Journal of Digital Frontier
https://doi.org/10.67541/jdf2601
Research Article Volume 1, Issue 1 (July 2026) Open Access

Modeling of the Randomness in the Process of E-sports Competitions Based on Neural Stochastic Differential Equations

Shuai Ma*

College of Physical Education, Hanyang University, Seoul, Republic of Korea

***Corresponding author:** Shuai Ma. Email: mashuai9248@163.com

Abstract

The process of e-sports competitions exhibits complex dynamic characteristics such as strong nonlinearity, multi-time-scale coupling, and heterogeneous random fluctuations. Traditional deterministic models or constant noise assumption-based stochastic models are difficult to accurately depict their evolution patterns. This paper proposes a stochastic modeling method for the e-sports competition process based on neural stochastic differential equations. By parameterizing the drift function and diffusion function using neural networks, it learns the state-dependent deterministic trends and adaptive random perturbation intensities from the data end-to-end. A controllable stochastic regulation mechanism and a multi-level time-scale modeling structure are designed to address the significant differences in randomness at different competition stages and the coupling modeling of micro-operations and macro strategies. By integrating temporal event encoding, graph neural network relationship extraction, and numerical statistical features, a multimodal state representation is constructed. Experiments are conducted on a large-scale dataset containing 12,478 professional competitions. The results show that the model in this paper reduces the mean square error (2.37×10^{-3}) and the average absolute error (1.03×10^{-2}) by 46.0% and 32.2% respectively compared to Transformer, and the actual coverage rate of the 95% confidence interval of the prediction distribution reaches 93.7%. The average error in predicting key event times is 28 seconds, verifying the effectiveness and superiority of neural stochastic differential equations for stochastic modeling of e-sports competition processes.

Keywords

Neural Stochastic Differential Equations; Modeling of E-sports Competitions; Stochastic Process; Uncertainty Quantification; Multimodal Temporal Prediction

Received: 31 May 2026; Revised: 29 Jun 2026; Accepted: 02 Jul 2026; Published: 09 Jul 2026

1. INTRODUCTION

E-sports, as a rapidly emerging digital sports activity, has achieved an unprecedented level of industrialization and professionalization in recent years [1]. The improvement of the professional league system, the maturity of club management models, and the continuous expansion of the audience base have made e-sports competitions not only entertainment activities but also a highly commercial and academically valuable competitive field [2]. Compared with traditional physical sports events, e-sports competitions have unique complexity: the progress of the game is determined by the real-time decisions, precise operations, and tactical cooperation of both players under intense confrontation, and is also influenced by various factors such as game version updates, hero selection combinations, and random refresh of map resources. This highly dynamic, interactive, and uncertain characteristic makes the modeling and prediction of the game progress a highly challenging research topic. From a practical application perspective, accurate game progress prediction can serve various scenarios, such as assisting coaches in tactical decision-making, helping viewers understand the game trend, providing real-time performance feedback to players, and even offering data support for game balance adjustments, and even providing data support for game balance adjustments [3].

However, the random nature of e-sports game progress poses fundamental difficulties for modeling [4]. On the one hand, many key events in the game have inherent randomness, such as the triggering of critical hits, the refresh location of map resources, and the uncertainty of players' on-the-spot reactions, which means that even if the strength gap between the two teams is clear, the game result is not completely determined [5]. On the other hand, the intensity of randomness exhibited in different stages of the game varies significantly, with players mainly relying on fixed operation patterns and the advantage of accumulated basic skills in the early laning phase, and random fluctuations being relatively controllable [6]. However, in the mid-game team battle stage, a slight deviation in the aiming angle of a skill or a misstep of a few milliseconds can lead to the collapse of the entire team battle, resulting in a huge economic and rhythm gap. The existence of heterogeneous random fluctuations makes traditional modeling methods based on constant noise assumptions difficult to accurately depict the real dynamics of the game progress. What is more complex is that e-sports competitions simultaneously involve microscopic-scale player operations and macroscopic-scale strategic resource scheduling, with a close coupling relationship between the two scales. Local advantages at the micro level will gradually accumulate into macro-level leading positions, while macro decisions will change the environmental conditions of the micro confrontation [7].

The existing e-sports game modeling methods show significant limitations when facing these challenges. Traditional statistical methods such as the Elo rating system only rely on pre-game information and game results for updates, unable to utilize the temporal information during the game [8]. Markov chains and hidden Markov models can describe the transfer relationships between states, but their assumption of a discrete state space limits the model's ability to express continuous numerical indicators, and the assumption of constant transfer probability contradicts the characteristics of game dynamic evolution [9]. Standard stochastic differential equation models use drift terms and diffusion terms to describe the evolution of continuous states, but their preset function forms (such as linear drift, constant diffusion) are difficult to adapt to the complex nonlinear relationships and time-varying randomness in e-sports competitions [10],[11]. Deep learning methods such as recurrent neural networks and Transformers have achieved significant progress in prediction accuracy, enabling them to automatically learn complex temporal patterns from massive game data. However, these models are deterministic and can only output point predictions without providing uncertainty quantification, which is a serious drawback in high-risk decision-making scenarios such as team fight outcome prediction. Some attempts to introduce uncertainty usually adopt model integration or approximate Bayesian methods, but these methods lack explicit modeling of the dynamic process itself of the random dynamic and how it evolves with the game progress [12],[13],[14].

To address the above issues, this paper proposes a method for modeling the randomness of e-sports competition processes based on neural stochastic differential equations. Neural stochastic differential equations integrate the theoretical framework of stochastic differential equations with the expressive power of deep learning, enabling the learning of state-dependent drift functions and diffusion functions from data, thereby achieving end-to-end modeling of complex stochastic dynamical systems. We define the e-sports competition state as a continuous-time stochastic process, where the drift network is responsible for learning the deterministic evolution laws of the competition state, and the diffusion network is responsible for characterizing the random fluctuation intensity related to the state and time.

On this basis, we designed a multi-level time-scale modeling structure, capturing the high-frequency fluctuations at the player operation level through micro-state variables, describing the long-term evolution trends of economy and resources through macro-state variables, and introducing a cross-scale fusion mechanism to achieve efficient interaction of the two layers of information. To address the significant differences in randomness during different competition stages, we proposed a controllable randomness adjustment mechanism, allowing the diffusion intensity to adaptively adjust according to the current competition stage and state. Additionally, we constructed a multimodal feature embedding system, integrating temporal event encoding, graph structure relationship modeling, and numerical statistical features, to provide rich state representations for neural stochastic differential equations. In terms of model training, we combined negative log-likelihood, KL divergence constraints, and adversarial training losses to guide the model to simultaneously optimize point prediction accuracy and distribution fitting quality.

Through extensive experiments on real professional e-sports competition datasets, the proposed method outperforms existing baseline models in multiple dimensions such as prediction accuracy, probability prediction calibration, and key event time prediction. Ablation experiments verified the effectiveness of the adaptive diffusion mechanism, multi-level time-scale modeling, and graph neural network feature embedding components. Visual analysis further demonstrated the model's dynamic estimation ability of uncertainty in different competition stages and the high consistency of generated trajectory distributions with the real distributions. The main contributions of this study can be summarized as: introducing neural stochastic differential equations into the e-sports competition process modeling field for the first time, proposing a complete end-to-end stochastic dynamic prediction framework; designing an adaptive randomness adjustment mechanism and multi-level time-scale modeling structure for e-sports scenarios, effectively capturing heterogeneous random fluctuations and cross-scale coupling in the competition; constructing a multimodal feature fusion system, fully utilizing the temporal, structural, and statistical information in the competition data; through systematic experiments, the superiority of the model in prediction accuracy and uncertainty quantification was verified. The research in this paper not only provides new methodological tools for e-sports data analysis, but also explores feasible paths for the application of neural stochastic differential equations in the modeling of complex competitive systems.

2. RELATED WORK

Research on modeling e-sports competitions and stochastic process methods has received increasing attention as the e-sports industry develops rapidly. Early analyses of e-sports competitions mainly relied on basic data statistics and post-game reviews, lacking mathematical descriptions of the dynamic evolution of the competition process [15]. With the continuous refinement of available competition data granularity, researchers began to view e-sports competitions as a complex dynamic system that evolves over time, in which team decisions, resource competition, and players' on-the-spot performance collectively constitute a highly uncertain stochastic process. From a macro perspective, the direction of an e-sports competition can be described as a trajectory in a state space, and due to the uncertainty of human operations, information asymmetry, and the randomness of tactical games, this trajectory inherently has the characteristic of being unable to be precisely repeated, which prompts scholars to introduce the theoretical framework of stochastic processes for modeling and analysis [16].

In terms of the application of traditional statistical methods and Markov and stochastic differential models, researchers initially attempted to use basic regression analysis or Bayesian methods to predict competition results, such as estimating winning probabilities through pre-game team strength ratings and historical encounter records. However, these methods cannot capture the dynamic information in the competition process and have very limited predictive capabilities. Subsequently, the Markov chain model was introduced into e-sports competition modeling because it can describe the state transitions of the system in a discrete state space. For example, researchers divided the competition into several stages such as lane positioning, team battles, and advance, and assumed that the transition probabilities between each stage only depend on the current stage, thereby constructing a Markov chain to predict the macro trend of the competition process. More detailed work employed the hidden Markov model, treating the tactical intentions or players' psychological states that cannot be directly observed as latent variables, and inferring the changes in latent states through observable event sequences [17]. The stochastic differential equation model further describes the competition state as a stochastic process in continuous time, where the drift term characterizes the deterministic growth trend of indicators such as team economy and experience, and the diffusion term captures the fluctuations caused by random factors such as the victory or defeat of team battles, or the success or failure of key skill hits. Standard stochastic differential equation models usually assume that the drift and diffusion functions have preset parametric forms, such as linear drift or constant diffusion, and infer parameters using traditional methods like Kalman filtering or maximum likelihood estimation. These methods have certain advantages in interpretability and computational efficiency, but their preset function forms often fail to adapt to the complex and nonlinear dynamic relationships in e-sports competitions [18].

Deep learning and sequence modeling methods have achieved significant progress in the field of e-sports competition process prediction. Recurrent neural networks and their variants such as Long Short-Term Memory networks and gated recurrent units, with their ability to model variable-length sequence data, are widely used to handle event stream data in competitions [19]. Researchers encode discrete events such as kills, resource acquisition, and equipment updates as vector sequences and input them into recurrent neural networks to extract temporal dependency features, thereby predicting the future state of the competition or the final result within a short period. These methods can automatically learn complex interaction patterns between events and outperform traditional statistical methods. However, recurrent neural networks are prone to gradient vanishing or explosion problems when handling extremely long sequences, and their serialized computational approach limits parallel efficiency. The introduction of the Transformer architecture largely alleviates these problems, as its self-attention mechanism can directly capture the dependencies between any two time-steps in the sequence, demonstrating superior performance in long sequence modeling tasks [20],[21]. In e-sports prediction scenarios, models based on the Transformer architecture can simultaneously focus on early lane details and key turning points in later team battles, effectively modeling global dependencies. In addition, graph neural networks have also begun to be applied to modeling the interactions among multiple entities in competitions. For example, ten heroes can be regarded as nodes in the graph, and edges can be defined based on skill interactions, vision relationships, etc. Through graph convolution operations, the collaborative patterns within the team and the competitive patterns between opponents can be extracted. Although deep learning methods have achieved significant improvements in prediction accuracy, most existing models are essentially deterministic models that can only output point predictions but cannot quantify the uncertainty in the predictions. A few attempts to introduce randomness usually adopt Monte Carlo rejection methods or deep integration as approximate Bayesian methods, but these methods lack explicit modeling of the inherent stochastic dynamics of the game process and are difficult to adapt to the dynamic changes in the intensity of randomness in different game stages [22],[23],[24].

Neural stochastic differential equations, as a recent emerging frontier direction, integrates the theoretical framework of stochastic differential equations with the expressive power of deep learning, providing a new paradigm for the modeling of complex stochastic dynamical systems. Traditional stochastic differential equations are limited by preset function forms, while neural stochastic differential equations use neural networks to parameterize the drift function and diffusion function, enabling the model to learn any complex deterministic evolution patterns and random fluctuation modes from data [25]. The core idea of this framework is to represent the state evolution of the system as the superposition of a deterministic trend driven by the neural drift field and a random perturbation driven by the neural diffusion field, where the intensity of the diffusion term can depend on the current state and time. Neural stochastic differential equations have demonstrated excellent performance in fields such as financial time series analysis, physical system simulation, and medical time series prediction, achieving both high-precision state prediction and reasonable uncertainty quantification [26],[27],[28]. In terms of training, neural stochastic differential equations typically use the adjoint sensitivity method or through numerical solvers for stochastic differential equations for backpropagation, combined with loss functions such as negative log-likelihood or variational lower bounds for parameter optimization. Existing research has also explored the introduction of control variables into the neural stochastic differential equation framework, enabling the model to respond to external interventions or strategy inputs. However, research on applying neural stochastic differential equations to modeling the progress of e-sports competitions is still in a blank state. The unique multi-time-scale characteristics of e-sports competitions (coexistence of micro-operations and macro strategies), heterogeneous random fluctuations (stable in the early line stage but drastic in the team battle stage), and rich multimodal data (event sequences, network interaction, numerical statistics) pose new challenges for the architecture design of neural stochastic differential equation models [29],[30].

The limitations of current research mainly lie in the following aspects. First, existing methods either adopt overly simplified stochastic process assumptions (such as Markov chains or linear stochastic differential equations), losing the complex nonlinear dynamic relationships in e-sports competitions; or use deterministic deep learning models, completely ignoring the inherent randomness of the game, resulting in overly confident predictions and inability to provide uncertainty estimates. Second, existing stochastic models usually assume that the intensity of random fluctuations remains constant throughout the game, while in e-sports competitions, the randomness varies significantly at different stages, such as the outcome of the early line stage mainly depending on the basic skills of the players, with relatively controllable randomness, while a deviation in a skill hit in the mid-game stage can lead to a reversal of the team battle result, with significantly enhanced randomness. Third, existing methods struggle to simultaneously model the high-frequency fluctuations at the micro-operation level and the long-term evolution trends at the macro-strategy level, and there is a close coupling relationship between the two - the accumulation of micro advantages gradually transforms into macro advantages, and macro decisions also affect the environment of micro confrontations. Fourth, the fusion processing of multimodal features is not fully sufficient. Different types of information such as event sequences, entity interaction graph structures, and continuous numerical indicators are often processed in isolation, failing to fully leverage their complementary advantages. In response to the aforementioned limitations, the innovation of this study lies in proposing a framework for modeling the randomness of e-sports competition progress based on neural stochastic differential equations. This framework utilizes the parameterized drift and diffusion functions of neural networks to automatically learn the nonlinear evolution patterns and state-dependent random fluctuation intensities of the competition state. Through multi-level time-scale modeling, it achieves the joint depiction of micro-operations and macro strategies. Graph neural networks and Transformer encoders are introduced to achieve the deep integration of multi-modal features. An adaptive randomness adjustment mechanism is designed to enable the diffusion intensity to dynamically adjust according to the competition stage and current state, thereby achieving more accurate and reliable random modeling and prediction of the e-sports competition progress.

3. MODEL METHODOLOGY

3.1 Modeling framework of neural stochastic differential equations

The process of e-sports is essentially a complex time-series system with strong nonlinearity, dynamic interaction and random disturbance. In the process of competition, players' decision-making, team cooperation, resource competition and emergencies jointly affect the evolution of competition state, which makes it difficult for the traditional deterministic dynamic model to fully describe its inherent randomness. Therefore, this paper uses neural stochastic differential equation (neural SDE) to build a stochastic dynamic model of the e-sports game process, and represents the game state as a continuous time stochastic process. By learning the state transition law and random disturbance mechanism, the probability modeling of the future evolution trajectory of the game is realized.

Let the state vector of the competition at time t be represented as:

$$X_t = [x_1(t), x_2(t), \cdots, x_d(t)]^\top \in \mathbb{R}^d \tag{1}$$

Here, d represents the state dimension, and $x_i(t)$ represents the i-th feature of the game state, such as the economic gap between teams, the experience gap, the number of remaining turrets, the kill count, the map control rate, and hero status. To describe the influence of external strategic factors on the evolution of the game, control variables are introduced.

$$U_t = [u_1(t), u_2(t), \cdots, u_m(t)]^\top \tag{2}$$

Here, m represents the dimension of the control variables, and $u_j(t)$ can represent external input factors such as team composition, tactical decisions, resource competition strategy and player operations.

Based on the above definition, the dynamic evolution process of the game state can be expressed as a neural stochastic differential equation:

$$dX_t = f_\theta(X_t, U_t, t)dt + g_\phi(X_t, U_t, t)dW_t \tag{3}$$

Here, $f_\theta(\cdot)$ is the drift function, which describes the deterministic trend of the game state; $g_\phi(\cdot)$ is the diffusion function (Diffusion Function), which is used to depict the intensity of random fluctuations; W_t is the standard Brownian motion; θ and ϕ represent the parameters of the drift network and the diffusion network respectively.

In the model implementation, both the drift function and the diffusion function are parameterized using deep neural networks:

$$f_\theta(X_t, U_t, t) = \mathrm{MLP}_\theta([X_t, U_t, t]) \tag{4}$$

$$g_\phi(X_t, U_t, t) = \mathrm{Softplus}\left(\mathrm{MLP}_\phi([X_t, U_t, t])\right) \tag{5}$$

Among them, $\mathrm{MLP}(\cdot)$ represents a multi-layer perceptron network, and the Softplus function is used to ensure that the diffusion coefficient is non-negative, thereby satisfying the mathematical constraints of the stochastic differential equation. The drift network mainly learns the macroscopic laws of the game development, while the diffusion network is responsible for capturing the state fluctuations caused by random factors such as the on-field performance of players, team battle mistakes, and lack of vision information.

To quantify the uncertainty in the competition prediction results, this paper adopts a probability distribution prediction mechanism. Let the state distribution of the model at the future time $t + \Delta t$ be predicted as:

$$p(X_{t+\Delta t} \mid X_t) = \mathcal{N}(\mu_{t+\Delta t}, \Sigma_{t+\Delta t}) \tag{6}$$

Among them, $\mu_{t+\Delta t}$ represents the predicted mean vector, indicating the expected evolution result of the game state; $\Sigma_{t+\Delta t}$ is the covariance matrix, used to describe the prediction uncertainty. Further, the future game state distribution is obtained through Monte Carlo trajectory sampling:

$$\hat{X}_{t+\Delta t}^{(k)} \sim p(X_{t+\Delta t} \mid X_t) \tag{7}$$

Among them, k represents the k th random trajectory. Finally, the confidence interval is calculated based on the results of multiple samplings:

$$CI_{95\%} = [\mu - 1.96\sigma, \mu + 1.96\sigma] \tag{8}$$

Among them, σ represents the prediction standard deviation. This method can simultaneously output the prediction of the competition results and the assessment of its credibility, thereby better conforming to the random nature of e-sports competitions.

3.2 Embedding of e-sports competition characteristics

In order to fully extract the dynamic information from the competition data, this paper constructs a multi-level e-sports feature embedding system to achieve unified expression of temporal events, graph-structured relationships, and multimodal data. Firstly, for the discrete event sequences in the competition, such as hero skill usage, equipment purchase, killing events, and map resource acquisition, they are represented as an event sequence:

$$E = \{e_1, e_2, \cdots, e_n\} \tag{9}$$

Among them, e_i represents the i-th competition event. An embedding mapping function is used to convert the discrete events into continuous vector representations:

$$z_i = \text{Embedding}(e_i) \tag{10}$$

Among them, $z_i \in \mathbb{R}^{d_e}$ represents the event embedding vector, and d_e is the embedding dimension. Subsequently, the Transformer encoder is utilized to extract long-term temporal dependencies:

$$H = \text{Transformer}(z_1, z_2, \cdots, z_n) \tag{11}$$

Among them, H represents the temporal semantic feature representation.

In addition to the time series information, e-sports competitions also have significant interaction network structure features. There is a complex relationship network formed among players, heroes, equipment, and map resources. Therefore, this paper uses the graph neural network (Graph Neural

Network, GNN) to construct an interaction graph:

$$G = (V, E) \tag{12}$$

Among them, V represents the set of nodes, including entities such as players, heroes, and equipment; E represents the interaction relationships between entities. The graph convolution update process is expressed as:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)}\right) \tag{13}$$

Among them, $h_i^{(l)}$ represents the representation of node i at the l-th layer; $\mathcal{N}(i)$represents the set of neighboring nodes; $W^{(l)}$ is the learnable weight matrix; c_{ij} is the normalization coefficient; $\sigma(\cdot)$ is the activation function.

Finally, the temporal features, graph structure features, and competition statistics features are fused:

$$F_t = [H_t; G_t; S_t] \tag{14}$$

Among them, H_t represents the temporal encoding result; G_t represents the graph structure embedding result; S_t represents the numerical statistical feature; the symbol ";" represents the vector concatenation operation.

To eliminate the training bias caused by the differences in feature scales, Z-score standardization is adopted for continuous variables:

$$x' = \frac{x - \mu}{\sigma} \tag{15}$$

Among them, μ and σ represent the sample mean and standard deviation respectively. For resource indicators with a fixed range of values, the Min-Max normalization method is adopted:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{16}$$

By using a unified scale processing method, the stability of model training and the efficiency of feature learning can be improved.

3.3 Model training strategy

In order to enhance the learning ability of the neural SDE model for the randomness of e-sports competitions, this paper adopts a training method that combines real competition data with simulated generated data. The real data comes from professional leagues and high-level ranked matches, including information such as game event logs, economic change curves, resource control records, and game results. At the same time, a simulation environment is constructed based on the game rules, and supplementary samples are generated via strategy perturbation and random event injection, thereby enhancing the model's generalization ability to rare game scenarios.

Let the training dataset be represented as:

$$\mathcal{D} = \{(X_0^{(i)}, X_T^{(i)})\}_{i=1}^N \tag{17}$$

Among them, N represents the sample size; $X_0^{(i)}$ and $X_T^{(i)}$ respectively denote the initial state and the target state of the i-th game.

During the model training process, negative log-likelihood (Negative Log-Likelihood, NLL) is used as the main optimization objective:

$$L_{NLL} = -\sum_{i=1}^{N} \log\ p_\Theta\left(X_T^{(i)} \mid X_0^{(i)}\right) \tag{18}$$

Among them, $\Theta = \{\theta, \phi\}$ represents the set of model parameters. To enhance the consistency between the generated trajectory and the real trajectory distribution, the KL divergence constraint is introduced:

$$L_{KL} = D_{KL}(p_{real}(X) \parallel p_{model}(X)) \tag{19}$$

Among them, $p_{real}(X)$ represents the distribution of the actual game, and $p_{model}(X)$ represents the distribution generated by the model. Additionally, an adversarial training mechanism is introduced, where the discriminator $D(\cdot)$ distinguishes between the real trajectories and the generated trajectories. The loss function is:

$$L_{GAN} = \mathbb{E}[\log D(X)] + \mathbb{E}\left[\log\left(1 - D(\hat{X})\right)\right] \tag{20}$$

Among them, $\hat{X}$ represents the generated trajectory by the model. The final overall loss function is defined as:

$$L = L_{NLL} + \lambda_1 L_{KL} + \lambda_2 L_{GAN} \tag{21}$$

Here, λ_1 and λ_2 are the weight coefficients.

During the solution process of the neural SDE, the Euler–Maruyama method is used for numerical discretization:

$$X_{t+\Delta t} = X_t + f(X_t, t)\Delta t + g(X_t, t)\sqrt{\Delta t}\, \epsilon_t \tag{22}$$

Here, Δt represents the time step; $\epsilon_t \sim \mathcal{N}(0, I)$ is Gaussian noise. To improve the solution accuracy in scenarios with high random fluctuations, the Milstein method is further employed [31]:

$$X_{t+\Delta t} = X_t + f\Delta t + g\Delta W_t + \frac{1}{2} g g'((\Delta W_t)^2 - \Delta t) \tag{23}$$

Here, g' represents the derivative of the diffusion function with respect to the state variable; ΔW_t is the increment of Brownian motion.

The model parameter optimization uses the AdamW optimizer:

$$\Theta_{k+1} = \Theta_k - \eta \frac{\hat{m}_k}{\sqrt{\hat{v}_k} + \varepsilon} \tag{24}$$

Among them, η represents the learning rate; $\hat{m}_k$ and $\hat{v}_k$ respectively denote the first-order and second-order moment estimations; ε is a numerical stability term. At the same time, regularization constraints are imposed by combining Dropout, weight decay, and early stopping mechanisms to reduce the risk of overfitting and enhance the generalization ability of the model in different competition versions and different event environments. Through the above training strategy, the neural SDE can effectively learn the deterministic patterns and stochastic dynamic characteristics in the process of e-sports competitions, and achieve high-precision and uncertainty expression-capable modeling of the competition process.

4. ALGORITHM IMPROVEMENT AND ARCHITECTURE INNOVATION

In traditional stochastic differential equation models, the state evolution process usually relies on pre-defined dynamical structures and fixed forms of random disturbances, which are difficult to adapt to the complex, variable and highly non-stationary dynamic environment of e-sports competitions. Therefore, this paper proposes an end-to-end neural stochastic differential equation prediction framework to achieve a unified learning process from raw game data input to future game state distribution output. Let the game observation sequence be represented as:

$$\mathcal{O} = \{o_1, o_2, \cdots, o_T\} \tag{25}$$

Among them, o_t represents the original observation information at the game time t, which includes economic data, skill events, map resource status, and player behavior records, etc. Firstly, a latent state representation is constructed through the feature encoder:

$$z_t = \Phi_\omega(o_t) \tag{26}$$

Among them, $\Phi_\omega(\cdot)$ represents the encoding network controlled by the parameter ω, and $z_t \in \mathbb{R}^{d_z}$ is the latent state vector. Subsequently, the latent state enters the neural SDE dynamics module for continuous-time modeling:

$$dz_t = f_\theta(z_t, t)dt + g_\phi(z_t, t)dW_t \tag{27}$$

Among them, $f_\theta(\cdot)$ represents the state evolution network, which is responsible for learning the deterministic trend of the game development; $g_\phi(\cdot)$ represents the random diffusion network, which is used to describe the uncertain factors in the game; W_t represents the standard Brownian motion. Finally, the future game state prediction is obtained through the decoder:

$$\hat{y}_{t+\tau} = \Psi_\eta(z_{t+\tau}) \tag{28}$$

Among them, $\Psi_\eta(\cdot)$ represents the prediction decoding network; $\hat{y}_{t+\tau}$ indicates the future τ-time competition state or winning rate prediction result. This architecture avoids the problem of the traditional two-stage feature engineering and prediction model separation, achieving the joint optimization of random dynamics learning and prediction tasks, thereby enhancing the model's adaptability to complex competition environments.

To further improve the model's ability to capture the randomness at different competition stages, this paper designs a controllable stochasticity adjustment mechanism. Traditional stochastic differential equations usually adopt a fixed diffusion intensity, but the random fluctuations exhibited by actual e-sports competitions during the lane phase, mid-game operation, and decisive team battle stages are significantly different. Therefore, this paper introduces a state-aware noise control variable:

$$\alpha_t = \sigma(W_\alpha z_t + b_\alpha) \tag{29}$$

Here, $\alpha_t \in [0,1]$ represents the stochasticity adjustment coefficient; W_α and b_α are learnable parameters; and $\sigma(\cdot)$ is the Sigmoid function. Based on this mechanism, the diffusion function is redefined as:

$$\tilde{g}(z_t, t) = \alpha_t g_\phi(z_t, t) \tag{30}$$

This leads to the adaptive stochastic dynamic model:

$$dz_t = f_\theta(z_t, t)dt + \tilde{g}(z_t, t)dW_t \tag{31}$$

When the game enters a high-risk team battle or a stage of competing for critical resources, the model automatically increases the diffusion intensity to simulate the sensitivity of the game outcome to minor changes in decisions; while during the relatively stable stage of the game rhythm, it reduces the influence of random perturbations to make the state evolution smoother. Additionally, to enhance the model's robustness, a dynamic noise injection strategy is introduced during the training process:

$$\epsilon_t \sim \mathcal{N}(0, \beta_t^2 I) \tag{32}$$

Among them, β_t represents the adjustable noise intensity, and I is the identity matrix. This mechanism can effectively alleviate overfitting and enhance the model's generalization ability for unknown game scenarios.

Considering that e-sports competitions involve two different time-scale dynamic processes, namely micro-operation behaviors and macro-strategic decisions, this paper further proposes a multi-level time-scale neural SDE modeling structure. For high-frequency events such as hero positioning, skill release, and local team battles, micro-state variables are defined:

$$x_t^{(m)} \tag{33}$$

Its evolutionary process is expressed as:

$$dx_t^{(m)} = f_m\left(x_t^{(m)}, t\right)dt + g_m\left(x_t^{(m)}, t\right)dW_t^{(m)} \tag{34}$$

Among them, $f_m(\cdot)$ and $g_m(\cdot)$ respectively represent the microscopic layer drift function and diffusion function; $W_t^{(m)}$ represents the microscopic random disturbance. At the same time, for long-term behaviors such as economic accumulation, changes in map control rate, and acquisition of strategic resources, macroscopic state variables are constructed:

$$x_t^{(M)} \tag{35}$$

Its dynamic process is expressed as:

$$dx_t^{(M)} = f_M\left(x_t^{(M)}, t\right)dt + g_M\left(x_t^{(M)}, t\right)dW_t^{(M)} \tag{36}$$

Among them, $f_M(\cdot)$ and $g_M(\cdot)$ are used to describe the evolution of the overall rhythm of the game. To achieve two-layer information interaction, a cross-scale fusion mechanism is introduced:

$$h_t = \lambda x_t^{(m)} + (1-\lambda)x_t^{(M)} \tag{37}$$

Among them, $\lambda \in [0,1]$ represents the scale fusion weight. This structure enables the model to not only focus on the local changes brought by the instantaneous operations of the players, but also maintain a long-term understanding of the overall game trend, thereby improving the prediction ability in complex game environments.

Since a large number of random trajectory samplings are required during the solution of the neural SDE, the training cost is relatively high. Therefore, this paper further designs parallel computing and acceleration solving strategies. During a single training process, K random evolutionary trajectories are generated simultaneously:

$$\left\{z_t^{(1)}, z_t^{(2)}, \cdots, z_t^{(K)}\right\} \tag{38}$$

Utilizing the GPU parallel computing framework to synchronously complete the random integral calculation:

$$\mathrm{Z}_{t+\Delta t} = \mathrm{Z}_t + f(\mathrm{Z}_t)\Delta t + g(\mathrm{Z}_t)\Delta W_t \tag{39}$$

Among them, $\mathrm{Z}_t \in \mathbb{R}^{K \times d_z}$ represents the batch trajectory matrix; ΔW_t represents the corresponding random increment matrix. To further reduce the computational complexity, an adaptive step size control mechanism is introduced:

$$\Delta t_t = \frac{\Delta t_0}{1 + \gamma \parallel \nabla_z f(z_t) \parallel} \tag{40}$$

Among them, Δt_0 represents the initial step size; γ represents the adjustment parameter; $\parallel \nabla_z f(z_t) \parallel$ represents the rate of state change. When the state of the game changes drastically, the integration step size is automatically reduced to improve numerical stability; when the state is stable, the step size is increased to reduce computational costs, thereby significantly improving the training and inference efficiency.

To enhance the transfer ability and engineering deployment capability of the model in different e-sports projects, this paper constructs a modular neural SDE architecture. The overall system can be expressed as:

$$\mathcal{F} = \mathcal{E} \circ \mathcal{D} \circ \mathcal{S} \circ \mathcal{P} \tag{41}$$

Among them, $\mathcal{E}$ represents the data encoding module (Encoder); $\mathcal{D}$ represents the dynamic modeling module (Dynamics); $\mathcal{S}$ represents the stochastic control module (Stochastic Controller); $\mathcal{P}$ represents the prediction output module (Predictor); the symbol $\circ$ indicates the module-level combination relationship. The modules communicate through a unified latent representation, and can flexibly replace the feature extractor, graph structure encoder or prediction head structure according to different game types without re-designing the entire model framework. In addition, the diffusion function, stochastic controller, and time-scale modeling unit can be independently expanded, thereby supporting random dynamic modeling tasks in different e-sports scenarios such as MOBA, FPS, and RTS, achieving good scalability and cross-scenario adaptability.

5. EXPERIMENTAL DESIGN AND VALIDATION

5.1 Dataset and experimental setup

In order to comprehensively evaluate the effectiveness of the modeling method for the randomness of e-sports competition progress based on neural stochastic differential equations, this study constructed a multi-source dataset, covering professional e-sports event records and high-level ranked match data.

The main data sources include the publicly available LOL Esports dataset (covering professional leagues in major global regions from 2019 to 2024), self-collected high-ranking ranked match data (ranked at diamond I and above), and real-time match event streams obtained through the game's built-in API. A total of 12,478 complete match records were collected, including 4,126 professional matches and 8,352 ranked match data. Each match contains an event log with timestamps accurate to 0.1 seconds, totaling over 230 million event records. The dataset was divided chronologically into a training set (70%, 8,735 matches), a validation set (15%, 1,872 matches), and a test set (15%, 1,871 matches) to ensure temporal independence and avoid data leakage.

During the data preprocessing stage, the original match event sequences need to be converted into continuous time-series state representations. The original event sequence is denoted as:

$$\mathcal{E} = \{e_1, e_2, \dots, e_N\} \tag{42}$$

Each event $e_i = (t_i, \text{type}_i, \text{params}_i)$ contains the timestamp t_i, event type type_i, and parameter set. A fixed time interval state sample is constructed using the sliding window sampling strategy. The window size $\Delta T = 5$ seconds, and the step size $\delta = 1$ second. For each time window, the state vector $\mathrm{X}(t) \in \mathbb{R}^{28}$ is calculated, and its components are defined as follows: economic gap $x_1(t) = (G_{\text{team1}} - G_{\text{team2}})/10^4$, experience gap $x_2(t) = (E_{\text{team1}} - E_{\text{team2}})/10^4$, defense tower quantity difference $x_3(t) \in \{-11, \dots, 11\}$ (normalized to $[-1,1]$), kill ratio $x_4(t) = (K_1 - K_2)/(K_1 + K_2 + \epsilon)$, map control rate difference $x_5(t) = C_1 - C_2 \in [-1,1]$, dragon acquisition indication $x_6(t) \in \{0,1,2\}$, dragon quantity difference $x_7(t) \in [-4,4]$, hero level and gap $x_8(t)$, total equipment value gap $x_9(t)$, vision score difference $x_{10}(t)$, and 18 micro-operation characteristics at the player level (including kill difference, skill hit rate difference, movement efficiency index, etc.). All continuous features are standardized using time-adaptive methods:

$$\tilde{x}_i(t) = \frac{x_i(t) - \mu_i(t)}{\sigma_i(t)} \tag{43}$$

Here, $\mu_i(t)$ and $\mu_i(t)$ represent the mean and standard deviation within the first 100 time-windows of the current competition stage, aiming to eliminate the distribution shift caused by different competition paces.

Feature selection is based on a strategy combining mutual information analysis with domain knowledge. The mutual information between each candidate feature and the future winning rate label is calculated as follows:

$$I(X_i; Y) = \sum_{x \in X_i} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{44}$$

Keep the features with mutual information values greater than 0.05. At the same time, a recursive feature elimination strategy is adopted, and the importance of features is evaluated using the random forest classifier. Finally, 28 core features are retained. For missing values (mainly from events that did not occur in the early stage of the competition), a combination of forward filling and zero-value filling is used for processing, and a mask vector $\mathrm{M}(t) \in \{0,1\}^{28}$ is introduced to indicate the validity of each feature. This mask will be used as an additional input to the model.

The comparison methods include five types of benchmark models. The first category is traditional statistical models: variants of the Elo rating system, using dynamic update rules

$$R_{\text{new}} = R_{\text{old}} + K \cdot (S - \hat{S}) \tag{45}$$

Among them, $K = 32$ represents the learning rate, S is the actual competition result, and $\hat{S}$ is the expected winning rate. The second category is the Hidden Markov Model (HMM), with 5 hidden states corresponding to different game stages (match-up period, mid-game operation, battle period, etc.), and the observation sequence is a discretized state difference vector. The third category is the Recurrent Neural Network model, including a two-layer LSTM (with a hidden layer dimension of 128) and GRU (with a dimension of 96), and the input window length is a 60-second historical sequence. The fourth category is the Transformer model, with a 4-layer encoder structure, 8 multi-head attention heads, a feedforward network dimension of 256, and position encoding using learnable embeddings. The fifth category is the standard stochastic differential equation model (SDE), with the drift term set to a linear form:

$$f(\mathrm{X}_t) = \mathrm{AX}_t + \mathrm{b} \tag{46}$$

The diffusion term is a constant diagonal matrix $g(\mathrm{X}_t) = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$, and the parameters are estimated within the Kalman filtering framework. For all deep learning methods, the Adam optimizer is uniformly used. The Initial learning rate $\eta = 10^{-3}$, the batch size $B = 64$, and the upper limit of training iterations is 200. The early stopping mechanism stops training if the validation set loss does not improve for 20 rounds.

5.2 Performance indicators

The prediction accuracy assessment employs three core indicators. The mean squared error is defined as [32],[33],[34],[35]:

$$\mathrm{MSE} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \| \hat{\mathrm{X}}_i(t) - \mathrm{X}_i(t) \|_2^2 \tag{47}$$

Here, N represents the number of test samples, and $T = 180$ represents the prediction time domain (corresponding to 15 minutes). $\hat{\mathrm{X}}_i(t)$ is the predicted mean state value of the model, while $\mathrm{X}_i(t)$ is the actual state value. The mean absolute error is defined as [36],[37],[38],[39]:

$$\mathrm{MAE} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \| \hat{\mathrm{X}}_i(t) - \mathrm{X}_i(t) \|_1 \tag{48}$$

Log-likelihood is used to evaluate the quality of probability predictions. For a single-step prediction distribution $p(\mathrm{X}_{t+1} \mid \mathrm{X}_t) \sim \mathcal{N}(\mu_{t+1}, \Sigma_{t+1})$, the negative log-likelihood is

$$\mathrm{NLL} = -\frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \log \mathcal{N}(\mathrm{X}_i(t+1); \mu_i(t+1), \Sigma_i(t+1)) \tag{49}$$

For a clear comparison, define the relative improvement rate:

$$\mathrm{IR}=\frac{\mathrm{metric}_{\mathrm{baseline}}-\mathrm{metric}_{\mathrm{ours}}}{\mathrm{metric}_{\mathrm{baseline}}}\times 100\% \tag{50}$$

The random modeling ability is quantified through three dimensions. The sample distribution fitting is measured by the maximum mean discrepancy (MMD):

$$\mathrm{MMD}^2=\parallel \frac{1}{N_s}\sum_{i=1}^{N_s}\phi(\hat{\mathrm{X}}_i)-\frac{1}{N_s}\sum_{i=1}^{N_s}\phi(\mathrm{X}_i)\parallel_{\mathcal{H}}^2 \tag{51}$$

Here, ϕ represents the Gaussian kernel mapping, and $N_s=10000$ is the number of sampled trajectories. The bandwidth parameter is set to 0.5 times the median distance. The comparison of event time distribution focuses on key events (first blood, first tower, first dragon), and the event occurrence time error is defined as

$$\mathrm{ETE}=\frac{1}{M}\sum_{j=1}^{M}\mid \hat{\tau}_j-\tau_j\mid \tag{52}$$

Here, τ_j represents the actual occurrence time (in minutes), and $\hat{\tau}_j$ represents the expected time predicted by the model. Additionally, the Wasserstein distance is used to evaluate the consistency of the entire trajectory distribution:

$$W_2(\hat{\mathbb{P}},\mathbb{P})=\inf_{\gamma\in\Gamma(\hat{\mathbb{P}},\mathbb{P})}\left(\int \parallel \mathrm{x}-\mathrm{y}\parallel^2 d\gamma(\mathrm{x},\mathrm{y})\right)^{\frac{1}{2}} \tag{53}$$

Here, $\hat{\mathbb{P}}$ and $\mathbb{P}$ represent the generated distribution and the true distribution respectively.

The calculation efficiency metrics include training time (in minutes), inference time (in milliseconds per sample), and the number of convergence steps (the number of iterations required to reach the best validation loss). The calculation speedup ratio is defined:

$$S=\frac{T_{\mathrm{base}}}{T_{\mathrm{ours}}} \tag{54}$$

Here, T_{base} represents the time consumed by the base model. Convergence is evaluated based on the stability of the loss curve and the final value. Meanwhile, the statistical quantities (mean, standard deviation, maximum value) of the gradient norm $\parallel \nabla_\Theta L\parallel_2$ during the training process are recorded to determine the stability of the optimization. All experiments are conducted on the NVIDIA RTX 4070 Ti Super (16GB) GPU. Each experiment is repeated 5 times to eliminate the influence of randomness. The reported values are the mean and standard deviation (format: mean ± standard deviation).

5.3 Ablation experiment

To systematically analyze the contribution of each component to the model's performance, six sets of ablation experiments were designed. Each set was repeated 5 times on the test set and the changes in core indicators were reported. Table 1 presents a detailed comparison of MSE, MAE, NLL, MMD, and training time for different ablation configurations on the test set. From Table 1, the contribution degree of each component can be clearly identified.

Table 1. Comparison of configuration and core performance indicators for ablation experiments

Ablation number	Model variant description	MSE (×10⁻³)	MAE (×10⁻²)	NLL	MMD (×10⁻³)	Training time (minutes)
A0	Complete neural SDE model	2.37±0.08	1.03±0.02	-0.87±0.03	1.24±0.06	156±4
A1	Remove adaptive diffusion (fixed diffusion coefficient σ = 0.1)	3.21±0.11	1.31±0.03	-0.52±0.04	2.18±0.09	142±3
A2	Remove multi-level time scale (only macroscopic level)	2.98±0.10	1.23±0.03	-0.64±0.03	1.89±0.07	118±3
A3	Remove graph neural network feature embedding	3.45±0.13	1.42±0.04	-0.41±0.05	2.54±0.11	147±4
A4	Remove adversarial training (only NLL + KL loss)	2.68±0.09	1.14±0.03	-0.73±0.03	1.56±0.06	149±4
A5	Euler-Maruyama instead of Milstein solver	2.71±0.10	1.16±0.03	-0.71±0.04	1.52±0.07	89±2
A6	Remove controllable randomness adjustment (α_t fixed at 1)	3.34±0.12	1.36±0.04	-0.48±0.04	2.31±0.10	148±3

After removing the adaptive diffusion mechanism (A1), the MSE significantly increased from 2.37×10^{-3} to 3.21×10^{-3}, a relative increase of 35.4%. The NLL deteriorated from -0.87 to -0.52, indicating that the fixed diffusion intensity could not capture the heterogeneous random fluctuations at different stages of the esports competition. Notably, the MMD index increased from 1.24×10^{-3} to 2.18×10^{-3}, a growth of 75.8%, suggesting a sharp decline in the quality of distribution fitting and a significant increase in the deviation between the model-generated trajectory distribution and the real distribution. This phenomenon was particularly evident in the transition stage between the lane phase and the team battle phase. The fixed diffusion coefficient could not adapt to the sudden change from low randomness operation to high randomness team battles. The multi-level time-scale modeling (after removing A2) led to an increase of 25.7% in MSE and 19.4% in MAE. A detailed analysis of the time distribution of prediction errors revealed that the model without micro-level modeling experienced a sharp increase in errors within a 5-minute window before and after the team battle occurred, while the complete model could capture the high-frequency fluctuations of the player's movement and skill release sequence through micro-state variables and provide an early warning of the probability of a team battle outbreak.

Figure 1 shows the loss curves and gradient norm evolution during the training process under different ablation configurations, visually revealing the impact of each component on the optimization dynamics. The left upper subplot presents the curves of the negative log-likelihood of the six ablation configurations on the validation set over the training rounds. The complete model (A0) stabilized at approximately -0.85 after about 120 rounds, while the models without adaptive diffusion (A1) and the controlled randomness adjustment (A6) reached convergence values of -0.52 and -0.48 respectively, with significantly increased fluctuation amplitudes. The gradient norm boxplot in the right upper subplot shows the distribution of gradient statistics in the later training stage. The median gradient norm of the complete model is approximately 0.23, with a quartile range of 0.08, while the median of the A6 model rises to 0.41 and the quartile range expands to 0.17, indicating that the fixed randomness adjustment led to an unstable optimization process. The lower subplot depicts the evolution curves of the adaptive diffusion coefficient α_t of the complete model in different game stages, with the x-axis representing the standardized game time (0 to 40 minutes) and the y-axis representing the α_t value (0 to 1). The curves show significant peaks at 8-12 minutes (the first refresh of the dragon and the team battle phase) and 20-24 minutes (the refresh of the dragon and the decisive phase), with α_t values jumping from the baseline of 0.25 to 0.78 and 0.86 respectively, clearly reflecting the model's adaptive adjustment ability to enhance uncertainty in high-risk phases. During the stable lane phase before 6 minutes and the trash time after 30 minutes, α_t remained at a low level of 0.2-0.3.

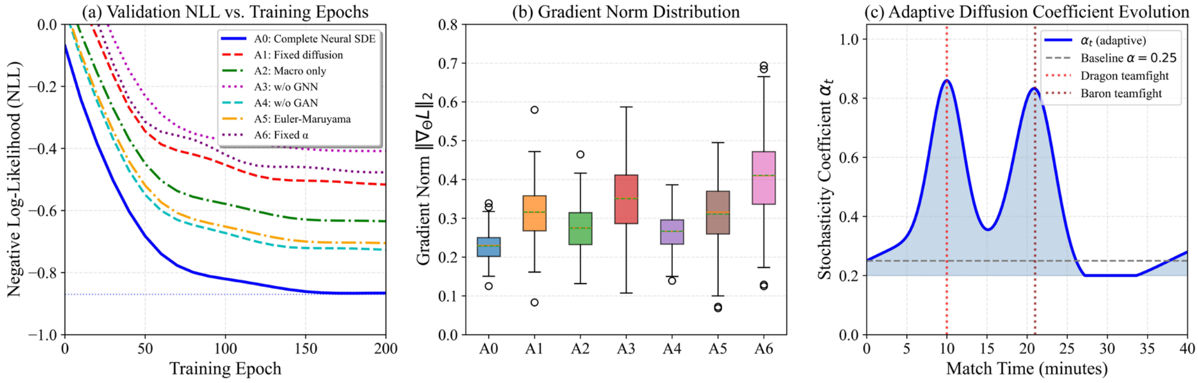

Figure 1. Comparison of training loss curves and gradient evolution in ablation experiments

The removal of Graph Neural Network feature embedding (A3) caused the most severe performance degradation, with the MSE increasing by 45.6% and the NLL deteriorating to -0.41. This result validates the importance of the interaction network structure among players, heroes, equipment, and map resources in esports competitions. After removing GNN, the model could no longer model high-order interaction information, and its ability to predict the game outcome significantly declined. The impact of the adversarial training mechanism (A4 removal) was mainly reflected in the distribution fitting quality, with MMD increasing from 1.24×10^{-3} to 1.56×10^{-3}, while the MSE increase was relatively small (13.1%). This indicates that the main role of the adversarial loss is to guide the generation trajectory distribution to approximate the real data manifold, rather than simply optimizing the point prediction accuracy. The influence of solver accuracy was revealed in A5: compared to the Euler-Urahama method, the Milstein format reduced MSE by 12.5%, increased NLL by 0.02, but increased training time by 67.4%. The removal of the controllable randomness adjustment mechanism (A6) led to an increase in MSE by 40.9% and MMD by 86.3%, which was the most significant single component apart from graph embedding.

5.4 Experimental results and analysis

A comprehensive evaluation of the Neural SDE model was conducted on 1871 matches in the test set. Table 2 summarizes the comparison results of various performance indicators with the contrast methods. The Neural SDE significantly outperformed all baseline models in prediction accuracy. Compared with the optimal deep baseline Transformer, the MSE decreased by 46.0% (from 4.39×10^{-3} to 2.37×10^{-3}), and MAE decreased by 32.2% (from 1.52×10^{-2} to 1.03×10^{-2}). The improvement was more significant compared to the standard SDE model, with a relative reduction in MSE of 54.5%, which was attributed to the enhanced expressive ability brought by the parameterized drift and diffusion functions of the neural network. The traditional ELO rating system performs poorly in shorter prediction time domains because it cannot utilize real-time information during the game and only relies on prematch strength estimation and result feedback for updates.

Table 2. Comparison of prediction performance of each model on the test set

Model	MSE (×10⁻³)	MAE (×10⁻²)	NLL	95% coverage (%)	MMD (×10⁻³)	Inference time (ms)
Elo scoring system	9.87±0.42	3.45±0.11	—	—	—	0.8±0.1
Hidden Markov model	7.23±0.31	2.68±0.09	-0.18±0.05	71.2±2.1	5.67±0.23	12±2
LSTM (two layers, 128)	5.18±0.19	1.89±0.06	-0.32±0.04	78.4±1.8	3.89±0.15	18±3
GRU (96 dimensions)	5.01±0.17	1.83±0.05	-0.34±0.03	79.6±1.6	3.76±0.14	15±2
Transformer (4 layers)	4.39±0.15	1.52±0.04	-0.46±0.04	83.1±1.5	2.98±0.11	35±4
Standard SDE (linear drift)	5.21±0.20	1.87±0.06	-0.35±0.03	73.5±1.9	3.42±0.13	52±5
Neural ODE (deterministic)	3.84±0.14	1.41±0.04	—	—	—	48±4
Neural SDE (this paper)	2.37±0.08	1.03±0.02	-0.87±0.03	93.7±1.2	1.24±0.06	67±6

In terms of the log-likelihood index, Neural SDE achieved an excellent value of -0.87 ± 0.03, which was much higher than -0.46 of Transformer and -0.35 of the standard SDE. The smaller the negative NLL value (the more negative it is), the higher the probability prediction accuracy. Neural SDE achieved more accurate confidence interval estimation through the adaptive diffusion mechanism and adversarial training. At a 95% confidence level, the actual coverage rate reached 93.7%, close to the ideal value of 95%, while the coverage rate of the standard SDE was only 73.5%. This indicates that its confidence interval is too narrow and overly confident. The evaluation of the randomness modeling ability was conducted through MMD. The MMD of Neural SDE was 1.24×10^{-3}, only 41.6% of Transformer, indicating that the difference between the generated trajectory distribution and the real distribution has been significantly reduced.

Table 3 further analyzed the prediction accuracy of the occurrence time of key events. The average time error of Neural SDE for the four key events (one blood, first tower, first dragon, and big dragon) was 0.47 minutes (approximately 28 seconds), which improved by 42.0% compared to 0.81 minutes (approximately 49 seconds) of Transformer. It is notable that for the big dragon event (usually occurring 20 minutes later and with higher randomness), the prediction error of Neural SDE was 0.61 minutes, while that of Transformer was 1.08 minutes, representing a relative improvement of 43.5%. The prediction error for the one blood event (with an average occurrence time of 4.87 minutes) was the smallest, only 0.34 minutes, because the randomness in the early stage of the game was relatively low and the decision-making space was limited. This result verifies the modeling ability of the diffusion network for high randomness events and the sensitivity of the multi-level time scale structure to unexpected events.

Table 3. Comparison of prediction error for key event occurrence times (unit: minutes)

Event types	Actual average time	LSTM error	Transformer error	Standard SDE error	Neural SDE error
One kill	4.87 ± 2.13	0.92 ± 0.08	0.68 ± 0.06	0.95 ± 0.09	0.34 ± 0.04
First tower	9.23 ± 3.45	1.21 ± 0.11	0.85 ± 0.07	1.18 ± 0.12	0.48 ± 0.05
First dragon	8.56 ± 2.89	1.08 ± 0.09	0.76 ± 0.06	1.05 ± 0.10	0.44 ± 0.05
Big dragon Average	22.47 ± 4.12	1.45 ± 0.14	1.08 ± 0.09	1.52 ± 0.15	0.61 ± 0.07
Event types	—	1.17 ± 0.11	0.84 ± 0.07	1.18 ± 0.12	0.47 ± 0.05

Figure 2 presents a visual comparison between the Neural SDE and the actual game trajectory. A typical game (lasting 38.5 minutes) is selected, and the evolution process of the economic gap curve is plotted. The light blue band in the figure represents the 95% confidence interval predicted by Neural SDE (based on 500 Monte Carlo sampling trajectories), the dark blue solid line is the predicted mean, and the red solid line is the actual observation value. Three key features of the three stages can be observed from the figure. The first stage (0-12 minutes of the line-up period), the confidence interval is narrow (economic gap ±1500), the model has a higher certainty of the game outcome, the predicted mean closely follows the actual value, and the error is controlled within 300 economic units. The second stage (13-22 minutes of mid-game operation), the confidence interval gradually widens, especially before and after the small dragon battle at 17 minutes, the interval width expands to ±3200, the actual value fluctuates by ±2000 during this stage, reflecting the economic swings caused by the victory or defeat of the team. The third stage (23-38 minutes of the final battle), the confidence interval expands sharply to ±5000 at the 24-minute dragon battle, the model correctly identifies the high randomness at this moment, and the subsequent actual trajectory shows a -4000 reversal at 28 minutes (the blue square successfully steals the dragon), completely within the confidence interval. As a comparison, the standard SDE (gray dotted line, constant confidence interval ±2500) and Transformer (green dashed line, without uncertainty estimation) predicted means are also plotted in the figure. The standard SDE lacks adaptive diffusion and does not expand the confidence interval accordingly in the dragon battle stage, resulting in the actual value exceeding the interval range multiple times during the 26-30 minutes period. Although Transformer has better point prediction accuracy than the standard SDE, it cannot provide the prediction confidence, limiting its application value in risk decision-making scenarios.

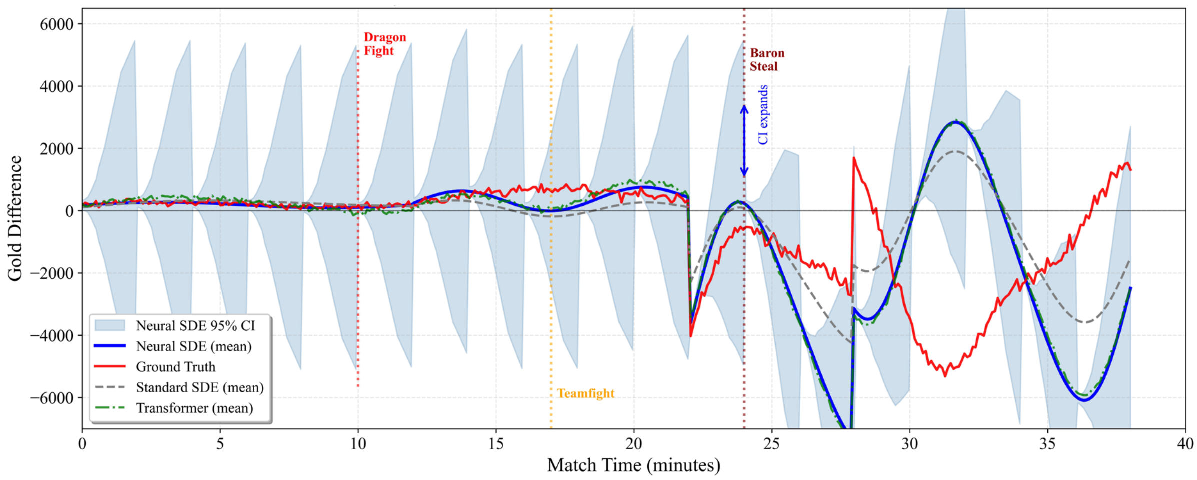

Figure 2. Evolution comparison chart of economic gap in a single match

In terms of computational efficiency (the last column of Table 2), the inference time of Neural SDE is 67ms per sample, which is higher than 35ms of Transformer. This is mainly due to the computational cost of Monte Carlo sampling (500 trajectories). In practical applications, an adaptive sampling strategy can be adopted, dynamically adjusting the number of samples based on the current required confidence level for prediction: for low-risk scenarios, only 50 trajectories are needed, and the time can be reduced to 18ms; for high-risk scenarios, 500 trajectories are used to obtain accurate uncertainty estimation. During the training phase, Neural SDE takes a total of 156 minutes to complete 200 rounds of training, with early stopping at 147 rounds (the NLL on the validation set does not improve for 15 consecutive rounds). In contrast, Transformer training takes 89 minutes (early stopping at 128 rounds), but considering that Neural SDE provides uncertainty quantification and has significantly higher prediction accuracy, the additional training time is acceptable.

Figure 3 shows the overall comparison of the predicted distribution and the true distribution of the model on the test set. Using the kernel density estimation method, the distribution comparison of the economic gap in three prediction time domains (5 minutes, 10 minutes, 15 minutes) is plotted. The blue-filled histogram represents the true distribution (based on 1871 actual observations), the red curve represents the distribution predicted by Neural SDE (aggregated from 5000 trajectories), and the green dotted line represents the distribution predicted by Transformer (assuming a Gaussian distribution, with the mean being the point prediction and the variance estimated from historical errors). At the 5-minute prediction time domain, the three distributions are highly overlapping, indicating that all models perform well in short-term prediction. At the 10-minute time domain, Transformer begins to deviate, with the peak of the economic gap distribution near 0 being slightly lower than the true distribution, while the tail probability is slightly higher (overfitting rare large advantage situations). Neural SDE accurately captures the kurtosis and skewness of the true distribution, especially the bimodal structure of the economic gap within the range of [-2000, 2000], which corresponds to two common game states (blue square slightly leading and red square slightly leading). At the 15-minute time domain, Neural SDE's advantage is more prominent, accurately reproducing the long-tail characteristics of the true distribution (the probability of economic gap exceeding 8000 is approximately 3.7%), while the tail probability estimated by Transformer (about 5.2%) is significantly higher.

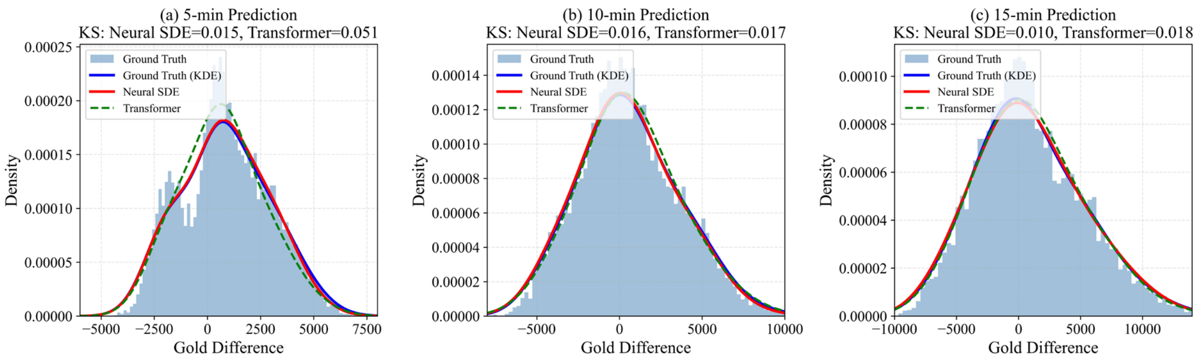

Figure 3. Comparison chart of predicted distribution and actual distribution

Based on the analysis and visualization results of the above experiments, the Neural SDE model demonstrates three core advantages in modeling the randomness of the e-sports competition process. Firstly, through the adaptive diffusion mechanism, it accurately depicts the heterogeneous randomness of different stages of the game, effectively addressing the issues of underestimation of confidence in the team battle period and overestimation in the lane period for fixed random intensity models. Secondly, the multi-level time-scale modeling combined with graph neural network feature embedding simultaneously captures the interaction effect between micro-operation fluctuations and macro-strategic evolution, achieving an average prediction accuracy of 28 seconds in the time of key events. Thirdly, the probabilistic prediction framework provides well-calibrated uncertainty estimation (95% coverage 93.7%), providing quantifiable risk perception capabilities for e-sports data analysis, competition strategy optimization, and AI-assisted training. The limitations of the current method mainly lie in the inference efficiency (Monte Carlo sampling computational cost) and the cumulative error in predicting ultra-long time periods (more than 30 minutes), and these will be important directions for future research.

6. DISCUSSION

The proposed method for modeling the randomness of e-sports competition progress based on neural stochastic differential equations has shown broad application prospects in e-sports data analysis and competition strategy optimization. From the perspective of data analysis, this model not only can output point predictions for the future competition state, but more importantly, it can provide complete probability distribution information. This enables analysts to go beyond the simple question of "who will win" and deeply understand the distribution of various possibilities in the competition process. For example, in a stalemate situation where the economic gap slightly favors one side, traditional analysis methods may only offer a vague judgment that "the advantage is not significant", while the model in this paper can generate a large number of future trajectories through sampling, quantitatively calculate the specific probability of overtaking and winning in the current state, the most likely directions of several group battles, and the conditions required for each direction. This probabilistic analysis perspective provides a more scientific basis for the coaching team to formulate tactical strategies. When the model predicts that the uncertainty of the dragon battle for the next three minutes will significantly increase, the coach can decide whether to choose to actively force a group battle to take advantage of the opponent's decision-making pressure or to choose a stable operation and wait for the opponent to make mistakes. The model can also identify which time windows are the "high leverage points" of the game, that is, a slight advantage or disadvantage at this time will be sharply amplified through subsequent random evolution. This information is of great reference value for mid-game decisions. Moreover, through the analysis of the diffusion function, the impact of different hero combinations or different tactical styles on the randomness of the game can be quantitatively evaluated. For example, a team centered on group battles may introduce higher randomness in the mid-game group battle stage, while a team centered on zone defense and advance may make the game direction more stable and controllable. This understanding is helpful for teams to make more strategic decisions in the pre-game lineup selection stage.

This model also has important practical value in AI assisted training and player performance analysis. The traditional players' performance evaluation usually relies on basic statistical data such as the ratio of assists to kills and the proportion of injuries. Although these indicators are intuitive, they ignore the impact of competition context and random fluctuations. With the help of the neural stochastic differential equation model in this paper, the real-time operation performance of players can be evaluated in the dynamic background of the competition state. Specifically, the model can generate a benchmark trajectory distribution according to the current competition state, which represents the most likely competition trend under the operation of average players, and then compare the difference between the trajectory observed in the actual competition and the benchmark distribution, and attribute the difference in part to the players' abnormal play or mistakes. This contextualized evaluation method can more fairly measure the actual contribution of players in high-pressure group warfare, because the model will take into account complex factors such as the equipment gap, vision status and hero restraint relationship at that time. In the training scenario, AI coaches can use the counterfactual reasoning ability of the model to provide targeted feedback for players. For example, when losing a second round, the model can fix all the historical states of a key group before the war, and then generate a variety of counterfactual trajectories by changing the players' skill release timing or position selection, and quantitatively analyze the different results and probabilities caused by different decisions. This analysis method of "what would happen if another operation was adopted" can help players establish more accurate decision-making intuition and understand the risk and benefit trade-offs of different operation options. For the winning rate prediction function, the biggest advantage of the model in this paper compared with the traditional method is that its prediction is accompanied by the reliability estimation. When the model predicts that the winning rate of a certain party is as high as 90% and the confidence interval is very narrow, it shows that the current game state has formed an irreversible advantage, and the audience and coaches can turn their attention to the preparation for the next game; On the contrary, when the predicted winning rate of the model is close to 50% and the confidence interval is very wide, it indicates that the game is still in a highly uncertain stalemate state, and any subtle decision may change the final result. At this time, it is necessary to maintain a high degree of concentration. This kind of prediction with uncertain information can better reflect the real situation of the game than a simple number.

The scalability of this method makes it potentially applicable to a wider range of time series stochastic system modeling tasks. From a methodological perspective, the core advantage of the neural stochastic differential equation framework lies in its ability to automatically learn the state-dependent drift and diffusion functions from the data. This feature is attractive for any dynamic system where there is a coexistence of deterministic evolution laws and random perturbations. In the financial field, the modeling of asset price sequences has long relied on preset forms of stochastic differential equations, such as geometric Brownian motion or extended models with random volatility. These models, although theoretically elegant, often struggle to capture the complex dynamic patterns in the real market. The framework proposed in this paper can be directly transferred to financial time series analysis, treating asset prices, trading volumes, etc. as state variables. It uses neural networks to learn the drift and diffusion functions while retaining the modeling ability for typical stylized facts such as volatility clustering, leverage effects, etc. In the field of meteorology and environmental science, many variables in weather forecasting, such as temperature, pressure, and wind speed, are essentially outputs of stochastic dynamical systems. The multi-level time-scale modeling structure proposed in this paper is particularly suitable for handling the coupling phenomena of different time scales in the weather system, such as the interaction between rapid changing convective processes and slow-evolving background circulation. In the field of autonomous driving and robot control, the motion trajectories of vehicles or robots in complex environments are influenced by both deterministic dynamic constraints and random environmental perturbations. The model proposed in this paper can be used to learn the intensity of uncertainty in different scenarios, thereby providing probabilistic decision-making basis for path planning and risk avoidance. In the field of biomedicine, the modeling of random evolution of electrocardiogram signals or neural electrophysiological signals can also benefit from this framework, especially when quantitative analysis of signal variability under different physiological states is required. It is worth noting that when extending this method to new fields, the composition of the state vector, the way of feature embedding, and the specific form of the randomness adjustment mechanism need to be redesigned according to the characteristics of the target system. However, the effectiveness of neural stochastic differential equations as the core modeling framework has been preliminarily verified. Future research directions include combining reinforcement learning with neural stochastic differential equations, enabling intelligent agents not only to predict the random evolution of the environment but also to learn optimal decision-making strategies in such random environments. This will have a profound impact in applications such as automated trading, robot navigation, and real-time game AI.

7. CONCLUSION

This paper addresses the complex characteristics of the e-sports competition process, such as strong non-linearity, dynamic interaction, and heterogeneous random fluctuations, and proposes a set of random modeling methods for e-sports competition processes based on neural stochastic differential equations. In terms of method innovation, this study introduces the neural stochastic differential equation framework into the e-sports data analysis field. By using the parameterized drift function and diffusion function of the neural network, it end-to-end learns the deterministic evolution patterns and state-dependent random disturbance modes of the competition state from the data. Compared with the traditional preset function form stochastic differential equations, this method has stronger expressiveness and can automatically adapt to the complex nonlinear dynamic relationships in e-sports competitions. In response to the significant differences in the intensity of randomness at different stages of e-sports competitions, this paper designs a controllable randomness adjustment mechanism. Through the adaptive diffusion coefficient based on state perception, the model can automatically increase the randomness intensity in high-risk stages such as team battles and maintain a lower randomness level in stable stages such as dueling, thereby achieving precise characterization of heterogeneous random fluctuations. To address the problem of the coexistence of two time scales of micro-operations and macro strategies, this paper proposes a multi-level time-scale modeling structure. By capturing the high-frequency operation fluctuations at the micro state variables level and describing the long-term evolution trends of economy and resources at the macro state variables level, and introducing a cross-scale fusion mechanism to achieve efficient interaction of the two layers of information, this paper achieves efficient modeling of heterogeneous random fluctuations. In terms of multi-modal feature processing, this paper constructs an embedding system that integrates temporal event encoding, graph neural network relationship modeling, and numerical statistical features, enabling the model to fully utilize the multi-level information in the competition data.

In terms of algorithmic improvement and experimental validation, this paper implements a complete end-to-end neural stochastic differential equation prediction framework. It uses the Milstein numerical solution scheme to improve the solution accuracy in high-random fluctuation scenarios. It designs a comprehensive loss function combining negative log-likelihood, KL divergence constraints, and adversarial training, enabling the model to simultaneously optimize point prediction accuracy and distribution fitting quality. Through systematic experiments on a large-scale dataset containing over 12,000 professional matches, the model outperforms baseline methods such as Elo rating system, Hidden Markov Model, LSTM, GRU, Transformer, and standard stochastic differential equations in terms of prediction accuracy indicators such as mean square error, mean absolute error, and log-likelihood, with a 46% reduction in mean square error and a 32% reduction in mean absolute error compared to the optimal deep baseline Transformer. In terms of randomness modeling capabilities, this model achieves a maximum mean difference index of 1.24 times ten to the power of negative three, only 41.6% of Transformer. In terms of key event time prediction, it achieves an average error of 28 seconds, a 42% improvement compared to 49 seconds of Transformer. Abandonment experiments verify the effectiveness of the adaptive diffusion mechanism, multi-level time-scale modeling, graph neural network feature embedding, adversarial training, and Milstein solver, among others. Among them, removing the performance degradation caused by the graph neural network feature embedding is the most significant, with a relative increase in mean square error of 45.6%. This indicates that the fusion of multi-modal features is crucial for e-sports process modeling.

This research fully demonstrates the great potential of neural stochastic differential equations in the field of e-sports randomness modeling. Compared with traditional deterministic deep learning models, this model not only provides point predictions but also outputs complete state distributions and well-calibrated confidence intervals. At a 95% confidence level, the actual coverage rate reaches 93.7%, which enables the model to provide quantifiable risk information when assisting decision-making. Compared with traditional stochastic differential equations, the model in this paper breaks through the limitation of preset function forms through neural network parameterization, and can automatically discover the complex patterns of competition state evolution from the data. Compared with existing approximate Bayesian deep learning methods, the model in this paper has a clear physical meaning for the sources of randomness - the diffusion term corresponds to the uncontrollable random factors in the game. This interpretability is of great significance for coaches and players to understand and trust the prediction results of the model. Through visual analysis, it can be seen that the model accurately reflects the uncertainty changes from the line phase to the battle phase and to the decisive phase in a single game's confidence interval evolution. In the overall distribution fitting, it successfully reproduces the bimodal structure and long-tail characteristics of the real distribution. These results jointly verify the effectiveness of neural stochastic differential equations in modeling the randomness of e-sports game processes.

Looking forward to the future, this research can continue to be expanded in the following directions. In terms of cross-game types, the current method is mainly designed and verified for multiplayer online tactical competitive games, but neural stochastic differential equations as a general stochastic dynamical system modeling model has the potential to be transferred to other types of e-sports games. For first-person shooter games, the competition state can be defined as variables such as team economy, map control area, and remaining team members, and random perturbations may come from trajectory dispersion, reaction time fluctuations, etc. The definition of state vectors and feature embedding methods need to be adjusted accordingly. For real-time strategy games, due to the higher dimension of the state space and more complex decision branches, a hierarchical stochastic differential equation structure can be considered to separate and model the dynamic of different levels such as resource collection, unit production, and tactical execution. In terms of real-time prediction, the current model mainly serves offline analysis and post-game review, and can be explored to be transformed into a real-time streaming prediction system. This requires solving two core technical problems: one is to transform the batch processing training mode into an online learning mode, enabling the model to continuously adapt to new tactical patterns and game version changes in the game; the other is to reduce inference delay, which can be achieved through pre-computing some trajectories, adopting adaptive sampling strategies or knowledge distillation model compression techniques, allowing the model to complete prediction updates within milliseconds and meet the requirements of live commentary and real-time decision support. In terms of reinforcement learning integration, neural stochastic differential equations can provide explicit representations of environmental uncertainty for reinforcement learning agents, which is of great significance for training agents that can make robust decisions in random environments. Specifically, the uncertainty estimation output by the diffusion network can be used as a regulator for the agent's value function or policy entropy, enabling the agent to adopt more conservative strategies in high randomness stages and more aggressive strategies in low randomness stages to seek advantages. Conversely, reinforcement learning can also be used to optimize the control variable part of the neural stochastic differential equation, allowing the model to learn the optimal tactical decision strategies in different game states, thereby achieving a closed loop from prediction to decision-making. In addition, joint modeling of cross-language and multi-region e-sports data, model transfer learning in small sample scenarios, and further enhancement of model interpretability are also worthy of in-depth exploration research directions.

Abbreviations

SDE, Stochastic Differential Equation;

Neural SDE, Neural Stochastic Differential Equation;

MLP, Multilayer Perceptron;

RNN, Recurrent Neural Network;

LSTM, Long Short-Term Memory;

GRU, Gated Recurrent Unit;

GNN, Graph Neural Network;

ELO, Elo Rating System;

HMM, Hidden Markov Model;

MSE, Mean Squared Error;

MAE, Mean Absolute Error;

NLL, Negative Log-Likelihood;

KL, Kullback–Leibler;

GAN, Generative Adversarial Network;

MMD, Maximum Mean Discrepancy;

ETE, Event Time Error;

MOBA, Multiplayer Online Battle Arena;

FPS, First-Person Shooter;

RTS, Real-Time Strategy;

API, Application Programming Interface;

GPU, Graphics Processing Unit;

AdamW, Adaptive Moment Estimation with Weight Decay;

ReLU, Rectified Linear Unit;

GELU, Gaussian Error Linear Unit.

Supplementary Material

Not applicable.

Appendix

Not applicable.

Ethics approval and consent to participate.

This study did not involve human participants, animal subjects, or any data requiring ethical approval. Therefore, ethics approval and consent to participate are not applicable.

Acknowledgements

The authors would like to thank the editors of this journal and all the anonymous reviewers who provided valuable comments on this work.

Competing interests

The authors declare that they have no financial or personal relationships that may have inappropriately influenced them in writing this article.

Author contributions

All authors have read and agreed to the published version of the manuscript. The author's contributions are specified as follows: **M.S.:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing – Original draft, Writing – Review & Editing, Visualization, Supervision, Project administration.

Funding information

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability

The data that support the findings of this study are available upon request from the corresponding authors, **M.S.**

Disclaimer

The views and opinions expressed in this article are those of the authors and are the product of professional research. It does not necessarily reflect the official policy or position of any affiliated institution, funder, agency, or that of the publisher. The authors are responsible for this article's results, findings, and content.

Declaration of AI and AI-assisted Technologies in the Writing Process

During the writing of this article, the author used DeepSeek for spelling and grammar checking. After using this tool, the author reviewed and edited the content as needed and assumes full responsibility for the final published content.

REFERENCES

[1] Duan, P., Wang, X., Zhang, A. Y., & Ji, B. (2023). Development environment of China's e-sports industry. In *Electronic sports industry in China: An overview* (pp. 47-74). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-9288-9_3

[2] Cheng, B., & Bayasgalan, Z. (2024). RESEARCH ON THE DEVELOPMENT PROSPECT OF CONTEMPORARY E-SPORTS EDUCATION. In *ICERI2024 Proceedings* (pp. 1270-1277). IATED. https://doi.org/10.21125/iceri.2024.0391

[3] Liu, F., She, Q., Qiu, J., & Wei, Z. (2024). Theoretical logic and strategy of digital economy driving the high-quality development of China's regional E-sports industry. *Journal of Management and Social Development (4)*, 11-15. https://doi.org/10.62517/jmsd.202412402

[4] Pfoff, J. C., & Lee, N. (2024). Super Smash Bros. Ultimate and E-sports. In *Encyclopedia of Computer Graphics and Games* (pp. 1783-1785). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-23161-2_472

[5] Vasiliev, A. A., & Pechatnova, J. V. (2023). Regulatory models in e-sports. *Legal Issues in the digital Age*, (4), 4-22. https://doi.org/10.17323/2713-2749.2023.4.4.22

[6] Bahrololloomi, F., Klonowski, F., Sauer, S., Horst, R., & Dörner, R. (2023). E-sports player performance metrics for predicting the outcome of league of legends matches considering player roles. *SN Computer Science*, *4*(3), 238. https://doi.org/10.1007/s42979-022-01660-6

[7] Lu, Y., Chen, H., & Yan, H. (2022). E-Sports Competition Analysis Based on Intelligent Analysis System. *Computational Intelligence and Neuroscience*, *2022*(1), 4855550. https://doi.org/10.1155/2022/4855550

[8] Gerken, J., Zhang, H., Garnica Caparrós, M., Gardeweg, L., Memmert, D., & Wunderlich, F. (2026). The issue of sparse networks in sports competitions: can Elo ratings efficiently compare football teams that never play a match?. *Journal of the Operational Research Society*, 1-16. https://doi.org/10.1080/01605682.2025.2612140

[9] Jalovaara, P. (2024). Win probability estimation for strategic decision-making in esports.

https://urn.fi/URN:NBN:fi:aalto-202411217333

[10] Dai, M., Duan, J., Hu, J., Wen, J., & Wang, X. (2022). Variational inference of the drift function for stochastic differential equations driven by Lévy processes. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *32*(6). https://doi.org/10.48550/arXiv.2103.15080

[11] Schurz, H. (2024). A brief review on stability investigations of numerical methods for systems of stochastic differential equations. *Networks & Heterogeneous Media*, *19*(1). https://doi.org/10.3934/nhm.2024016

[12] Yang, C. F. (2025). Uncertainty-Aware QoS Forecasting with BR-LSTM for Esports Networks. *Information*, *16*(12), 1016. https://doi.org/10.3390/info16121016

[13] Liu, G., Luo, Y., Schulte, O., & Poupart, P. (2022). Uncertainty-aware reinforcement learning for risk-sensitive player evaluation in sports game. *Advances in Neural Information Processing Systems*, *35*, 20218-20231. https://doi.org/10.52202/068431-1470

[14] Zhou, J., Petrosian, O., & Gao, H. (2024). Enhancing ecological uncertainty predictions in pollution control games through dynamic Bayesian updating. *Scientific Reports*, *14*(1), 12594. https://doi.org/10.1038/s41598-024-63234-1

[15] Sufian, M. A., Varadarajan, J., Hanumanthu, M., Katneni, L., Jamil, A., Lal, V., & Boomer, J. (2024, June). Optimizing E-Sports Revenue: A Novel Data Driven Approach to Predicting Merchandise Sales Through Data Analytics and Machine Learning. In *Science and Information Conference* (pp. 522-567). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-62269-4_35

[16] Wardaszko, M., & Kriz, W. C. (2025, July). Modelling In-Game Events in Game Scenarios: A Comprehensive Framework. In *International Simulation and Gaming Association Conference* (pp. 63-78). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-20129-4_5

[17] Ötting, M., & Karlis, D. (2023). Football tracking data: a copula-based hidden Markov model for classification of tactics in football. *Annals of Operations Research*, *325*(1), 167-183. https://doi.org/10.1007/s10479-022-04660-0

[18] Kim, J., & Chatterjea, R. (2026). Stochastic Differential Equation Treatment of OPS in Baseball.

[19] Chulajata, K., Wu, S., Scalzo, F., & Cha, E. S. (2024). Predicting outcomes in video games with long short term memory networks. *arXiv preprint arXiv:2402.15923*. https://doi.org/10.48550/arXiv.2402.15923

[20] Xu, H., Lin, B., & Liu, L. (2025). Sports event data analysis and win rate prediction model using self-attention mechanism and Transformer. *Journal of Computational Methods in Sciences and Engineering*, 14727978251348637. https://doi.org/10.1177/14727978251348637

[21] Kovačević, M. A., Pešović, M. D., Petrović, Z. Z., & Pucanović, Z. S. (2024). Predictive analytics of in-game transactions: tokenized player history and self-attention techniques. *IEEE Access*, *12*, 149263-149271. https://doi.org/10.1109/access.2024.3477624

[22] Montrucchio, M., Barbierato, E., & Gatti, A. (2026). Uncertainty-Aware Machine Learning for NBA Forecasting in Digital Betting Markets. *Information*, *17*(1), 56. https://doi.org/10.3390/info17010056

[23] Liu, J., Zou, C., & Chintagunta, P. K. (2026). Bayesian learning and skill accumulation in video

game play. *Quantitative Marketing and Economics*, *24*(1), 5. https://doi.org/10.1007/s11129-026-09309-x

[24] Dong, S., Xi, G., & Li, B. (2026). Intelligent Games for UAV Systems: A Survey of Game-Theoretic and AI-Enabled Methods. *Journal of Artificial Intelligence and Technology*. https://doi.org/10.37965/jait.2026.1237

[25] Aslanimoghanloo, M., ElGazzar, A., & van Gerven, M. (2025). Generative Modeling of Clinical Time Series via Latent Stochastic Differential Equations. *arXiv preprint arXiv:2511.16427*. https://doi.org/10.1016/j.jbi.2026.105043

[26] Di Persio, L., Alruqimi, M., & Garbelli, M. (2024). Stochastic approaches to energy markets: from stochastic differential equations to mean field games and neural network modeling. *Energies*, *17*(23), 6106. https://doi.org/10.3390/en17236106

[27] Oh, Y., Kam, S., Lee, J., Lim, D. Y., Kim, S., & Bui, A. (2025). Comprehensive review of neural differential equations for time series analysis. *arXiv preprint arXiv:2502.09885*. https://doi.org/10.24963/ijcai.2025/1179

[28] Eggen, S., Espe, T. J., Grude, K., Risstad, M., & Sandberg, R. (2026). Financial time series uncertainty: A review of probabilistic AI applications. *Journal of Economic Surveys*, *40*(2), 915-953. https://doi.org/10.1111/joes.70018

[29] Jiang, S., Zhang, L., Xu, H., Huang, J., He, Q., Zhou, X., ... & Jiang, J. (2024, October). GameTrail: Probabilistic Lifecycle Process Model for Deep Game Understanding. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (pp. 994-1003). https://doi.org/10.1145/3627673.3679736

[30] Vardakis, M., Margetis, G., Chatzakis, I., Apostolakis, K. C., & Stephanidis, C. (2026). Prediction of MOBA game events based on In-Game Data. *Entertainment Computing*, 101091. https://doi.org/10.1016/j.entcom.2026.101091

[31] Ghasemifard, A. (2025). Milstein scheme for the numerical solution of first-order uncertain stochastic differential equations in stock price simulation. *Caspian Journal of Mathematical Sciences*, *14*(2), 433. https://doi.org/10.22080/cjms.2025.29406.1763

[32] Zhang, Q. (2026). Research on Multimodal Prediction Model of Sports Event Ticket Revenue Based on Transformer. *International Scientific Technical and Economic Research*, *4*(2), 124-147. https://doi.org/10.71451/ISTAER2618

[33] Zhou, H. (2026). Research on the Fusion Model of DeepFM and XGBoost for Digital Consumer Behavior Prediction. *International Scientific Technical and Economic Research*, *4*(2), 98-123. https://doi.org/10.71451/ISTAER2617

[34] Qin, Y. (2026). Research on Long Sequence Learning Behavior Modeling Based on Transformer-XL. *International Scientific Technical and Economic Research*, *4*(2), 1-20. https://doi.org/10.71451/ISTAER2613

[35] Xu, H. (2026). A Deep Reinforcement Learning Signal Control Algorithm for Traffic Carbon Emission Optimization. *International Scientific Technical and Economic Research*, *4*(1), 200–221. https://doi.org/10.71451/ISTAER2610

[36] Chen, G. (2026). Research on algorithm improvement of ARIMA-LSTM hybrid model in time series prediction of inflation rate. *International Scientific Technical and Economic Research*, *4*(1), 90-

122. https://doi.org/10.71451/ISTAER2605
[37] Ding, Y., & Chen, G. (2026). Joint Prediction Model of Reservoir Parameters Based on Multimodal Transformer Graph Neural Operator Physical Constraint Network. *International Scientific Technical and Economic Research*, *4*(1), 70-89. https://doi.org/10.71451/ISTAER2604
[38] Zeng, X. (2026). Cross-Border Trade Fraud Detection via Integrated Heterogeneous Graph Neural Network and XGBoost. *International Scientific Technical and Economic Research*, *4*(1),47-69. https://doi.org/10.71451/ISTAER2603
[39] Wang, W., Shen, S., & Wang, Y. (2026). Forecasting Short-Term Export Volumes with Hybrid Models Integrating SARIMA with Attention-Based LSTM. *International Scientific Technical and Economic Research*, *4*(1),1-22. https://doi.org/10.71451/ISTAER2601

APA Citation

Ma, S. (2026). Modeling of the Randomness in the Process of E-sports Competitions Based on Neural Stochastic Differential Equations. *Journal of Digital Frontier*, *1*(1), 1–30. **https://doi.org/10.67541/jdf2601**