ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Exploration of deep deterministic policy gradient algorithm in high-frequency trading strategy in power market

Yunpeng Feng[1,*]

[1] School of Automation Science and Engineering, South China University of Technology, Guangzhou, 510000, Guangdong, China

SUMMARY: *This paper focuses on the strategy generation task in high-frequency trading in the electricity market, and explores the application of deep deterministic policy gradient algorithm in continuous bidding decision-making. Based on the information of node marginal electricity price, load deviation, renewable output and offer depth, a transaction dataset consisting of 168,000 market time slices, 42 price variables, six bidding features and 18,400 settlement records is constructed. Through time series feature encoding, market interaction environment modeling and Actor-Critic structure, the continuous bidding actions under revenue and risk constraints are jointly learned. Compared with DQN, PPO and rule-based methods, the proposed framework achieves 18.7% cumulative return rate, 1.64 Sharpe ratio and 9.8% maximum backoff, and the average decision delay remains at 7.8 ms. The results show that DDPG can provide more stable strategy learning ability, more accurate continuous bidding control effect, higher computational execution efficiency and online deployment ability for the power market trading system.*

KEYWORDS: *DDPG; Electricity market; High-frequency trading; Strategy learning*

1 Introduction

1.1 Association between high-frequency trading and deep reinforcement learning in the electricity market

High-frequency trading in electricity market is a continuous decision-making activity formed on the basis of short-term price fluctuations, load changes and matchmaking feedback. The core of high-frequency trading is not a single quote itself, but the rapid recognition of time series state, action generation and profit response calculation. Deep reinforcement learning organizes environment perception, policy search and reward estimation in a unified computing link, so it can form a direct correspondence with high-frequency trading in the power market. Taghizadeh et al. studied the bidding strategy in the transacted energy market, and proposed a deep reinforcement learning assisted bidding method, so that agents could modify the clearing behavior according to market feedback [1]. Ochoa et al. studied the multi-time scale bidding process in the day-ahead and real-time markets of hybrid power plants, and proposed a multi-agent deep reinforcement learning framework to coordinate the bidding decisions under different time granularities [2]. Xu et al. studied the procurement and pricing process of electricity sales side, and proposed a deep reinforcement learning method combined with long short-term memory network to make the price prediction and transaction

*f18801388550@163.com
https://doi.org/10.65102/is2026549
Published: 30 April, 2026

decision maintain temporal consistency [3]. Xu et al. further proposed a deep learning decision-making mechanism for retail electricity price and procurement collaboration, so that continuous market signals could be mapped into iterative trading actions [4]. Zhang et al. studied the retail pricing behavior considering user flexibility, and proposed a strategy generation method combining reinforcement learning and imitation learning, so that the pricing process had both the characteristics of revenue-oriented and behavioral constraints [5]. From the perspective of computer, high-frequency trading in power market can be represented as a sequence of decision-making tasks consisting of states, actions, rewards and transition probabilities. Price curve, transaction depth, load offset, renewable output and node marginal signals are encoded into the strategy network, and then the value evaluation unit jointly describes the profit and risk. Therefore, deep reinforcement learning not only provides the algorithmic basis for adapting to the continuous bidding space, but also establishes the computational support for the real-time deployment, online update and stable execution of high-frequency trading systems. This association makes the trading strategy no longer stay in static rule matching, but transformed into a calculation process oriented to stream data processing, temporal representation learning and continuous control output, which is more consistent with the direction of Informatica algorithm modeling, information processing and system implementation.

1.2 Application design of deep deterministic policy gradient algorithm in high-frequency trading of power market

The design of the application of deep deterministic policy gradient algorithm in high-frequency trading in power market is to embed the continuous action reinforcement learning method into the power market matchmaking link, so that the quote generation, risk constraint and profit feedback are completed within a unified computing framework. The design takes the market state flow as input, the bidding action vector as output, and realizes the online calculation of high-frequency trading decision through time series feature coding, strategy network inference and value evaluation unit linkage. Different from traditional methods that rely on static rules and offline thresholds, Deep Deterministic Policy Gradient (DDPG) can directly learn the mapping relationship between declared power, price offset and adjustment amplitude in continuous quotation space. Thus it can adapt to the trading environment in which node price, load disturbance, renewable output and transaction depth change synchronously. Wang et al. studied the scheduling behavior of a virtual power plant with electric vehicles, and proposed a policy generation method based on deep reinforcement learning to make the dynamic control of multi-source energy units have stronger real-time response ability [6]. Han et al. studied the virtual bidding behavior in the electricity market and proposed a machine learning analysis framework to provide data support for trading signal recognition and bidding decision modeling [7]. Li et al. studied the virtual bidding mechanism driven by machine learning, and proposed a modeling method oriented to market efficiency analysis, so as to form a clearer computational correlation between bidding behavior and market feedback [8]. Based on this, this paper designs the HFT application as a closed-loop structure consisting of data access, state construction, strategy output, market feedback and parameter update, whose organization is shown in Fig. 1.

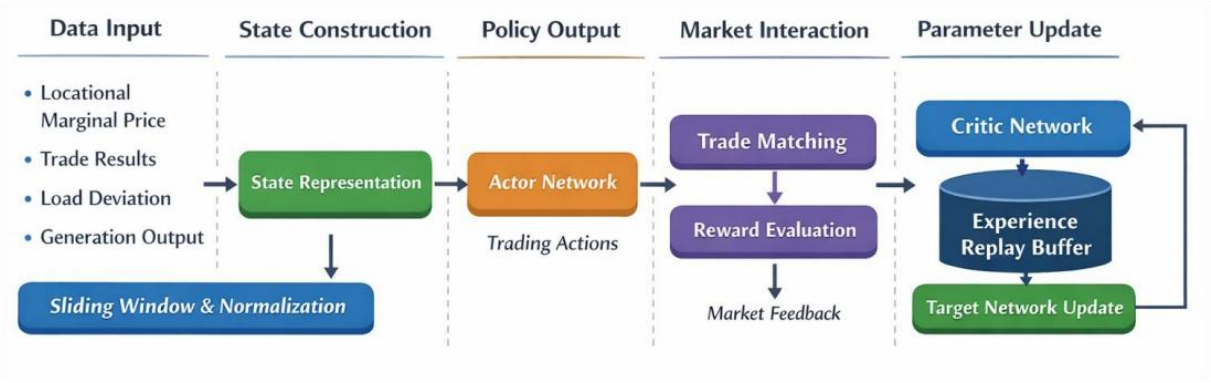

Figure 1: Design process of DDPG application in high-frequency trading in electricity market

The figure illustrates the complete link of the HFT strategy from market data input to strategy parameter writeback. The front-end receives stream data such as marginal electricity price, transaction results, load deviation and unit output of nodes, and forms strategy input after sliding time window segmentation, feature normalization and state splicing. Then the Actor network outputs continuous bidding actions, and obtains transaction feedback through market matching, and the return value is generated by the income and risk joint calculation unit. The Critic network completed the value evaluation according to the reward, participated in the soft update of the target network after the experience sample entered the replay cache, and finally wrote the corrected parameters back to the policy network.

In this process, data flow, decision flow and feedback flow are organized as an iterable computation graph, so that trading actions are no longer dependent on manual threshold switching, but are jointly driven by state representation, value estimation and continuous control. Wang et al. studied residential peer-to-peer energy trading and load scheduling, and proposed a deep reinforcement learning trading framework to make distributed trading actions have continuous decision-making characteristics [9]. Harder et al. studied the real modeling process of wholesale electricity market, and proposed a multi-agent deep reinforcement learning method, so that the market environment, agent behavior and price evolution could be co-expressed in a unified simulation platform [10]. Based on the above research path, the application focus of DDPG in high-frequency trading has shifted to the construction of computation graph, state compression representation, continuous action control and rapid inference deployment, and it provides a clear algorithm interface, data path and structure foundation for the implementation of software systems for electricity market.

2 Related work

The research in this part mainly focuses on reinforcement learning bidding, virtual power plant cooperative control, electricity price prediction and market simulation. Existing results have shown that the power market trading behavior can be gradually transformed from static rules to a computational process combining data-driven and strategy learning. However, there are still obvious differences in the expression of action space, the organization of market feedback, the price time series modeling and the efficiency of online deployment.

Ren et al. studied the double-layer bidding behavior of gas units in the coupled market of electricity and natural gas, and proposed a hierarchical strategic bidding model based on reinforcement learning to make the bidding strategy under multi-market constraints have stronger dynamic coordination ability [11]. Feng et al. studied the collaborative control process of multiple virtual power plants with electric vehicles, and proposed a robust

federated deep reinforcement learning method, so that distributed agents can complete collaborative optimization under the condition of parameter sharing [12]. Liu et al. studied the optimal bidding behavior in the real-time multi-participant electricity market, and proposed a bidding strategy generation method based on deep reinforcement learning, so that short-term load disturbances could directly enter the strategy update link [13]. Wu et al. studied the strategic bidding behavior in the competitive electricity market, and proposed a modeling method that integrated multi-agent simulation and deep reinforcement learning, so that the interaction relationship between market entities could be more completely expressed in the process of strategy learning [14].

In order to more clearly compare the technical focus of the existing research, the main conclusions and the connection relationship with the work in this paper, the relevant results are arranged in Table 1.

Table 1: Overview of related research on HFT in electricity markets

References	Research Content	Method Characteristics	Main Conclusion	Relation to This Study
[11]	Bi-level strategic bidding in integrated energy markets	Reinforcement learning-based hierarchical decision-making	Enhances multi-market coordination capability	Indicates that continuous bidding is suitable for strategy learning
[12]	Coordinated control of multiple virtual power plants	Federated deep reinforcement learning	Improves distributed robustness	Provides a reference for multi-agent state aggregation
[13]	Real-time optimal bidding for multiple participants	Deep reinforcement learning	Improves adaptability in real-time bidding	Supports the modeling of high-frequency trading strategies
[14]	Strategic bidding in competitive markets	Multi-agent simulation and deep reinforcement learning	Strengthens the expression of agent interactions	Helps construct market environments
[15]–[24]	Electricity price forecasting and market signal computation	Machine learning, LSTM, Transformer, and related methods	Improves the accuracy of price modeling	Provides a basis for state input and reward design

In terms of price prediction and market signal calculation, Tschora et al. studied the prediction process of day-ahead market electricity price, and proposed a prediction framework based on machine learning to stably depict the statistical changes of day-ahead price series [15]. Das et al. studied the prediction of node price spread between day-ahead and real-time markets, and proposed a modeling method combining long short-term memory network and sequence-to-sequence structure to make the cross-market price difference have a learnable representation [16]. Meng et al. studied the electricity price prediction process under the condition of high proportion of renewable energy access, and proposed an LSTM network based on attention mechanism to make the price sequence with strong volatility get a more detailed time series representation [17]. Khojasteh et al. studied the aggregate bidding process

of energy storage and wind power resources in the joint energy and reserve market, and proposed a robust aggregate bidding model to make uncertain resources have stronger risk tolerance ability when participating in transactions [18].

Jiang et al. studied multivariate short-term electricity price forecasting, and proposed a modeling scheme combining artificial intelligence and multi-input multi-output to enable heterogeneous market variables to enter a unified forecasting structure [19]. Krishna Prakash et al. studied the hybrid deep learning network in electricity price prediction, and proposed a multi-model combined prediction method, so that the local pattern and the overall trend of price change could be learned synchronously [20]. Cantillo-Luna et al. studied the forecasting process of intraday electricity price and proposed the probabilistic Transformer neural network structure, so that the short-term price distribution and fluctuation range can be output at the same time [21]. Sai et al. studied wholesale electricity price and frequency modulation price prediction under event-driven conditions, and proposed a joint prediction framework based on machine learning algorithm to form a direct mapping between market events and price response [22].

Jiang et al. studied the quantile regression average electricity price prediction method with non-convex regularization term, and proposed a price estimation mechanism for uncertain distribution, which made the prediction results have finer interval expression ability [23]. Yang et al. studied the day-ahead electricity price prediction process with spatial correlation, and proposed a prediction method considering spatial dependence, so as to make full use of the price linkage characteristics between regional nodes [24].

Compared with the above work, the existing research has completed the method expansion from price prediction to strategy learning and from single-agent decision-making to multi-agent simulation. However, there is still room for further deepening the unified modeling of continuous bidding control, compressed representation of time series state and joint feedback of profit and risk in high-frequency trading scenarios. On this basis, this paper will organize the streaming market data, continuous quotation action and value evaluation link into the same DDPG framework, so that the related research will further transition from price prediction support to deployable high-frequency trading strategy calculation. This path is more in line with the writing requirements of information processing, temporal modeling, strategy optimization and system realization in the computer field, and also provides a clear reference for the subsequent experimental design.

3 Research Methods

3.1 Time series data organization and feature construction of high-frequency trading in power market

The goal of the organization and feature construction of high-frequency trading time series data in power market is to convert matching records, node price flow, load deviation, unit output and quotation depth into continuous state inputs that can be called by DDPG. The data access terminal aligned the market matching log, the quotation table, the node marginal price sequence, the real-time load curve and the output record according to the unified time benchmark, retained the 5 min granularity time slice, and established a sequential cache structure. The price, volume, quote queue, congestion marker and load offset were written in each time slice, and then the missing value repair, abnormal truncation and field mapping were completed, so that the original record was transformed from discrete transaction messages to continuous transaction frames. This processing method preserves the fine-grained changes of market fluctuations, and also makes the subsequent state construction and strategy

training based on a unified timeline.

In order to maintain the continuous organization of the transaction sample in the time dimension, this paper uses the fixed window segmentation method to construct the state sequence:

$$S_t = \{x_{t-L+1}, x_{t-L+2}, \cdots, x_t\} \tag{1}$$

where S_t represents the state window formed at time t; x_t is the original observation vector at the current trading moment. Let L denote the window length. This formula is used to organize continuously arriving market records into iterable input units, so that each decision can retain the information of price, transaction and constraint change in the latest period of time.

Fig. 2 shows the data organization process.The data flow in the figure consists of six parts: market log input, time alignment, exception cleaning, window segmentation, feature splicing and state cache. Node electricity price, trading volume, quote queue, load offset and renewable output first enter the unified time axis, and then are patched and normalized to write into the sliding window, and then the state tensor is generated by the cache module and written into the experience replay area. In this way, state construction, action generation and feedback writing back share the same temporal basis, and also provide a stable data entry for subsequent continuous bidding decisions.

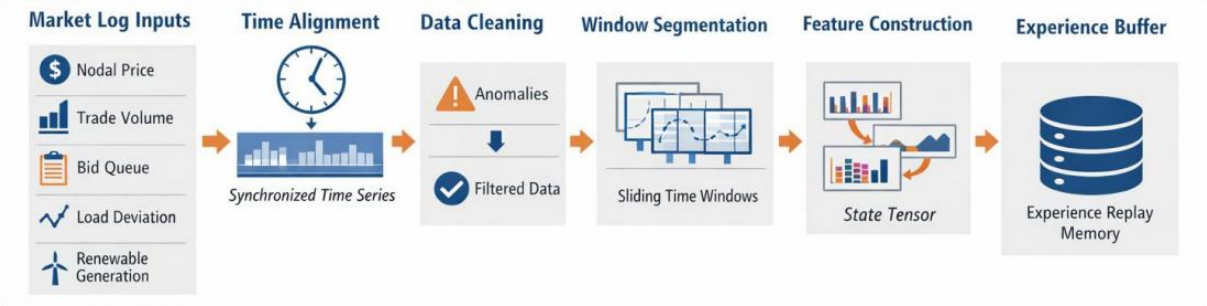

Figure 2: time series data organization and feature construction process of high-frequency trading in electricity market

In the feature construction stage, this paper divides the input variables into four subsets: price class, transaction class, constraint class and volatility class. Since different subsets differ in dimension, sampling frequency and fluctuation intensity, unified mapping and fusion compression are required before entering the policy network:

$$z_t = \sigma(W_p p_t + W_v v_t + W_c c_t + W_r r_t + b) \tag{2}$$

where p_t represents the price subvector; v_t represents the transaction subvector; c_t represents the constraint subvector; r_t represents the fluctuation sub-vector; W_p, W_v, W_c W_r denote the corresponding weight matrix; b represents the bias term; Let $\sigma(\cdot)$ denote the nonlinear mapping function. The function of this formula is to compress multi-source market variables into a unified representation space, so that heterogeneous data can enter the subsequent strategy learning link under the condition of consistent dimensions.

Considering the synchronous appearance of price shock, transaction jump and boundary constraints in high-frequency trading scenarios, this paper further introduces the time decay term, transaction feedback term and constraint mask term to enhance the input state:

$$h_t = \sum_{i=0}^{L-1} \alpha_i z_{t-i} + \beta \Delta q_t + \gamma \Delta m_t + \eta (z_t \odot u_t) \quad (3)$$

where, h_t represents the enhanced state vector; Let α_i denote the attenuation coefficient at the i historical moment inside the window. Δq_t represents the volume increment; Δm_t represents the node price offset; β, γ, η denote the adjustment coefficient; $\odot$ indicates element-wise multiplication. The formula writes historical trend, market feedback and constraint information into a unified input, so that DDPG can simultaneously perceive profit opportunities, transaction response and transaction boundaries when generating continuous quotation actions.

After completing the above processing, the original market records are reorganized into continuous state sequences, fused feature vectors, and replayable sample units. The data structure thus formed is suitable for both batch training in GPU environment and fast inference in online trading scenarios. Time series data organization and feature construction is not a simple process of data collation, but a key link in transforming market behavior into computable state representation. After the processing of this link, electricity price fluctuation, transaction change and constraint disturbance can enter the subsequent market interactive environment modeling and strategy learning process in a unified form, so as to provide a stable, reusable and engineering implementation conditions for the generation of high-frequency trading strategies.

3.2 Modeling of market interaction environment for high-frequency Trading strategy learning

The market interactive environment of high-frequency trading strategy learning in power market does not simply write the matching results into the sample cache, but organizes the bidding behavior, transaction feedback, price jump and risk constraint into a continuous decision-making closed loop. When modeling, the market environment advances according to a fixed time step, and each time step receives the previous round state, quotation action and post-match feedback at the same time, and completes the income settlement, inventory update and constraint verification in the trading engine. In this way, the market environment not only retains the temporal continuity of power trading, but also has the state transition characteristics required by reinforcement learning. For high-frequency scenarios, node electricity price, transaction depth, load deviation, renewable output and quotation queue position all change in a short time. Therefore, environmental modeling must map these signals to a unified interaction space, so that the strategy network can perceive the fluctuation and the synchronous change of global revenue in the process of continuous bidding.

In order to make the state transition computable, this paper sets up four interaction layers in the environment: price channel, transaction channel, constraint channel and position channel. The price channel is responsible for writing the node's marginal electricity price, day-ahead price spread and real-time offset. The trading channel records the declared volume, the actual trading volume and the unsettled residual volume; The constraint channel writes the upper and lower bounds of the quotation, the frequency limit of the transaction and the capacity boundary. The open position channel maintains the net trading position and return accumulation in consecutive periods. After finishing splicing at the same time slice, each channel enters the environment state storage area, and then triggers the state update together with the action result of the previous moment. The state update rules are as follows:

$$y_{t+1} = Ay_t + Ba_t + Cg_{t+1} + D\xi_{t+1} \tag{4}$$

Here,y_{t+1} represents the state of the environment at the next time. y_t represents the current state. a_t represents the continuous offer action given by the policy network; g_{t+1} denotes the vector of market observations at the next time. Let ξ_{t+1} denote the disturbance term generated after matching. A, B, C and D denote the corresponding mapping matrix. This formula is used to write the state history, current action, market input and disturbance feedback into the transition process, so that the environment update has the dynamic expression ability under the condition of continuous bidding.

Under this rule, the environmental return value is no longer represented only by the single transaction return, but the price offset, execution bias and risk penalty are synchronously incorporated into the return calculation. In order to avoid excessive aggressive behavior in local fluctuations, this paper adds a transaction slip point term and inventory pressure term to the trading return, so that the reward function can not only characterize the return, but also constrain the stationarity of the action. The form of the combined return is as follows:

$$r_t = m_t(\pi_t^{clr} - \pi_t^{bid}) - \lambda_1|a_t - m_t| - \lambda_2 \max(0, |e_{t+1}| - \bar{e}) - \lambda_3 Var(\pi_{t-w:t}) \tag{5}$$

Here, r_t represents the comprehensive return at the current moment. m_t represents the actual volume; Let π_t^{clr} denote the market clearing price; π_t^{bid} denotes the declared price; e_{t+1} represents the net position offset at the next time instant; $\bar{e}$ denotes the upper bound of allowed inventory; From λ_1 to λ_3 denote the penalty coefficients; $Var(\pi_{t-w:t})$ represents the price variance within the window. This equation unifies revenue, execution error, inventory pressure and price fluctuation into the reward link, so that DDPG can balance profitability and action stability in the training process.

The structure of the environment is shown in Fig. 3. This figure illustrates the market interaction link in HFT strategy learning. Firstly, the market flow data enters the state storage area, and then the continuous bidding action is output by the strategy network. According to the action, the transaction matching module completes the transaction judgment with the market queue, and then the profit, deviation and risk feedback are sent to the reward calculation unit. The whole process is driven by a uniform time step, which ensures that the interaction results can be directly written into the experience replay buffer.

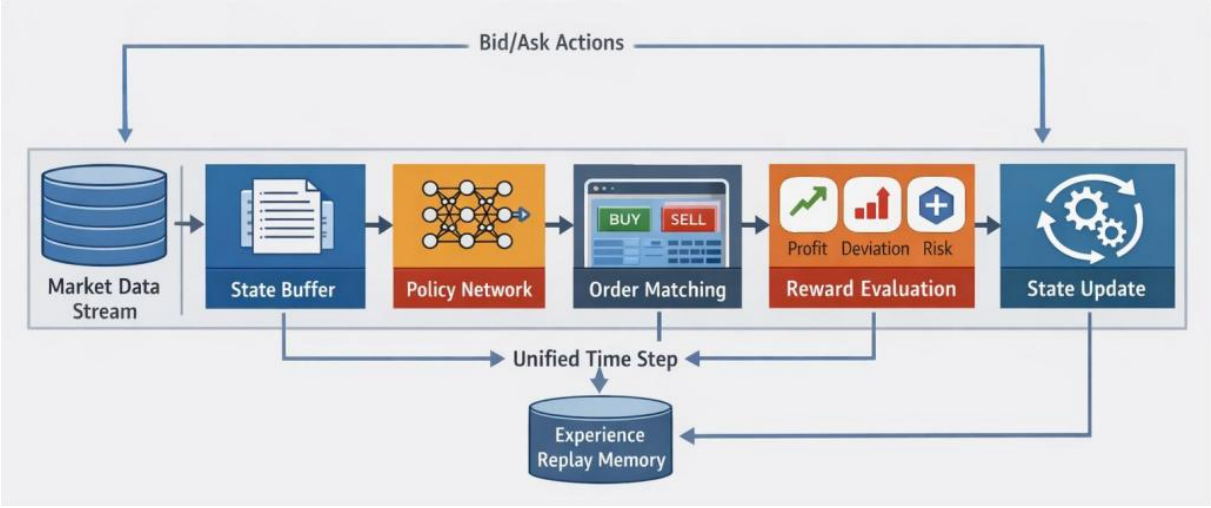

Figure 3: Structure of the market interaction environment for HFT strategy learning

Different channels do not play the same role in the environment. To illustrate the computational responsibilities of the components, the key variable configurations are shown in Table 2.

Table 2: Configuration of key variables in the market interaction environment

Variable Category	Constituent Content	Computational Role	Writing Location
Price Channel	Nodal electricity price, day-ahead price spread, real-time deviation	Describes price states and fluctuation directions	State register area
Transaction Channel	Bidding quantity, traded quantity, residual quantity	Reflects execution results and transaction intensity	Transaction buffer area
Constraint Channel	Upper and lower bidding bounds, frequency limits, capacity boundaries	Controls the legal range of actions	Constraint validation area
Position Channel	Net position, cumulative profit, inventory deviation	Maintains cross-period profit continuity	Return calculation area

The setting of the table ensures that there is a strict mapping relationship between action output, benefit feedback and state transition, and also enables the environment to adapt to both offline training and online inference. After the modeling is completed, the market interaction process no longer relies on discrete rule splicing, but is transformed into a continuous calculation process driven by state perception, action execution, reward feedback and constraint correction. Such an environment representation is more suitable for DDPG to deal with continuous bidding space, and also provides a stable operation foundation for subsequent policy input mapping and network training target design. At the same time, the matching results in the environment do not directly cover the original state, but first pass timestamp verification and outlier filtering, and then write to the buffer to ensure the sequential consistency and traceability of continuous samples in high-frequency scenarios. For the electricity market, this modeling method can more truly reflect the chain relationship among quote response, transaction delay and revenue writeback, and take into account the efficiency of online deployment.

3.3 Design of temporal feature coding and strategy input mapping for high-frequency trading decision making

The core of the design of temporal feature encoding and strategy input mapping for high-frequency trading decisions is to transform the continuously arriving market records into a compact representation that can be directly invoked by the strategy network. In the high-frequency trading scenario, node electricity price, trading volume, declaration queue position, net position deviation and renewable output change fluctuate together in a short period of time. If only the original vector is directly input into the network, the timing dependence and local shock information are easy to be diluted. Therefore, this paper sets up the timing coding layer, attention aggregation layer, gated fusion layer and strategy mapping layer after the state construction, so that the original transaction flow first completes the local pattern extraction before entering the action generation link. The sequence encoding is computed as follows:

$$e_t = \phi\left(\sum_{i=0}^{k-1} K_i\, x_{t-i} + b_e\right) \tag{6}$$

where e_t denotes the local temporal coding result at time t; K_i represents the ith convolution kernel parameter; x_{t-i} denotes the historical input vector; b_e represents the bias term; Let $\phi(\cdot)$ denote the nonlinear mapping. This formula is used to extract short time patterns such as price jump, transaction surge and constraint switching.

In order to preserve the importance difference of different time slices within the window, this paper further introduces the attention aggregation mechanism, and the attention weight is calculated as follows:

$$\alpha_{t,j} = \frac{\exp\left(q_t k_j^{\top}/\sqrt{d_k}\right)}{\sum_{m=t-L+1}^{t} \exp\left(q_t k_m^{\top}/\sqrt{d_k}\right)} \tag{7}$$

Here, $\alpha_{t,j}$ denotes the weight assigned to the j historical coding fragment at time t. q_t represents the query vector. k_j is the key vector; d_k denotes the scaling dimension. This formula is used to measure the intensity of attention to each historical segment at the current decision moment.

After obtaining the attention weights, the history encodings are aggregated into context vectors, which are expressed as follows.

$$c_t = \sum_{j=t-L+1}^{t} \alpha_{t,j}\, e_j \tag{8}$$

where c_t represents the context vector; Let e_j denote the j encoding result. This formula compresses the multi-moment information into the context representation that is most relevant to the current decision.

Considering that the market state still contains non-stationary components such as transaction feedback and inventory boundary, this paper uses the gated fusion structure to complete the deep combination, and the fusion form is as follows:

$$g_t = \sigma(W_g[c_t; y_t] + b_g) \odot \tanh(W_h[c_t; y_t] + b_h) \tag{9}$$

Here, g_t represents the gating output; W_g and W_h represent mapping matrices; b_g and b_h represent the bias terms; $[c_t; y_t]$ represents the concatenation result of the context vector and the environment state. Let $\sigma(\cdot)$ denote the gating function; $\odot$ indicates element-wise multiplication. This formula is used to control the proportion of context information injected into the current environment information.

After fusion, the policy input mapping layer projects the encoding result to the action space prior representation, which is calculated as follows.

$$u_t = \rho(W_u g_t + U_p p_t + U_n n_t + b_u) \tag{10}$$

where u_t represents the policy network input; W_u, U_p, and U_n denote the projection parameters; b_u denotes the bias term; Let $\rho(\cdot)$ denote the mapping function; p_t denotes the price prior; n_t denotes the net position correction term. This formula writes coded

features, price trends and position constraints into the policy input together.

After the above processing, the original transaction sequence no longer enters the DDPG in the form of loose observations, but is transformed into a hierarchical representation with both local patterns, global dependencies and constrained responses. The input formed in this way can not only improve the recognition accuracy of the Actor network for consecutive offer actions, but also enhance the estimation stability of the Critic network for reward changes. Temporal feature encoding and strategy input mapping thus become an important link connecting market data and high-frequency trading decisions, and provide a clear structure, consistent calculation and easy to deploy input basis for subsequent network training target design. At the same time, the mapping method preserves the timestamp order, transaction feedback direction and quotation boundary information, so that the online inference phase can complete the input reorganization at the millisecond level, and maintain the data structure of the training end and the deployment end. It has direct engineering support value for the realization of power market software system.

4 Analysis Techniques

4.1 DDPG network structure and training objective design for continuous bidding decision

In the continuous bidding scenario, the bidding action is not a discrete label selection, but a continuous control output composed of declared power, price offset, inventory adjustment and risk mitigation. In order to make the high-frequency trading strategy complete stable iteration in millisecond market feedback, this paper designs DDPG as a dual-channel structure composed of timing input layer, feature compression layer, Actor decision-making layer, Critic evaluation layer and target network layer. The temporal input layer receives the state vector formed in the previous section, the feature compression layer completes the shared representation extraction, the Actor is responsible for generating continuous offer actions, the Critic is responsible for estimating the value of the action in the current market state, and the target network provides a smooth reference for parameter updating. Such a network layout not only retains the expression advantage of deep reinforcement learning in continuous action space, but also weakens the drastic disturbance of gradient update caused by price jump in high-frequency trading scenarios.

The action output of the policy network is of the following form:

$$a_t = \Gamma(W_a n_t + b_a), \quad \Gamma(\cdot) = l + \frac{u - l}{2}(1 + \tanh(\cdot)) \tag{11}$$

where a_t denotes the continuous offer action at time t; n_t denotes the policy input representation obtained in the previous section; W_a and b_a represent the parameters of the Actor output layer; Let $\Gamma(\cdot)$ denote the interval scaling function; l and u denote the action lower bound and upper bound, respectively. This formula is used to project the state representation directly into the continuous bidding space, so that the action generation is consistent with the market time series state.

In order to ensure that the action range conforms to the transaction boundary, this paper adds a scaling mapping to the output, so that the declared price and the declared quantity simultaneously meet the constraints of the capacity ceiling, the quoted bandwidth and the inventory margin. The value network uses the state-action joint input method to evaluate the profit potential of the current bidding action. The form of value estimation is as follows:

$$q_t = Q(y_t, a_t; \theta^Q) \tag{12}$$

Here, q_t represents the value estimate given by the Critic network; y_t represents the state of the environment. Let θ^Q denote the network parameters of the Critic. This formula represents the profit judgment result of the value branch for the current state action pair.

Since the single step return fluctuates greatly in the high-frequency market, the local noise is easy to be amplified if the immediate return is directly used to drive the training, so this paper uses the target network to construct a smooth supervision signal. The goal value is calculated as follows:

$$\hat{q}_t = r_t + \gamma Q'(y_{t+1}, \mu'(y_{t+1}; \theta^{\mu'}); \theta^{Q'}) \tag{13}$$

where $\hat{q}_t$ represents the target value; r_t represents the current reward. Let γ denote the discount factor; $Q'(\cdot)$ and $\mu'(\cdot)$ denote the target Critic and target Actor, respectively. This formula combines the immediate payoff with the discount value at the next moment, so that the training objective has temporal continuity and forward-looking.

According to this target value, the parameter update of the value network is driven by the mean square error loss. The loss function is as follows:

$$L_Q = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{q}_i - Q(y_i, a_i; \theta^Q)\right)^2 \tag{14}$$

Here, L_Q represents the network loss of Critic; N represents the number of mini-batch samples; $\hat{q}_i$ represents the target value of the i sample; $Q(y_i, a_i; \theta^Q)$ denote the current estimate. This equation is used to constrain the deviation between the estimated value and the supervised target, and ensure that the training direction of the value branch is stable and controllable.

In order to make the policy network complete continuous action modification along the direction of value improvement, the Actor uses deterministic policy gradient update. The gradient is expressed as follows:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q(y, a; \theta^Q)|_{y=y_i, a=\mu(y_i)} \nabla_{\theta^\mu} \mu(y; \theta^\mu)|_{y=y_i} \tag{15}$$

Here, $\nabla_{\theta^\mu} J$ denotes the gradient of the Actor target to the parameter. θ^μ denotes the Actor parameters; $\mu(\cdot)$ denotes the current policy function; Let $\nabla_a Q(\cdot)$ denote the derivative of the value function with respect to the action. This equation indicates that the Actor parameter is updated through the sensitivity of Critic to the action value, so that the continuous offer gradually moves towards higher revenue.

To illustrate the network architecture and the key configurations in the training objective, the relevant Settings are shown in Table 3.

Table 3: DDPG network structure and training target key configuration

Module	Input Content	Output Content	Computational Role
Temporal Input Layer	State window, price deviation, transaction feedback	State tensor	Maintains temporal continuity
Feature Compression Layer	State tensor	Shared representation	Performs dimensionality reduction and unified encoding
Actor Decision Layer	Shared representation	Continuous bidding actions	Outputs price and power decisions
Critic Evaluation Layer	State, action	Value estimation	Evaluates the return of the current action
Target Network Layer	Online network parameters	Smoothed target parameters	Suppresses training oscillation

The table configuration ensures that there is a clear division of labor between action output, value evaluation and goal construction, and also enables the network to maintain a consistent data path and parameter interface during the offline training and online inference phases. Due to the fast update speed of the online network in the training process, the supervision signal will shift sharply if the target network changes synchronously, so we use the soft update method to maintain the smooth evolution of the target parameters. The update rules are as follows:

$$\theta^{Q\prime} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q\prime}, \quad \theta^{\mu\prime} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu\prime} \tag{16}$$

where $\theta^{Q\prime}$ and $\theta^{\mu\prime}$ represent the target network parameters. θ^{Q} and θ^{μ} represent the online network parameters; Let τ denote the soft update coefficient. This formula maintains the smooth change of the target network through a small proportion of parameter transfer, and avoids the jump of the estimation target in the continuous bidding environment.

In this structure, Actor and Critic do not run independently, but form a closed loop under the condition of sharing state representation and experience playback samples. The state batch first enters the shared encoding layer, and then passes to the action branch and the value branch respectively, and the updated parameters are slowly written back through the target network. Designed in this way, the network can both absorb price changes in short time Windows and maintain consistency of returns across multiple time steps. During the whole training process, the experience samples are written into the playback cache in chronological order, and then sent to the GPU for parallel update by random mini-batch sampling. The network forward inference and loss backpropagation both adopt a unified tensor structure. The modeling method enables DDPG to transform electricity price deviation, transaction feedback, inventory pressure and risk constraints into the same learning objective, and form a continuous bidding decision network with clear structure, closed calculation and easy deployment in high-frequency trading scenarios, which provides a stable model foundation for the subsequent analysis of strategy update mechanism.

4.2 Analysis of strategy update mechanism and calculation process in high-frequency trading decision-making process

Strategy update in high-frequency trading decision making is not a static repair of a single quote result, but a synchronous correction of action output, value evaluation and target

parameters under continuous market feedback. When the electricity market enters the state of high-frequency matching, the node's electricity price, load deviation, declaration queue and transaction delay will change the distribution of strategic revenue in a very short time. Therefore, the update mechanism must organize the environmental reward, sample extraction and network iteration into a unified computing link. Based on this idea, this paper divides the update process of DDPG into five stages: experience writing, batch sampling, value return, policy modification and goal smoothing, and maintains the consistent structure between online decision making and offline training through tensorization calculation.

The experience replay buffer receives the current state, action, reward and the next state, and then writes it to the index according to the timestamp and the completion mark. In order to avoid the strong correlation between adjacent samples in the high fluctuation range, the cache module introduces a segmented random sampling mechanism while preserving the integrity of the time series, so that the trading samples under different price ranges and load levels participate in the training together. After the batch samples are formed, the Critic network calculates the estimated value according to the current parameters, and then returns the error with the target value, and the Actor network modifies the action direction according to the value gradient. Fig. 4 gives the organization of the whole update process.

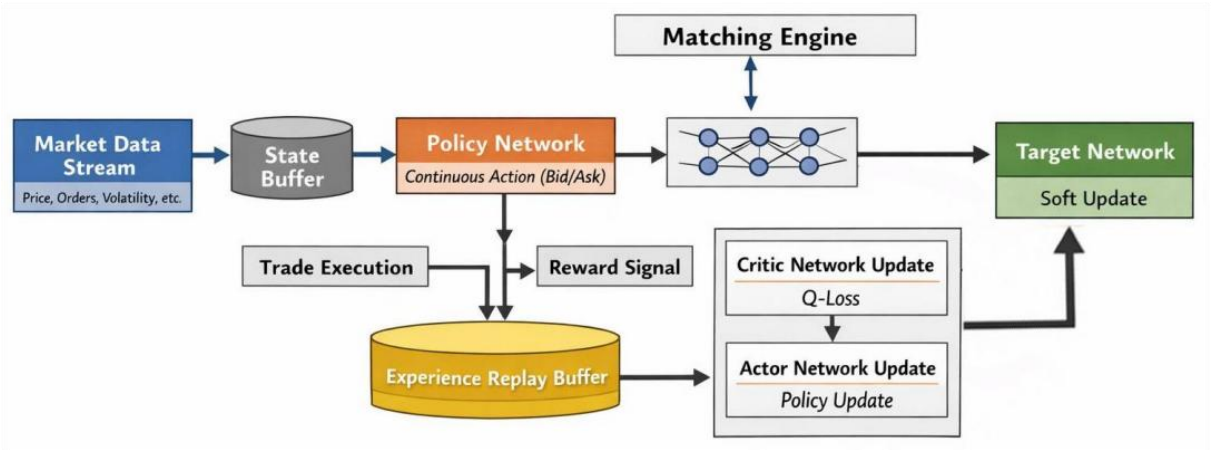

Figure 4: Strategy update and calculation process in HFT decision making

The figure shows that the market flow data first enters the state buffer, the strategy network outputs continuous quotation actions, the match engine returns the transaction results and profit signals, and then the experience samples are written into the replay cache. After extracting a small batch of samples from the cache, the training thread first updates the Critic, then updates the Actor, and finally writes the online network parameters to the target network in a soft update manner to form a closed-loop iteration.

The batch sampling link does not read samples in a fixed order, but determines the training priority according to the timing novelty, transaction deviation and risk exposure, so that the segments with high volatility can maintain sufficient weight in a limited batch. The value network update phase aligns the target value with the current estimate by error, and then uses back propagation to correct the parameters, so as to keep the return estimate stable in the continuous offer space. In the update phase of the policy network, the sensitivity of the value function to the action is directly transmitted to the output layer, so that the price direction gradually converges along the revenue improvement path.

In order to obtain stable attention for high-volatility trading clips in batch training, this paper writes timing novelty, value bias and risk exposure into the sampling weight calculation formula as follows:

$$\omega_i = \frac{\left(\nu_i + \kappa_1|\delta_i| + \kappa_2 r_i^{\text{risk}}\right)^{\zeta}}{\sum_{j=1}^{N}\left(\nu_j + \kappa_1|\delta_j| + \kappa_2 r_j^{\text{risk}}\right)^{\zeta}} \tag{17}$$

Here, ω_i represents the sampling weight of the i sample in the current batch; Let ν_i denote the temporal novelty; Let δ_i denote the temporal difference error; r_i^{risk} is the intensity of risk exposure. κ_1 and κ_2 are the regulation coefficients. Let ζ denote the priority compression index. This formula unifies the time series novelty, value bias and risk level into the sampling stage, so that the high volatility trading segments can obtain more stable attention in the training.

In order to reduce the instantaneous pull caused by high-frequency price jump on value estimation, this paper adopts the smooth error backpropagation method to update the value network parameters. The calculation formula is as follows:

$$\theta_Q^{(s+1)} = \theta_Q^{(s)} - \eta_Q \frac{1}{N}\sum_{i=1}^{N} \nabla_{\theta_Q} \mathcal{H}\left(\hat{q}_i - Q\left(y_i, a_i; \theta_Q^{(s)}\right)\right) \tag{18}$$

Here, $\theta_Q^{(s)}$ represents the Critic parameters before the s update round. Let $\theta_Q^{(s+1)}$ denote the updated Critic parameters; η_Q is the learning rate; N is the sample batch size. $\left(y_i, a_i; \theta_Q^{(s)}\right)$ denote the current estimate. $\mathcal{H}(\cdot)$ denotes the Huber-type error function. This formula is used to suppress the violent pulling of abnormal samples on the value branch under the condition of high frequency fluctuation.

In order to ensure the steady output of continuous bidding actions in the process of revenue growth, this paper adds action continuity constraints on the basis of policy gradient, and its correction formula is expressed as follows:

$$\theta_\mu^{(s+1)} = \theta_\mu^{(s)} + \eta_\mu \left[\begin{array}{c} \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q\left(y_i, a; \theta_Q\right)|_{a=\mu(y_i)} \nabla_{\theta_\mu} \mu\left(y_i; \theta_\mu^{(s)}\right) \\ -\lambda \nabla_{\theta_\mu} \|\mu(y_i) - \mu(y_{i-1})\|_2^2 \end{array}\right] \tag{19}$$

Here, $\theta_\mu^{(s)}$ and $\theta_\mu^{(s+1)}$ represent the Actor parameters before and after updating, respectively. η_μ represents the policy learning rate; Let $\nabla_a Q(\cdot)$ denote the sensitivity of the value function to the action; $\mu(\cdot)$ represents the current policy output; Let λ denote the smoothing regularity coefficient. This formula adds an action continuity constraint in addition to the policy gradient, so that the output of continuous bidding is stable while the revenue is improved.

After completing the above steps, the update thread does not immediately cover the target network, but slowly migrates the parameters according to the smooth ratio to reduce the target oscillation in the high-frequency matchmaking environment. The update mechanism thus formed takes into account real-time revenue response, sample diversity and training stability, and also enables the strategy iteration to maintain a consistent calculation rhythm under the conditions of price jump, transaction delay and constraint switching. For high-frequency trading in the electricity market, this mechanism improves the correction accuracy of continuous bidding actions, and also provides a traceable training basis for subsequent strategy performance analysis. At the same time, the update link uses a unified tensor

interface to connect the state cache, experience playback and parameter migration modules, so that the training end and the deployment end maintain the same data structure and call order, which is convenient for GPU parallel execution, online replication and engineering deployment, and makes the system have high online execution stability.

5 Results

5.1 Description of high-frequency trading strategy generation and income risk assessment results

This section explains the results of HFT strategy generation and its return risk performance. The experiment adopts a unified matching environment, and the data set contains 168000 5-minute market time slice, 42 price variables, 6 bidding features and 18400 settlement records. The state input is composed of node marginal price, real-time load deviation, quote depth, renewable output and net position. After complete training, DDPG formed a more stable continuous offer output in the testing phase, with a cumulative return rate of 18.7%, Sharpe ratio of 1.64, and average decision delay of 7.8 ms. Compared with DQN, PPO and rule-based methods, the strategy shows a better balance between revenue expansion and risk control, and the maximum back-off decreases by 13.2%, indicating that the model can not only capture short-term price changes, but also maintain good action stability in the process of continuous bidding.

In order to observe the action distribution and return response of the strategy in different market states, this paper draws the price offset, transaction depth and return per unit time as a two-dimensional heat map, as shown in Fig. 5. The high response area in the heat map is mainly concentrated in the state unit where the positive spread is 0.8%-1.6% and the normalized value of transaction depth is 0.55-0.72. Within this range, the average revenue per unit time is maintained between 0.041-0.048, and the average transaction completion rate is 0.67. In contrast, when the price offset is less than 0.3% or the transaction depth drops below 0.30, the action output shrinks significantly, and the unit time revenue mostly falls below 0.012. The results show that the strategy network does not amplify actions synchronously in all price fluctuation ranges, but adjusts the quotation sensitivity by combining the transaction conditions and the profit space, so as to maintain the coordination relationship between action output and market marketability.

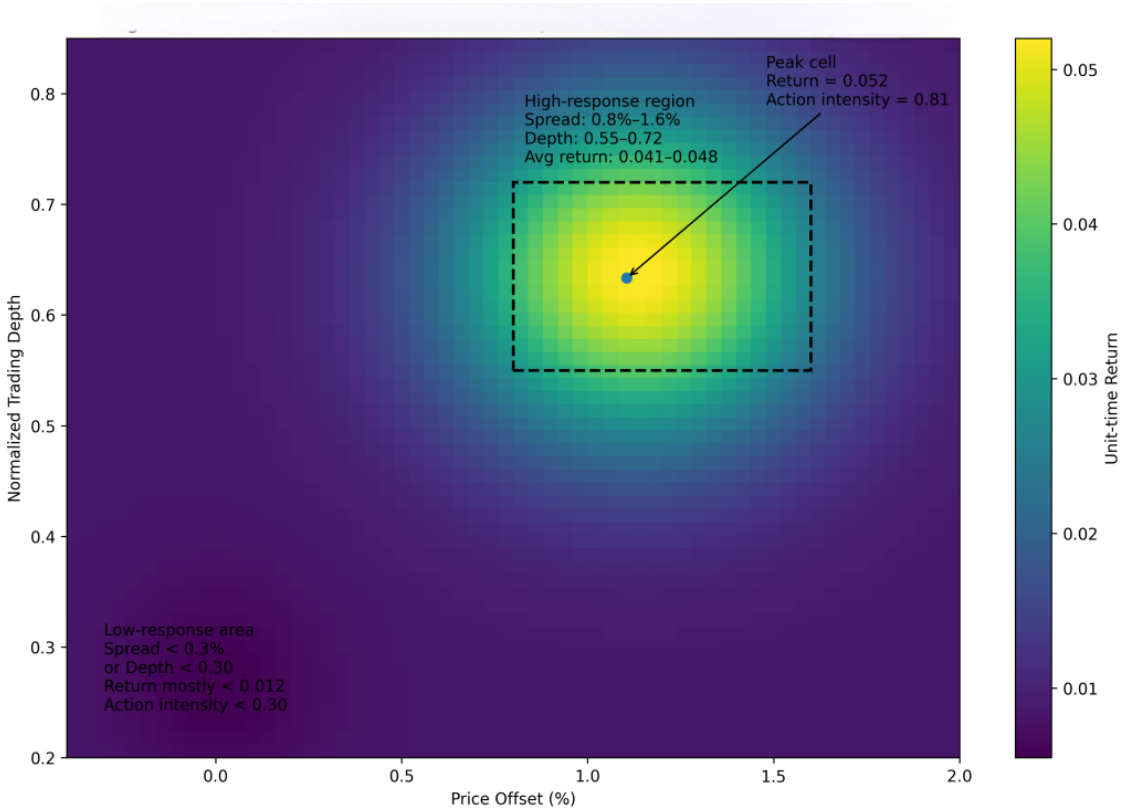

Figure 5: Heat map of DDPG quote action returns under different price offset and transaction depth conditions

Fig. 5 shows that the warm color area is mainly distributed in the position where the middle and high transaction depth and the middle positive spread overlap, the normalized action intensity of the maximum response unit reaches 0.81, and the peak value of revenue per unit time is 0.052. The cool color area is mainly concentrated in the low depth and weak spread signal area, and its action intensity is generally lower than 0.30. Combined with the test log, it can be seen that the average declaration offset of the strategy in the high response area is 1.14%, and the proportion of the actual trading volume in the declaration volume is 0.69, which indicates that the timing feature coding and strategy input mapping constructed in the previous section have been able to write the price signal, transaction feedback and constraint boundary into the action generation process.

From the perspective of risk performance, DDPG did not show obvious return collapse in the continuous trading period. In this paper, box plots are further used to compare the distribution of weekly returns of different methods in the testing phase, as shown in Fig. 6. The results show that the median return of DDPG in 12 test weeks is 0.31, the lower quartile is 0.27, the upper quartile is 0.36, and the overall weekly return remains in the positive range, with the highest weekly return reaching 0.44 and the lowest weekly return reaching 0.18. In contrast, although PPO and DQN also maintained a positive return distribution, the median level and overall interval were lower than DDPG. The rule-based approach is more volatile and individual weekly returns have approached below the zero return benchmark. This result shows that DDPG forms a more stable balance relationship between revenue expansion and risk constraint, and maintains better continuous decision consistency in the process of cross-week migration. Combined with the results of the maximum retracement decline of 13.2%, it can be seen that the strategy output does not show significant instability due to the local price jump, but shows strong return continuity and risk convergence characteristics.

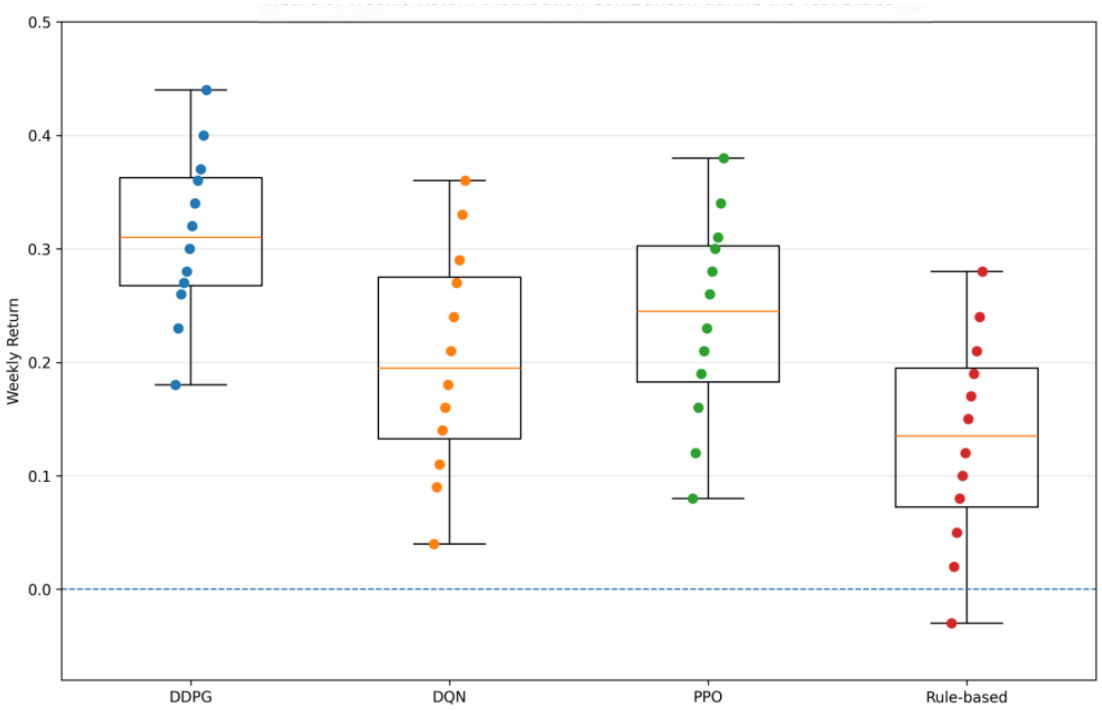

Figure 6: Boxplot of the distribution of weekly returns during the testing phase of DDPG

Fig. 6 shows that the median line of DDPG is significantly higher than that of the other comparison methods, the length of the box is relatively shorter, and the upper and lower whisker are stably distributed around 0.18 and 0.44, indicating that the weekly return distribution is more concentrated and the inter-week fluctuation is more controllable. Although the revenue interval of DQN and PPO was in the positive range, the box position moved down as a whole, indicating that the revenue level and stability of DQN and PPO were weaker than that of DDPG. The rule-based method is more discrete, and the lower bound of return has approached zero, which reflects that it is more susceptible to local volatility interference under complex market conditions. Combined with the inference record, the average single decision delay of the model in the test phase is 7.8 ms, and the video memory occupation is maintained at about 186 MB. It shows that the method is not only consistent with the summary results in the cumulative return rate and Sharpe ratio, but also has the computational efficiency required for online deployment. Based on the income distribution and the above-mentioned risk indicators in the figure, it can be judged that DDPG has formed a clear returnrisk correspondence, and shows stronger continuous decision-making ability and engineering adaptability in high-frequency trading scenarios.

5.2 Policy performance analysis of deep deterministic policy gradient algorithm and comparison model

This section further compares the strategic performance of DDPG with contrast models in the HFT task. In order to ensure that the evaluation results are consistent with the above income risk analysis, the experiment still uses 168000 5-min market time slices, 42 price variables, 6 bidding features and 18400 settlement records to repeat the tests of different models in a unified matching environment. The comparison objects include DQN, PPO and rule-based bidding methods. Each model receives the same state input, and completes the training and inference under the same transaction boundary and matching frequency. Table 4 shows the results of each model on cumulative return rate, Sharpe ratio, maximum rewind and decision

delay.

As shown in Table 4, the advantage of DDPG in the continuous offer space is obvious. The cumulative return rate and Sharpe ratio of this model are 18.7% and 1.64, which are higher than those of the other three methods. The decision delay remains at 7.8 ms, indicating that the strategy output can meet the real-time requirements of high-frequency trading scenarios. Due to the strong discretization of actions, DQN is prone to insufficient action granularity in the face of the market state in which the declaration price and the declaration quantity change synchronously, and the cumulative return rate is 14.2%. PPO performs better in training stability, but the continuous action correction range is conservative, the cumulative return rate is 16.1%, and the Sharpe ratio is 1.39. Although the rule-based method has faster inference speed and the delay is only 5.9 ms, its revenue expansion ability and risk convergence ability are weaker than the learning strategy. It can be seen that continuous action modeling is more suitable than fixed rule matching to deal with the bidding process with price jump, transaction feedback and inventory adjustment.

Table 4: Strategy performance comparison between DDPG and contrast models

Model	Cumulative Return / %	Sharpe Ratio	Maximum Drawdown / %	Decision Latency / ms
Rule-based	11.6	1.02	14.8	5.9
DQN	14.2	1.21	12.9	6.8
PPO	16.1	1.39	11.4	8.6
DDPG	18.7	1.64	9.8	7.8

In order to further illustrate the source of DDPG performance, we design ablation experiments to remove the temporal feature encoding layer, risk penalty term, target network smoothing and priority sampling mechanism, respectively. The relevant results are shown in Table 5. The full model remains optimal in terms of cumulative return, Sharpe ratio and retracement control. After removing the time series feature coding layer, the cumulative return rate drops to 15.8%, indicating that the local price pattern and transaction rhythm information have a direct impact on action generation. After removing the risk penalty term, the model can still maintain a high trading activity, but the maximum pullback expands to 13.7%, indicating that if the return driver lacks risk correction, the strategy is easy to expand the position in high volatility segments. After removing the smoothing of the target network, the Sharpe ratio decreases to 1.31, and the fluctuation of the value estimate is significantly enhanced. After removing the priority sampling mechanism, the cumulative rate of return and stability both decline, indicating that the effective proportion of high volatility samples in the training batch will affect the strategy identification ability.

Table 5: Ablation experimental results of key modules of DDPG

Model Configuration	Cumulative Return / %	Sharpe Ratio	Maximum Drawdown / %	Decision Latency / ms
Full Model	18.7	1.64	9.8	7.8
Without Temporal Feature Encoding Layer	15.8	1.34	12.6	7.3
Without Risk Penalty Term	17.2	1.22	13.7	7.7
Without Target Network Smoothing	16.0	1.31	12.8	7.6
Without Priority Sampling Mechanism	16.4	1.36	12.1	7.5

Comprehensive comparison results show that the advantage of DDPG does not come from a single module, but from the synergy of timing encoding, continuous action output, risk constraint and stable update mechanism. The model formed a balanced technical performance among benefit, risk and deployment efficiency, and also verified the market interaction environment, strategy input mapping and update process in the experiment. This structure not only retains the expressive power of deep reinforcement learning, but also has good conditions for engineering implementation.

6 Discussion

From the experimental results, the advantages of DDPG in high-frequency trading scenarios are not only reflected in a single income index, but also reflected in the synchronous improvement of income expansion, risk convergence and online execution efficiency. Based on the training and testing links formed by 168000 5-min market time segments, 42 price variables, 6 bidding features and 18400 settlement records, the model achieves 18.7% cumulative return rate and 1.64 Sharpe ratio under a unified matchmaking environment, and the average decision delay is maintained at 7.8 ms. It shows that the method has the ability of real-time inference for streaming market data. Compared with DQN, PPO and rule-based methods, the advantages of DDPG in continuous action modeling are clearer. DQN is limited to discrete action division, and the action granularity is coarse in the face of the scene where the offer offset and the declaration quantity change synchronously. Although PPO maintained good training stability, its motion correction amplitude was relatively conservative in high fluctuation periods. The regular method has low computational overhead, but it is difficult to continuously track the nonlinear linkage between node price, transaction depth and net position. The ablation results also illustrate that temporal feature encoding, risk penalty term, target network smoothing, and priority sampling are not additional modules independent of each other, but are important components that jointly determine the convergence quality of the policy. After removing these structures, both the cumulative return and Sharpe ratio decrease, and the maximum drawdown increases, indicating that the stable generation of HFS strategies depends on the complete computational link. Further, the high response area in Fig. 5 is concentrated in the positive spread of 0.8%-1.6% and the transaction depth range of 0.55-0.72, the unit time return is maintained between 0.041-0.048, and the median weekly return is 0.31, indicating that the strategy output can not only identify the tradable window, but also maintain a good migration consistency in the cross-week operation.

7 Conclusions

Focusing on the strategy generation task in the power market high-frequency trading, this paper constructs a DDPG computing framework for continuous bidding decision-making, and completes the organization of time series data, the modeling of market interaction environment, the mapping of strategy input, the design of network training target and the analysis of strategy update process. Experimental results show that the framework can integrate node marginal price, load deviation, transaction feedback and inventory constraint into a unified state representation, and output executable continuous bidding actions according to the unified state representation, which has a good ability of online transaction adaptation. At the same time, there is still room for further improvement of the current research. The sample data mainly come from the unified matching environment, and the coverage of different market systems and regional differences is still insufficient. Although

the reward function includes return, slip point and risk factors, the description of tail shock under extreme volatility conditions can still be further deepened. The existing framework focuses on single-agent, and the expression of multi-agent game relationships and cross-market coupling scenarios is not complete enough. The follow-up research can be further advanced from three aspects. On the one hand, the coverage of cross-regional and cross-time period samples can be further extended, so that the model can adapt to more complex market environment and price structure. On the other hand, the robust training mechanism can be improved for abnormal fluctuation scenarios, and the multi-agent cooperative update and lightweight deployment method can be combined to enhance the migration ability, execution efficiency and engineering adaptability of the model in the real power trading system. In addition, interpretable constraints and online self-correction mechanism can be added to make the policy generation process have clearer decision basis and better landing conditions while maintaining stability.

References

[1] Taghizadeh A, Montazeri M, Kebriaei H. Deep reinforcement learning-aided bidding strategies for transactive energy market[J]. IEEE Systems Journal, 2022, 16(3): 4445-4453.

[2] Ochoa T, Gil E, Angulo A, et al. Multi-agent deep reinforcement learning for efficient multi-timescale bidding of a hybrid power plant in day-ahead and real-time markets[J]. Applied Energy, 2022, 317: 119067.

[3] Xu H, Wen J, Hu Q, et al. Energy procurement and retail pricing for electricity retailers via deep reinforcement learning with long short-term memory[J]. CSEE Journal of Power and Energy Systems, 2022, 8(5): 1338-1351.

[4] Xu H, Wen J, Hu Q, et al. Energy procurement and retail pricing for electricity retailers via deep reinforcement learning with long short-term memory[J]. CSEE Journal of Power and Energy Systems, 2022, 8(5): 1338-1351.

[5] Zhang Y, Yang Q, Li D, et al. A reinforcement and imitation learning method for pricing strategy of electricity retailer with customers' flexibility[J]. Applied Energy, 2022, 323: 119543.

[6] Wang J, Guo C, Yu C, et al. Virtual power plant containing electric vehicles scheduling strategies based on deep reinforcement learning[J]. Electric power systems research, 2022, 205: 107714.

[7] Han D, Huang W, Ren H, et al. Machine learning analytics for virtual bidding in the electricity market[J]. International Journal of Electrical Power & Energy Systems, 2022, 143: 108489.

[8] Li Y, Yu N, Wang W. Machine learning-driven virtual bidding with electricity market efficiency analysis[J]. IEEE Transactions on Power Systems, 2021, 37(1): 354-364.

[9] Wang J, Li L, Zhang J. Deep reinforcement learning for energy trading and load scheduling in residential peer-to-peer energy trading market[J]. International journal of electrical power & energy systems, 2023, 147: 108885.

[10] Harder N, Qussous R, Weidlich A. Fit for purpose: Modeling wholesale electricity markets realistically with multi-agent deep reinforcement learning[J]. Energy and AI, 2023, 14: 100295.

[11] Ren K, Liu J, Liu X, et al. Reinforcement Learning-Based Bi-Level strategic bidding model of Gas-fired unit in integrated electricity and natural gas markets preventing market manipulation[J]. Applied Energy, 2023, 336: 120813.

[12] Feng B, Liu Z, Huang G, et al. Robust federated deep reinforcement learning for optimal control in multiple virtual power plants with electric vehicles[J]. Applied Energy, 2023, 349: 121615.

[13] Liu C, Rao X, Zhao B, et al. Deep reinforcement learning-based optimal bidding strategy for real-time multi-participant electricity market with short-term load[J]. Electric Power Systems Research, 2024, 233: 110404.

[14] Wu J, Wang J, Kong X. Intelligent strategic bidding in competitive electricity markets using multi-agent simulation and deep reinforcement learning[J]. Applied Soft Computing, 2024, 152: 111235.

[15] Tschora L, Pierre E, Plantevit M, et al. Electricity price forecasting on the day-ahead market using machine learning[J]. Applied Energy, 2022, 313: 118752.

[16] Das R, Bo R, Chen H, et al. Forecasting nodal price difference between day-ahead and real-time electricity markets using long-short term memory and sequence-to-sequence networks[J]. IEEE Access, 2021, 10: 832-843.

[17] Meng A, Wang P, Zhai G, et al. Electricity price forecasting with high penetration of renewable energy using attention-based LSTM network trained by crisscross optimization[J]. Energy, 2022, 254: 124212.

[18] Khojasteh M, Faria P, Vale Z. A robust model for aggregated bidding of energy storages and wind resources in the joint energy and reserve markets[J]. Energy, 2022, 238: 121735.

[19] Jiang P, Nie Y, Wang J, et al. Multivariable short-term electricity price forecasting using artificial intelligence and multi-input multi-output scheme[J]. Energy Economics, 2023, 117: 106471.

[20] Krishna Prakash N, Singh J G. Electricity price forecasting using hybrid deep learned networks[J]. Journal of Forecasting, 2023, 42(7): 1750-1771.

[21] Cantillo-Luna S, Moreno-Chuquen R, Lopez-Sotelo J, et al. An intra-day electricity price forecasting based on a probabilistic transformer neural network architecture[J]. Energies, 2023, 16(19): 6767.

[22] Sai W, Pan Z, Liu S, et al. Event-driven forecasting of wholesale electricity price and frequency regulation price using machine learning algorithms[J]. Applied Energy, 2023, 352: 121989.

[23] Jiang H, Dong Y, Wang J. Electricity price forecasting using quantile regression

averaging with nonconvex regularization[J]. Journal of Forecasting, 2024, 43(6): 1859-1879.

[24] Yang Y, Guo J, Li Y, et al. Forecasting day-ahead electricity prices with spatial dependence[J]. International Journal of Forecasting, 2024, 40(3): 1255-1270.