on computational resources through inference frequency throttling. It should be noted that during cycles in which inference results are not updated, the most recent result is reused, introducing a trade-off between responsiveness and the freshness of cached results.

The processing flow of the proposed method is shown in Fig. 3.

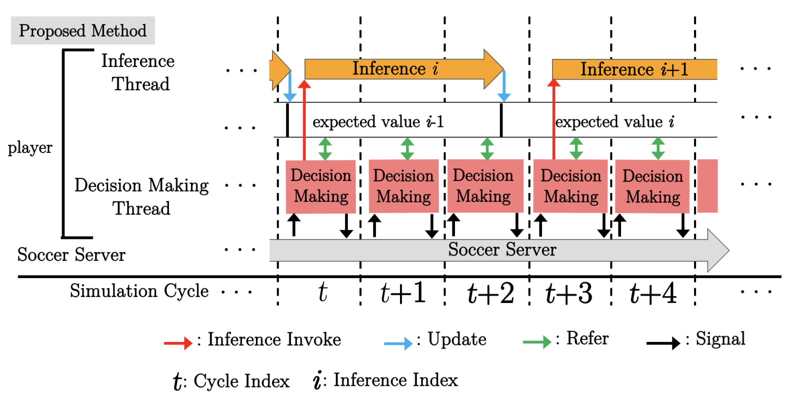

Fig. 3. Processing flow of asynchronous pass decision-making using EPV

In the conventional approach, inference and decision-making are executed on a single thread every cycle, causing decision-making to enter a waiting state until inference completes and resulting in action determination delays spanning multiple simulation cycles. To address this problem, the proposed method shown in Fig. 3 separates the inference thread from the decision-making thread, allowing both to operate in parallel. Inference is executed once every three simulation cycles, and the decision-making thread determines actions every cycle by referencing the most recent inference result without waiting for the current inference to complete.

Specifically, while the i-th inference is being executed on its thread at cycle t, the decision-making thread references the $(i-1)$-th inference result (expected value $i-1$). Upon completion of the i-th inference, the result is updated as the expected value i, and subsequent cycles use this new value.

This asynchronous architecture eliminates the need for synchronous waiting, enabling action determination to continue within the 100 ms constraint at every cycle. However, a trade-off exists in that cycles during active inference reuse the previous result, leading to reduced information freshness.

To evaluate the effectiveness of each method, the models listed in Table 1 were implemented and compared.

Table 1. Configuration and evaluation purpose of comparison models

Model	Language	Quantization	Async	Full-State
A (Prior study)	Python	–	–	Yes
B (Ported)	C++	–	–	No
C (Async)	C++	–	Yes	No

Model A in Table 1 is a reference baseline from our previous implementation and can only be executed with full-state information. Models B and C, on the other hand, use partial observations under competition rules. Accordingly, strategic comparisons should be interpreted with this difference in mind.

Model A from the study by Ban et al. employs Python for its inference component, whereas Models B and C are ported to C++. Model C further incorporates asynchronous processing.

2.4 Experimental Setup

To evaluate the effectiveness of the proposed acceleration methods, Models B and C were each matched against HELIOS_BASE V18 in 30 games. A "successful pass" was defined as a teammate receiving the ball after a kick command without interception by an opponent.

Two categories of evaluation metrics were used: processing performance and strategic performance. Processing performance was assessed by mean decision-making time, maximum decision-making time, standard deviation, and the success rate within the acceptable time (72.8 ms). Decision-making time was measured as the duration of a single action determination process on the main thread, excluding communication processing. The mean decision-making time and success rate were calculated across all cycles in 30 matches. Strategic performance was assessed by ball possession rate, pass success rate, mean goals scored, and mean goals conceded. Model A from the prior study was included only as a reference for strategic performance comparison, as its measurement environment differed.

2.5 Processing Performance Results and Discussion

The processing performance measurement results for each model are presented in Table 2.

Table 2. Comparison of decision-making time across models

Model	Mean time [ms]	Max [ms]	Std. dev. [ms]	Success rate[1] [%]
B (C++)	193	432	30.6	0.0
C (Async)	0.259	58.0	1.29	100.0

As shown in Table 2, Model B with C++ porting alone achieved a completion rate within the time constraint of 0.0%, whereas Model C with asynchronous processing achieved 100.0%. This is because C++ porting alone was insufficient to reduce the cost of the decision-making process for Model B. In contrast, Model C made the inference asynchronous, so that the main thread only referenced results from shared memory, resulting in a substantial reduction in processing time. Furthermore, the standard deviation of 1.29 ms is small, and the maximum value remains below the acceptable time threshold, indicating that our method ensures stable real-time processing in the competition environment.

Based on these results, real-time operation of the Pass EPV model via asynchronous processing is feasible in environments where a dedicated core can be allocated for inference.

2.6 Strategic Performance Results and Discussion

The strategic performance results for each model are presented in Table 3.

Table 3. Comparison of strategic metrics

Model	Possession [%]	Pass success [%]	Mean scored	Mean conceded
A (Reference)	75.61	96.24	0.65	0.94
B (Ported)	59.84	99.70	0.97	1.37
C (Async)	63.03	99.65	1.27	1.23

[1] Percentage of decision-making completed within the acceptable time limit (72.8 ms)

As shown in Table 3, both Models B and C exceeded the reference values in pass success rate and mean goals scored. However, the ball possession rate and mean goals conceded fell short of the reference values, indicating that the defensive advantages were not fully reproduced.

The primary factor behind this discrepancy is the reduced frequency of pass selection at the decision-making stage. The average number of passes in Models B and C decreased by 40% relative to the reference value, with a corresponding increase in the selection of dribbles and shots. This behavioral shift may be attributed to observation noise arising from the use of partial observation information, which increased the estimation error in the Pass EPV and led to the underestimation of the expected value of passes. Consequently, the evaluation values of non-pass actions became relatively higher in the decision-making process, reducing the frequency of pass selection. This reduced pass frequency decreased opportunities for maintaining ball possession, ultimately contributing to the increase in goals conceded. By contrast, the consistently high pass success rate indicates that the Pass EPV model's assessment of the expected value remains sound when passes are actually selected.

These findings indicate the need for a decision-making framework that eliminates bias arising from differences in evaluation methods. Such a framework would allow all actions to be compared on an equal basis in the competition environment. Quantitative verification of the effects of observation noise on pass evaluation therefore constitutes a key direction for future work.

3 Conclusion

This paper presented an overview of Team FC-AIT and its activities, and proposed acceleration methods for the practical application of the EPV model to pass decision-making in SS2D. In terms of processing performance, C++ porting alone failed to meet the target threshold of 72.8 ms. However, the introduction of asynchronous processing achieved a 100% success rate within the acceptable time limit, demonstrating the feasibility of real-time operation in practice. Regarding strategic performance, the pass success rate was maintained above 99%. However, the ball possession rate fell below the level reported in the prior study (Model A). A plausible explanation is that noise arising from the transition from full-state to partial observation information led to the underestimation of Pass EPV, thereby reducing pass selection frequency. To address this, our future work will establish a fair decision-making framework by quantitatively mitigating observation noise. Furthermore, the integration of this approach into Team FC-AIT reduced the average goals conceded against Helios2025 from 6.7 to 4.4, demonstrating an improvement in defensive skill compared to the previous version.

References

1. BAN, Masaki, et al. Analysis of the Expected Possession Value in RoboCupSoccer Simulation 2D. In: 2023 14th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI). IEEE, 2023. p. 293-298.
2. ONNX Runtime developers. ONNX Runtime. https://onnxruntime.ai/. 2021. Version: 1.16.3
3. AKIYAMA, Hidehisa; NAKASHIMA, Tomoharu. Helios base: An open source package for the robocup soccer 2d simulation. In: Robot Soccer World Cup. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 528-535.
4. AMANO, Takumi, et al. A Study on the Effectiveness of Expected Possession Value in RoboCupSoccer Simulation 2D. The Japanese Society for Artificial Intelligence, 2023, 2022.SAI-045: 01.
5. H. Akiyama, "Online Tactics Planning using Action Sequence Search, " in *JSAI Technical Report SIG-Challenge-B101-6 (5/4)*
6. Diego T. M. Fidalgo, Felipe V. Coimbra, Gustavo F. Gottschild, Jony dos S. S.Filho, Vinicius F. Almeida, and Marcos R. O. A. Maximo. "ITAndroids 2D Soccer Simulation Team Description Paper 2021"
7. Linsheng Guan, Qiuna Chen, Jingjing Wang, Haotian Xiang, Shuaishuai Meng,Cunyuan Wang, Haoyu Xu, Haicheng Fang, Shengbing Chen. MT2022: Team Description Paper
8. Fan Wang, YuTang Guo,XingQi Zheng and XinYu Liu. "Alice2021: Team Description Paper"
9. Erfan Fathi , and Soroush Mazloum. "EMPEROR Soccer Simulation 2D Team Description Paper 2023"
10. Hidehisa Akiyama,Tomoharu Nakashima,Kyo Hatakeyama, Takumi Fujikawa, and Akei Hishiki.HELIOS2023: Team Description Paper
11. CERVONE, Daniel, et al. A multiresolution stochastic process model for predicting basketball possession outcomes. Journal of the American Statistical Association, 2016, 111.514: 585-599.
12. Fernández, Javier, Luke Bornn, and Daniel Cervone. "A framework for the fine-grained evaluation of the instantaneous expected value of soccer possessions." Machine Learning 110.6 (2021): 1389-1427.
13. Javier Fern´andez and Luke Bornn. Soccermap: A deep learning architecture for visually- interpretable analysis in soccer. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 491-506. Springer, 2021.