Toward Automatic Market Making: An Imitative Reinforcement Learning Approach With Predictive Representation Learning

Siyuan Li, Yafei Chen, Hui Niu, Jiahao Zheng, Zhouchi Lin, Jian Li, Jian Guo, and Zhen Wang, *Fellow, IEEE*

***Abstract*—Market making (MM) is a crucial trading problem, where a market maker stands ready to buy and sell the asset at a publicly quoted price to provide market liquidity continuously. The primary challenges in market making include position risk, liquidity risk, and adverse selection. Emerging research works investigate applying reinforcement learning (RL) techniques to derive automatic MM strategies. However, existing methods mainly focus on addressing inventory risk using only single-level quotes, which restricts the trading flexibility. In this paper, we shed light on the optimization of market makers' returns under a smaller risk while ensuring market liquidity and depth. This paper proposes a novel RL-based market-making strategy *Predictive and Imitative Market Making Agent* (PIMMA). First, to ensure adequate liquidity, we design an action space to enable stably allocating orders of multi-level volumes and prices. Beyond that, we apply queue position information from these multi-price levels to encode them in the state representations. Second, aiming at alleviating adverse selection, we draw auxiliary signals into state representation and design a representation learning network structure to catch implicit information from the price-volume fluctuations. Finally, we develop a novel reward function to earn a fortune while avoiding holding a large inventory. With a provided expert demonstration, our method augments the RL objective with imitation learning and learns an effective MM policy. Experiments are conducted to evaluate the proposed method based on realistic historical data, and the results demonstrate PIMMA outperforms RL-based strategy in the perspectives of earning decent revenue and information by adopting the multi-risk aversion strategy.**

***Index Terms*—Market making, deep reinforcement learning, predictive representation learning, imitation learning.**

Received 11 November 2023; revised 6 May 2024 and 17 June 2024; accepted 7 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62306088 and in part by Songjiang Lab under Grant SL20230309. *(Corresponding author: Siyuan Li.)*

Siyuan Li and Yafei Chen are with the Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China (e-mail: siyuanli@hit.edu.cn; chenyf0330@163.com).

Hui Niu and Jian Li are with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China (e-mail: niuh17@mails.tsinghua.edu.cn; lijian83@mail.tsinghua.edu.cn).

Jiahao Zheng, Zhouchi Lin, and Jian Guo are with the International Digital Economy Academy (IDEA), Shenzhen 518048, China (e-mail: zhengjiahao@idea.edu.cn; linzhouchi@idea.edu.cn; guojian@idea.edu.cn).

Zhen Wang is with the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhenwang0@gmail.com).

Recommended for acceptance by Y. Yu.

Digital Object Identifier 10.1109/TETCI.2024.3451476

I. Introduction

FOR the sake of liquidity provision, Market Making (MM) is considered a notable issue in financial markets, where market makers continuously quote both bid and ask prices referring to the data in the limit order book (LOB). During the trading process, market makers face inherent risks, including (1) position risk (2) liquidity risk, and (3) adverse selection. Position risk is also known as inventory risk, which denotes market makers holding a position subjected to inventory size and market volatility. Liquidity risk represents that market makers encounter difficulties in buying or selling assets without causing significant price movement, which arises from occasionally insufficient trading activity or interest. Adverse selection occurs in competitive transactions where information asymmetry leads to unfavorable outcomes for the less informed party. Based on inventory levels, market conditions, and individual variables, market makers dynamically quote bids and asks for limit orders in the LOB of specific securities, and their aim is to optimize risk-adjusted returns while ensuring market liquidity and depth.

Numerous research works have been proposed to investigate the market-making problem. Conventional market-making approaches [1], [2], [3] are based on heuristic rules and mathematical models, which can hardly dynamically adapt to changing market conditions. Advanced works such as [4], [5] utilize deep neural networks (DNNs) to predict price movements, and use the price prediction to conduct market making. Recently, as reinforcement learning (RL) has demonstrated strong abilities in sequential decision making [6], [7], [8], [9], several studies investigate applying RL techniques to realize automatic MM while obtaining a presentable price prediction [10], [11], [12], to achieve a superior MM strategy. Ref. [10] developed a market-making agent with value-based RL methods. As an improvement, [13], [14] proposed to employ the actor-critic RL method, such as proximal policy optimization (PPO) and advantage actor—critic (A2C) to learn the MM policy. However, the existing RL-based MM methods pay more attention to solving inventory risk, but neglect the liquidity risk and adverse selection. This potentially leads to wider spreads, reduced efficiency, decreased participation, and increased volatility. To accomplish better market making for a healthier market, we propose a novel RL-based market-making strategy, *Predictive and Imitative Market Making Agent (PIMMA)*. In contrast to the previous approaches, the PIMMA model is able to ensure

adequate liquidity, reduce inventory risk, and alleviate adverse selection with the same aim of earning a reasonable fortune.

PIMMA is a novel data-driven deep RL-based MM approach incorporating effective state representation learning and imitation learning techniques. To reduce liquidity risk, we develop an action space using multi-level quotes, which is able to keep a good queue position and avoid frequent cancellations of orders. Then, we encode the information of queue position[1] for the orders at each price level in state representation. Based on the stable reference price as formulated in Section III-B1, the proposed flexible action space can determine different volume distributions of orders covering all price levels. It is worth mentioning that existing RL-based methods in MM such as [3], [14], [15], [16], [17] only use unstable mid-price as reference to determine single-level price range between bid and ask, which causes inevitably frequent order cancellations and subsequently queue position losing. Consequently, these methods suffers from high liquidity risks. The following works [18], [19], [20], [21] improve liquidity by applying a ladder strategy through setting up more flexible volumes or prices of orders, yet they are still constrained by the fluctuating mid-price, since they also use it as a reference price.

High-frequency LOB data contains much richer information about the behavior of financial assets than their prices alone. DNNs have shown strong capabilities in knowledge extraction, and capturing complex patterns in LOB data [5]. Ref. [4] employs DNNs to generate signals and derive rule-based MM strategies in a supervised learning (SL) manner. For the reason of forecasting adverse selection, we analyze the LOB data to extract multi-granularity predictive signals as auxiliary observations. Note that in the proposed approach, these auxiliary observations are not used to handcraft rule-based strategies, since the rigid hand-crafted market-making strategies can hardly adapt to the changing markets. Instead, we design a representation learning network structure to mine the hidden information in the time-series LOB data, and train the market making policy end-to-end with this representation network.

We have developed an advanced action space and state representation for the market making problem. Moreover, we design a thorough reward function, taking spread, inventory risk, and performance reward into account with the main object of yielding considerable profits. Beyond that, since leveraging expert data is a favorable approach to improve the exploration ability of RL algorithms [22], [23], [24], we design a linear signal-based expert, and incorporate the trading knowledge generated by the sub-optimal expert into the actor-critic RL framework. Taking advantage of the delicate reward function and the expert demonstration, PIMMA is able to learn a successful market-making strategy in the complex trading markets.

To summarize, we design suitable state space, action space, and reward functions for the market making problem, and develop a novel RL-based market-making learning approach, PIMMA. The proposed approach combines a state representation learning unit (SRLU) with an imitative DRL unit to address the MM optimization problem. The SRLU employs multi-granularity predictive signals as auxiliary variables to take advantage of the labels containing future information. The SRLU also utilizes a temporal convolution and spatial attention (TCSA) network to abstract useful representations from noisy historical market data. Based on these effective representations given by SRLU, PIMMA trains the policy in the imitative DRL unit with an advanced RL algorithm that incorporates imitation learning techniques.

The contributions of this paper are summarized as follows:

- PIMMA can leverage the valuable information contained in high-frequency LOB data. It encodes the queue position information and extracts multi-granularity predictive signals for state representations. By incorporating a TCSA network in SRLU, the proposed approach enhances its ability to forecast risks and make informed market-making decisions.
- By designing an action space using multi-level quotes, PIMMA effectively reduces liquidity risk and minimizes order cancellations. Moreover, we design a reward function that considers spread, inventory risk, and performance rewards. Then we demonstrate the applicability and availability of these formulations in a real-world market for inventory management.
- PIMMA integrates the trading knowledge from a linear signal-based expert into the actor-critic RL framework. Leveraging both expert data and RL algorithms, the proposed approach is able to learn an effective market-making policy.
- Experiment results on four realistic future datasets demonstrate that the proposed approach outperforms the baselines including the rule-based methods and DRL methods in terms of the risk-adjusted returns. We also conduct thorough ablation experiments to investigate the performance of the learned policy and validate the effectiveness of the various components of PIMMA.

II. Preliminaries

A. Deep Reinforcement Learning

RL is a class of machine learning algorithms for direct adaptive control in a Markov Decision Process (MDP) based on learning by interacting with the environment. A discrete-time MDP can be denoted by a tuple $M =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ describes the state transition probabilities, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function and $\gamma \in (0, 1]$ is the discount factor.

At time step t, an agent observes the state $\boldsymbol{s}_t \in \mathcal{S}$ and takes an action $\boldsymbol{a}_t \in \mathcal{A}$. At time step $t + 1$, the agent is transited to a new state $\boldsymbol{s}_{t+1}$ and receives a reward $r_t = \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$. The procedure is then iterated and the trajectory of the sequence is obtained. Let T denote the ending time step, and $T = \infty$ if the episodes are never-ending. The agent needs to optimize a policy, $\boldsymbol{a}_t \sim \pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ (if the policy is deterministic, $\boldsymbol{a}_t = \pi(\boldsymbol{s}_t)$), to specify an action $\boldsymbol{a}_t$ for a given state $\boldsymbol{s}_t$. The optimization

[1]The queue position refers to the position of an order in the exchange's order queue, which indicates its relative priority for execution compared to other orders.

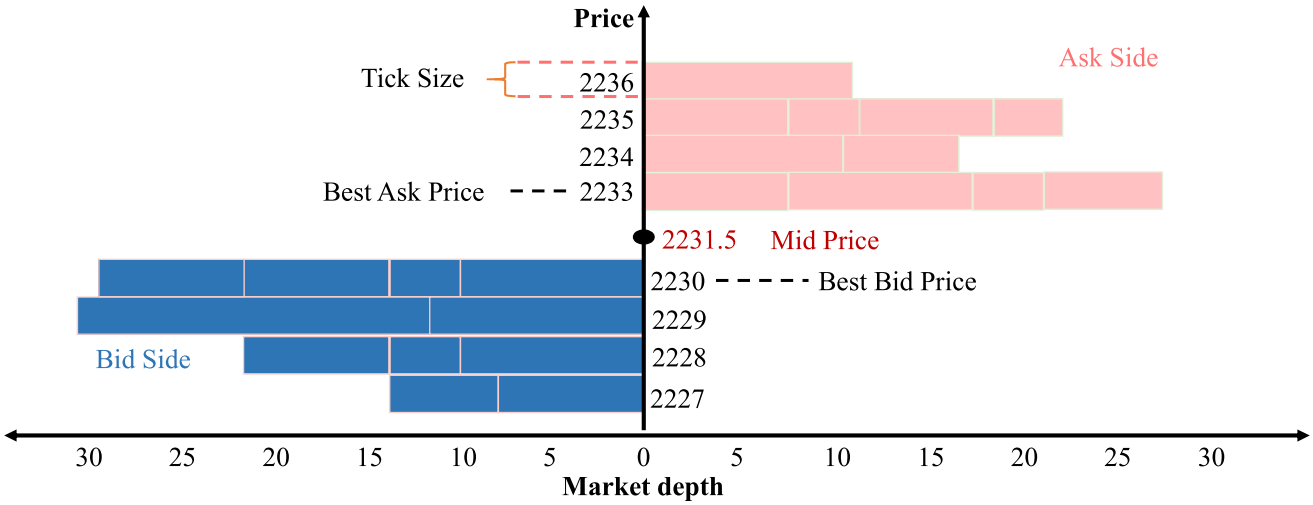

Fig. 1. Visualization of the LOB.

objective is to maximize the cumulative rewards under the learned policy $\mathbb{E}_\pi[\sum_{t=0}^{T}\gamma^t r_t]$.

B. LOB and Market Making

In this paper, we focus on the Market making (MM) in limit order book (LOB) markets, where market makers are the member firms to inject liquidity and trade volume into assets.

The LOB is a data structure that contains the entrusted prices and volumes of the buyer and seller. Fig. 1 visualizes the first five levels of a LOB. The buyer's orders - the *bid* orders(blue), are sorted in descending using their price, while the seller's orders - the *ask* orders(pink), are sorted increasingly [25]. Orders arrive at an arbitrary time. When a bid order's price exceeds an ask order's price, they are matched, and assets are exchanged. If volumes differ, the order with the largest volume remains in the LOB, deducting the utilized volume [5]. The difference between the highest bid price and the lowest ask price is the quoted spread, with the mid-price being the average of these prices. Levels of bids and asks show available volumes at their respective prices. And the tick size defines the minimal unit of the bid and ask price levels.

MM refers to a class of trading strategies where market makers provide liquidity to financial markets [26]. It involves continuously quoting both bid and ask prices, establishing a two-way market and ensuring a counterparty for buyers and sellers [26]. In this way, unmatched orders are collected in a LOB and matched orders result in trades.

Based on market conditions, market makers capture spread to maintain orderly markets, balance inventory, manage risk, and adjust prices to minimize adverse price movements and volatility. They ensure availability for transactions through continuous presence and offer bid and ask prices for securities or assets. Spreads would be widened during low liquidity or high volatility and narrowed during high liquidity or low volatility. To maintain depth, market makers provide sufficient orders at various prices for large trades with minimal price impact. They also absorb supply-demand imbalances by purchasing excess supply or selling inventory, promoting market liquidity. With the use of trading algorithms and market data analysis, they optimize decisions while monitoring the LOB. More advanced market makers may choose to hold a non-zero inventory to exploit clear trends, whilst simultaneously capturing the spread [10].

TABLE I
NOTATION LIST

Type	Notation	Meaning
state	$\mathbf{s^s}$	states containing predictive signals
	$\mathbf{s^m}$	states containing market variables
	$\mathbf{s^p}$	states containing variables
LOB	p_{ref}	the reference price of the market
	m_t	the midprice of the market
	ask_t	the best ask price of the market
	bid_t	the best bid price of the market
	Q_i	price limit at level i in an LOB
	l_i	queue length of the LOB at price level i
	v_t^i	volume of personal orders at price level i
	q_t^i	personal queue position at price level i
action	m_t^*	target quoted midprice
	δ_t^*	target quoted spread
	ϕ_t^{ask}	vector that determines the ask-side volume distribution
	ϕ_t^{bid}	vector that determines the bid-side volume distribution
	z_t	personal inventory

III. Problem Formulation

At the beginning of this section, we provide a notation list of the MM problem formulation in Table I for readers to look up the notations. Then, we introduce the MM procedure. Next, we present the formulated MM agent.

A. Market Making Procedure

To test our agent and replicate real-world market behavior with minimal assumptions, we take historical data, order-book depth, and transaction information into account, building a marketing environment with a realistic, data-driven simulator of the LOB developed in [10]. The simulator tracks the top 5 price levels in the LOB and allows an agent to place its orders on it.

The market is reconstructed using aggregated information of the transactions and the LOB at the half-a-second level. Although we submit orders every half a second, in Step (4) of the following procedure, the matching of the orders uses level-II data. These data are with a higher frequency than the half-a-second level, and they are in a tick-by-tick form provided by the real historical data in the exchange, which are with high fidelity. Notably, the orders placed by our simulated agent have no impact on the actual market since it is reconstructed from historical data.

We formulate MM as an episodic RL task. The procedure of running an episode is specified as follows:

1) Choose a random start time for the episode such that the entire episode occurs within a random trading day. Initialize the environment and the simulator.
2) Let the agent choose the desired volumes and price levels at which the agent would like to be positioned in the LOB.
3) Turn these desired positions into orders, including canceling orders from levels with too much volume and placing new limit orders.
4) Match the orders in the market-replay simulator according to the price-time priority. Update the agent's cash and inventory of the traded asset and track profit and loss.
5) Repeat steps 2-4 until the episode terminates.

B. Market Making Agent

1) State Space: At time step t, the MM agent observes the state formulated by:

$$\boldsymbol{s}_t = (\boldsymbol{s}_t^m, \boldsymbol{s}_t^s, \boldsymbol{s}_t^p), \tag{1}$$

where $\boldsymbol{s}_t^m$ denotes the *market variables* containing information about the current market conditions, including the technical indicators such as MACD, RSI, RSI, etc.; $\boldsymbol{s}_t^s$ denotes the *signal variables*, including auxiliary predictive signals; $\boldsymbol{s}_t^p$ denotes the *private variables*, including the current inventory z_t, the queue position information of the current order $\boldsymbol{s}_t^q$ on the LOB and the remaining volume at each price level $\boldsymbol{s}_t^v$:

$$\boldsymbol{s}_t^q = \left(\boldsymbol{q}_t^{-K}, \ldots, \boldsymbol{q}_t^{-1}, \boldsymbol{q}_t^1, \ldots, \boldsymbol{q}_t^K\right) \tag{2}$$

$$\boldsymbol{s}_t^v = \left(\boldsymbol{v}_t^{-K}, \ldots, \boldsymbol{v}_t^{-1}, \boldsymbol{v}_t^1, \ldots, \boldsymbol{v}_t^K\right) \tag{3}$$

Here $\boldsymbol{q}_t^i$ and $\boldsymbol{v}_t^i$ denotes the formations of queue position value and resting volume at price level *i* respectively. *K* represents the available number of price levels at each side of bid and ask.

We use following steps to formulate $\boldsymbol{q}_t^i$. As mentioned before, relying solely on unstable mid-prices as a reference to divide the price range in LOB into the bid and ask side, sometimes results in frequent unnecessary price level changes. So firstly, We divide the different price levels as follows. We formulate a reference price p_{ref} as the center of the order book according to [27]. Then, based on this reference price, we divide the order book into the bid side and the ask side. The price levels on the bid side are below the reference price, while the price levels on the ask side represent price levels above the reference price. In the bid side, we have *K* price levels, denoted as Q_{-i}, where $i = 1, 2, \ldots, K$, while in the ask side, we also have Q_{+i} as price levels, where $i = 1, 2, \ldots, K$. The price width of each price level is determined by the tick size. The distance between $Q_{\pm i}$ and p_{ref} is $i - 0.5$ ticks. Note that our approach allows for empty limits within the LOB. The queue length at Q_i is denoted by l_i.

Next, we determine the reference price p_{ref}. Firstly, we set up a reference price $\tilde{p}_{ref}$ following midprice.

$$\tilde{p}_{ref,t} = m_t \text{ if the spread is odd,} \tag{4}$$

$$\tilde{p}_{ref,t} = m_t \pm \frac{\text{tick size}}{2} \text{ else,} \tag{5}$$

where the selection of the sign is based on proximity to the prior value. At the beginning of an episode, we set $p_{ref,0} = \tilde{p}_{ref,0}$. When the midprice changes, only if $l_{-1,t} = 0$ (or $l_{1,t} = 0$), $p_{ref,t}$ is updated to $\tilde{p}_{ref,t}$.

Now we proceed to calculate the queue position value at each price level i. In the time t, let's assume there are m_t^i orders resting at level i. The queue position value of the j-th order can be defined as:

$$q_t^{i,j} = \frac{l_i^{\text{front},j}}{l_t^i}, \tag{6}$$

where $l_i^{\text{front},j}$ denotes the queue length in front of this order. Thus, the queue position value at price level i can be defined as the volume-weighted average of the queue position values of the m_t^i orders:

$$q_t^i = \sum \frac{l_t^{\text{front},i,j}}{l_t^i} \cdot \frac{v_t^{i,j}}{l_t^i} \tag{7}$$

Covering the information of the current quotes could help the agents avoid frequent canceling and replacing orders, thus losing their queue position.

2) Action Space: At time step t, the action $\boldsymbol{a}_t$ is given by

$$\boldsymbol{a}_t = (m_t^*, \delta_t^*, \boldsymbol{\phi}_t^{bid}, \boldsymbol{\phi}_t^{ask}), \tag{8}$$

where m_t^* and δ_t^* denote the desired quoted mid-price and spread respectively. That is, the desired lowest sell price for the agent is $m_t^* + \delta_t^*/2$ and the highest buy price is $m_t^* - \delta_t^*/2$; $\boldsymbol{\phi}_t^{bid}$ and $\boldsymbol{\phi}_t^{ask}$ are parameters that determine the volume distribution of the multi-level quotes. This action formulation enables the agent to decide how wide (the *width*) and how asymmetrically (the *skew*) w.r.t. the mid-price it sets the quotes. It resembles a smoothed version of the discrete action space proposed by [10]. Here m^* is the price relative to the right side of p_{ref}.

For example, in the situation where an agent places two-level orders on each side of the LOB, ϕ_t^{ask} can be a 1-dim parameter that defines a Binomial distribution for the volumes at the two price levels. The range of different action choices is depicted in Fig. 2(a). In practice, we employ a continuous action space and convert actions to the formulation defined in (8).

3) Reward Function: The way the market makers choose their prices is subject to several trade-offs, including probability of execution and spread, inventory risk, and compensation from the exchange. Three factors are suggested to be taken into consideration when formulating the rewards:

1) *Profit and loss (PnL):* PnL is a natural choice for the problem domain, including a realized PnL term (the left

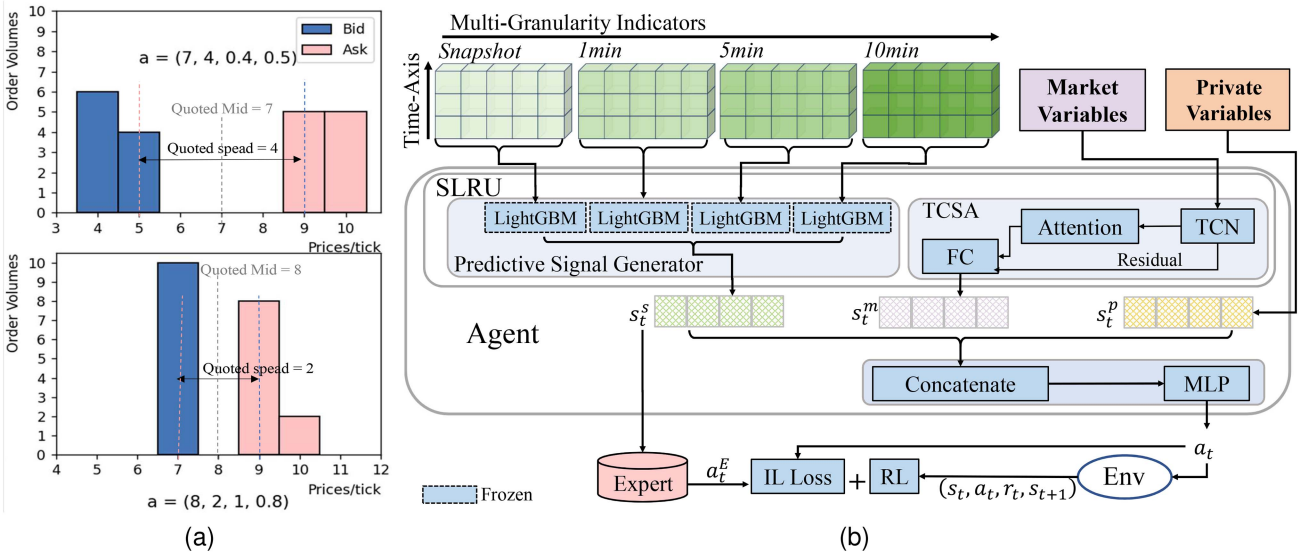

Fig. 2. (a) Two examples for action $a_t = (m_t^*, \delta_t^*, \phi_t^{bid}, \phi_t^{ask})$. (b) The proposed PIMMA learning framework.

part) and a floating PnL term (the right part), given by:

$$PnL_t = \left(\sum_{i \in A_t} p_i^a \cdot v_i^a - \sum_{j \in B_t} p_j^b \cdot v_j^b \right) + (p_{t+1} - p_t) \cdot z_{t+1}, \tag{9}$$

where p denotes the mid-price of the market; p^a, v^a (p^b, v^b) is the price and volume of the newly traded ask (bid) orders respectively; z denotes the current inventory and $z > 0$ when the agent holds a larger long position than the short position; A_t and B_t denote the ask order set and the bid order set at time t.

2) *Truncated Inventory Penalty:* [10] showed that the natural definition given above often leads to instability during learning and unsatisfactory out-of-sample performance. This phenomenon can be alleviated by introducing an additional inventory-dampening term. However, as advanced market makers may choose to hold a non-zero inventory to exploit clear trends while capturing the spread, it is reasonable to apply the dampening term only to a highly risky inventory level:

$$IP_t = -|z_t| \cdot \mathbb{I}(|z_t| > C). \tag{10}$$

When the inventory z_t is larger than a constant C, there is a penalty for the risks of holding the inventory.

3) *The Market Makers' compensation from the exchange* constitutes a primary revenue stream for numerous market makers. Therefore, ensuring a substantial volume of transactions to secure compensation holds significant importance. Therefore, a bonus term is incorporated to encourage transactions of the agent:

$$C_t = \sum_{i \in A_t} p_i^a \cdot v_i^a + \sum_{j \in B_t} p_j^b \cdot v_j^b. \tag{11}$$

Finally, to trade off the profits, risks, and number of transactions, we employ the combination of those three kinds of rewards to train the MM agent:

$$\mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) = PnL_t + \beta C_t + \eta IP_t, \tag{12}$$

where β and η are scaling factors. Considering exchanges generally encourage market makers to narrow bid-ask spreads to improve liquidity, the reward term in (11) is not crucial, so we set hyperparameter β as a small value.

IV. Algorithm

In this section, we introduce the proposed deep RL approach for MM with predictive representation learning. The learning framework is depicted in Fig. 2(b). Section IV-A elaborates on the SRLU which aims at predicting multiple signals while extracting useful representations from the noisy market data. Section IV-B presents the MM policy learning method, which incorporates the RL and imitation learning objectives.

A. State Representation Learning

The SRLU generates (i) auxiliary signals leveraging future labels and (ii) effective representations from historical market data.

1) Signal Generation: PIMMA pre-trains SL models to generate several signals with abundant information. The choice of the SL models is flexible. In this paper, we adopt LightGBM [4], a highly robust ensemble model based on decision trees, to generate four multi-granularity trend signals. These signals include both short-term and long-term ones. While the short-term indicators, the LOB indicators, that reflect the current imbalance of the buyers and sellers, indicate a short-term future price trend, the long-term technical signals generated from the price-volume time-series data, contain temporal relations of a long-term historical trading period. These signals are denoted by

$(y^{20}, y^{60}, y^{120}, y^{600})$, which are the labels of price movement trend after $1/6, 1, 2, 5$ minutes respectively.

The LOB indicators are derived from domain knowledge: the total number of trades in the previous 10 periods, the bid-ask spread at the start of the time step, the traded volume imbalance, the new-placed order imbalance, the volume-weighted average price in the previous k periods where $k = 10, 60, 120, 600$, the total number of large buys and sells. The natural labeling method is to classify the price movement trends as three categories: "Up", "Fluctuate", and "Down". Clearly, the labels $(y^{20}, y^{60}, y^{120}, y^{600})$ represent whether the prices will significantly increase or decrease after $\{1, 5, 15, 30\}$ minutes.

While training the RL policy, the parameters of the pre-trained LightGBM models are frozen, and the outputs constitute the auxiliary signal variables $\boldsymbol{s}^s$.

2) Attention-Based Representation Learning: Deep RL algorithms usually suffer from the low data-efficiency issue. Besides the auxiliary signal prediction, we propose a temporal convolution and spatial attention (TCSA) network to extract additional effective representations from the noisy market data $\boldsymbol{x}$. The structure of the TCSA network illustrated in Fig. 2(b) is described as follows. The proposed approach PIMMA first utilizes a temporal convolution network (TCN) [28] block to extract the time-axis relations in the data. Compared to recurrent neural networks, TCN has several appealing properties including parallel computation and longer effective memory. After conducting TCN operations on $\boldsymbol{x} \in \mathbb{R}^{F \times L}$ along the time axis, we obtain an output tensor denoted by $\hat{\boldsymbol{H}} \in \mathbb{R}^{F \times L}$, where F is the dimension of features, and L is the temporal dimension.

Afterward, PIMMA adopts an attention mechanism [29] to handle the spatial relationships among different features. Given the output vector of TCN, we calculate the spatial attention weight as

$$\hat{\boldsymbol{S}} = \boldsymbol{V} \cdot \text{sigmoid}\left((\hat{\boldsymbol{H}}\boldsymbol{W}_1) \cdot (\hat{\boldsymbol{H}}\boldsymbol{W}_2)^T + \boldsymbol{b}\right), \tag{13}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^L$, and $\boldsymbol{V} \in \mathbb{R}^{F \times F}$ are parameters to learn, $\boldsymbol{b} \in \mathbb{R}^{F \times F}$ is the bias vector. The matrix $\hat{\boldsymbol{S}} \in \mathbb{R}^{F \times F}$ is then normalized by rows to represent the correlation among features:

$$\boldsymbol{S}_{i,j} = \frac{\exp(\hat{\boldsymbol{S}}_{i,j})}{\sum_{u=1}^{F} \exp(\hat{\boldsymbol{S}}_{i,u})}, \forall 1 \le i \le F. \tag{14}$$

We adopt the ResNet [30] structure to alleviate the vanishing gradient problem in deep learning. The final representation abstracted from $\boldsymbol{x}$ is denoted by $H = S \times \hat{\boldsymbol{H}} + \boldsymbol{x}$, and it is then translated to a vector with dim F' using a fully connected layer: $\boldsymbol{s}^m = \text{sigmoid}(W_4 \cdot \text{ReLU}(\boldsymbol{H}W_3 + \boldsymbol{b}_3) + \boldsymbol{b}_4)$. The representation $\boldsymbol{s}^m$ is concatenated with the signal state $\boldsymbol{s}^s$ and private state $\boldsymbol{s}^p$.

B. Imitative Reinforcement Learning

Even with those effective trend signals generated by state representation learning, a rule-based strategy might sometimes does not behave well, since a rule-based strategy can hardly take all the possible market conditions into consideration. Due to the sequential nature of the MM task, we adopt the RL approach to train policies.

1) A Linear Signal-Based Expert for MM: With the aim of assisting to generalize and acquire a policy that can perform the task effectively, we devise *Linear in Trend and Inventory with Inventory Constraints(LTIIC)*, a linear rule-based expert to improve the performance of our agent. It is related to a strategy where a market maker adjusts its quote prices based on inventory z_t and predictive trend signals $\hat{y}_t \in \{-1, 0, 1\}$ at any time t. $LTIIC(a, b, c, d)$ is restricted by four constants a, b, c and d that a is an addend, b and d two factors, and $c(c > 0)$ control the range of inventory.

If $|z_t| < c$, the ask (bid) orders at the time t are posted at the following prices,

$$\begin{cases} ask_t^q = m_t + a + b \cdot z_t + d \cdot \hat{y}_t, \\ bid_t^q = m_t - a + b \cdot z_t + d \cdot \hat{y}_t, \end{cases} \tag{15}$$

where m_t is the mid price of the LOB. It should be highlighted that ask-side limit orders have a higher likelihood of execution during a short-term upward trend, compared to bid-side ones. Thus it is reasonable to expect that, on average, $|ask_t^q - m_t| > |m_t - bid_t^q|$ and therefore less risk exposure induced by adverse selection. If $|z_t| \ge c$, only the order on the opposite side is posted.

2) Policy Learning: We utilized the actor-critic RL framework [31], where the critic evaluates the action taken by the actor by computing the value function, and the actor (policy) is optimized to maximize the value output by the critic. To improve the sample efficiency, we use the off-policy actor-critic method TD3 [32] as the base learner,[2] and the policy π is updated with the deterministic policy gradient [33]:

$$\pi = \arg\max_{\pi} \mathbb{E}_{(\boldsymbol{s},a)\sim D}\left[Q(\boldsymbol{s}, \pi(\boldsymbol{s}))\right], \tag{16}$$

where Q is a value function approximating the expected cumulative reward, $Q(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}[\sum_{i=t}^{T} \gamma^{i-t} r_i | \boldsymbol{s}_t, \boldsymbol{a}_t]$.

In (16), D denotes the replay buffer collected by a behavior policy, which is generated by adding some noise to the learned policy π. Following the TD3 method [32], the value function Q is optimized in a twin-delayed manner with the data sampled from the replay buffer D.

Learning with a pure RL objective in (16) is extremely difficult, due to the high-dimensional state space and action space, and the stochastic trading environment causes a hard exploration problem. To promote policy learning in such a complex trading environment, we propose to augment the RL method with the objective of imitating the quoting behavior in an expert dataset D_{exp} as follows:

$$\begin{aligned} \pi = \arg\max_{\pi} & \mathbb{E}_{(\boldsymbol{s},a)\sim D}\left[Q(\boldsymbol{s}, \pi(\boldsymbol{s}))\right] \\ & - \mathbb{E}_{(\boldsymbol{s},\hat{a})\sim D_{exp}}\left[\lambda \cdot (\pi(\boldsymbol{s}) - \hat{a})^2\right], \end{aligned} \tag{17}$$

where λ is a scaling coefficient that balances maximizing the Q values and minimizing the behavior cloning (BC) loss. λ decreases with the growth of the training steps.

[2]The proposed approach is compatible with other actor-critic algorithms beyond TD3 as well.

TABLE II
THE COMPARISON RESULTS OF THE PROPOSED METHOD AND THE BENCHMARKS

	RB				FU				CU				AG			
	EPnL[10^3]	MAP[unit]	PnLMAP	RPT[%]	EPnL[10^3]	MAP[unit]	PnLMAP	RPT[%]	EPnL[10^3]	MAP[unit]	PnLMAP	RPT[%]	EPnL[10^3]	MAP[unit]	PnLMAP	RPT[%]
FOIC	3.23 ± 4.35	255 ± 111	14 ± 22	8 ± 10	-7.79 ± 9.25	238 ± 135	-43 ± 56	-18 ± 16	-33.05 ± 27.63	206 ± 141	-161 ± 224	-275 ± 232	-48.39 ± 28.83	189 ± 154	-250 ± 335	-323 ± 345
LIIC	2.26 ± 3.32	123 ± 32	20 ± 29	6 ± 8	-6.89 ± 6.66	115± 30	-66 ± 69	-15 ± 11	-24.19 ± 14.83	150 ± 20	-164 ± 513	-313 ± 149	-38.9 ± 26.2	142 ± 45	-302 ± 243	-319 ± 197
LTIIC	9.16 ± 4.87	65 ± 6	139 ± 68	45 ± 13	8.26 ± 2.64	52 ± 3	160 ± 50	27 ± 7	-16.74 ± 15.81	112 ± 109	-190 ± 203	-178 ± 166	-32.57 ± 22.8	128 ± 22	-264 ± 166	-224 ± 148
RL_{DS}	4.36 ± 1.64	**38 ± 4**	114 ± 38	14 ± 4	7.31 ± 5.38	76 ± 29	90 ± 46	24 ± 13	-19.7 ± 17	214 ± 109	-92 ± 298	-192 ± 169	-25.43 ± 23.83	107 ± 37	-237 ± 235	-165 ± 105
DRL_{OS}	8.22 ± 3.70	51 ± 4	156 ± 61	30 ± 6	11.03 ± 13.87	**37 ± 3**	30 ± 36	3 ± 4	-18.9 ± 18.02	647 ± 2367	-99 ± 147	-141 ± 267	-28.39 ± 27.92	169 ± 154	**-167 ± 135**	-199 ± 185
PIMMA	**16.46 ± 9.10**	96± 13	**165 ± 74**	**64 ± 20**	**28.10 ± 10.27**	102 ± 14	**274 ± 89**	**87 ± 18**	**-4.86 ± 10.17**	**111 ± 28**	**-43 ± 87**	**-108 ± 161**	**-14.5 ± 20.2**	**102 ± 14**	-274 ± 89	**- 87 ±18**

As the expert dataset contains reasonable suboptimal MM behaviors, the agent benefits from the imitation learning techniques (17) through abstracting advanced trading knowledge. Thus the proposed method could achieve more efficient exploration and policy learning in the highly stochastic market environment compared to the RL methods without imitation learning.

V. EXPERIMENTS

A. Experiment Setup

1) Dataset: We conduct experiments on four datasets comprised of historical data of the spot month contracts of the FU, RB, CU, and AG futures from the Shanghai Futures Exchange (SHFE).[3] These four datasets exhibit differences in terms of trading volume, price volatility, liquidity, and market structure that are sufficient to support our experiments. The data consists of the 5-depth LOB and aggregated trade information associated with a 500-milliseconds real-time financial period. We use the data from July 2021 to March 2022 (126 trading days) for training with 20% as the validation set, and test model performance on April 2022 $\sim$ July 2022 (60 trading days). In each episode, the agent adjusts its 2-level bids and asks every 500 milliseconds, with a fixed total volume $N = 20$ on each side. The episode length is set to 1.5 trading hours, with $T = 10800$ steps. We give an additional truncated inventory penalty to the agent when the inventory exceeds $C = (\pm)40$.

2) Hyperparameters: When tuning the hyperparameters of the state representation learning unit, the following hyperparameters are fixed:

1) boosting = "gbdt"
2) early_stopping_round = 50
3) n_estimators = 1000

The thresholds of the price changing rates for the four labels $(y^{20}, y^{60}, y^{120}, y^{600})$ in signal generation, is set to 0.05%, 0.1%, 0.15% and 0.2% of the current market price respectively for classification. For example, if the price changing rate after 1 minute is larger or smaller than 0.05% of the current market price, the first dimension of the label is 1 or -1, otherwise, the 1-minute label is set to 0. Those thresholds are determined empirically by the 75-th and 25-th percentile of the multi-granularity price changing rates. Other hyperparameters and the search space for tuning are in Table III. The selection of the hyperparameter in Table III is based on the accuracy of classification.

The hyperparameters of the imitative DRL unit in the comparison results are listed in Table IV. The selection of the hyperparameter in Table IV is based on the return received by the learned MM policy.

[3]Here RB refers to the Steel Rebar Futures Contract; FU refers to the Fuel Oil Futures Contract; CU refers to the Copper Futures Contract; and AG refers to the Silver Futures Contract.

TABLE III
SRLU HYPERPARAMETERS

Hyperparameter	Distribution
learning_rate	Uniform[0.01,0.05]
num_leaves	UniformInt[10,100]
min_child_weight	LogUniform[1e-5,1e-1]
min_child_samples	UniformInt[2,100]
subsample	Uniform[0.5,1.0]
colsample_bytree	Uniform[0.5,1.0]

TABLE IV
POLICY LEARNING HYPERPARAMETERS

Hyperparameter	Value
Actor learning rate	0.0001
Critic learning rate	0.001
Target network update rate	0.005
Discount factor (γ)	0.999
Target policy noise standard deviation	0.2
Noise smoothing for target policy	0.005
Action space noise standard deviation	0.1
Experience replay buffer size	1000000
Critic update frequency (per step)	2
Gradient clipping threshold	0. 5
Batch size	128
Optimizer	Adam
Hidden sizes	128
Behaviour cloning weight (λ)	1
Behaviour cloning weight decay	0.99
Truncated inventory threshold	250
Truncated inventory penalty weight (η)	2
Compensation bonus weight (β)	0.1

B. Benchmarks

We compare PIMMA with three linear rule-based and one RL-based benchmark strategies:

- **Fixed Offset with Inventory Constraints(FOIC)** Here $FOIC(c)$ [11] refers to the *Fixed Offset with Inventory Constraints* strategy that posts bid (ask) orders at the current best bid (ask) while adhering to the inventory constraint c.
- **Linear in Inventory with Inventory Constraints (LIIC)** $LIIC(a, b, c)$ [11] corresponds to the strategy where a market maker adjusts its quote prices based on inventory - when $-c < z_t < c$, the quote prices of ask (bid) orders at the time t are posted as (18).

$$\begin{cases} ask_t^q = m_t + b \cdot z_t + a \cdot tick_size, \\ bid_t^q = m_t + b \cdot z_t - a \cdot tick_size. \end{cases} \tag{18}$$

If $|z_t| \geq c$, only the orders on the opposite side are posted. Note that $LIIC(a, b, c)$ can be written as a special case of $LTIIC(a, b, c, d = 0)$.

- **LTIIC** We have described LTIIC in Section IV-B1 and we adopt it as the expert in the proposed strategy.
- $\mathbf{RL_{DS}}$ refers to the RL-based single-price level strategy proposed in [10].
- $\mathbf{DRL_{OS}}$ DRL_{os} refers to a RL-based multi-level strategy [20]. DRL_{os} lets the agent to make decisions on whether to retain one unit of volume on each price level while encoding their queue position values. However, it should be noted that this method enforces the same order volume distribution across all price levels.

C. Evaluation Metrics

We adopt several metrics to assess the performance of a market-making strategy:

- **Episodic PnL** is a natural choice to evaluate the profitability of a market-making agent, since there is no notion of starting capital for market making.

$$EPnL_T = \sum_{t=1}^{T} PnL_t. \tag{19}$$

- **Mean Absolute Position (MAP)** accounts for the inventory risk, defined as:

$$MAP_T = \frac{\sum_{t=1}^{T} |z_t|}{\sum_{t=1}^{T} 1 \cdot \mathbb{I}(|z_t| > 0)}. \tag{20}$$

- **Return Per Trade (RPT)** evaluates the agent's capability of capturing the spread. It is normalized across different markets by the average market spread $\overline{\delta^m}$.

$$RPT_T = \left(\frac{\sum_{i \in A_T} p_i^a * n_i^a}{\sum_{i \in A_T} n_i^a} - \frac{\sum_{j \in B_T} p_j^b * n_j^b}{\sum_{j \in B_T} n_j^b} \right) \Big/ \overline{\delta^m}. \tag{21}$$

- **PnL-to-MAP Ratio (PnLMAP)** simultaneously considers the profitability and the incurred inventory risk of a market-making strategy.

$$PnLMAP_T = \frac{EPnL_T}{MAP_T}. \tag{22}$$

D. Comparison Results

For a fair comparison, we tune the hyper-parameters of the baseline methods for their best risk-adjusted return performance. That is, we select the parameters that result in the maximum PnLMAP_T value on the validation dataset for the benchmarks. The comparison results of PIMMA and the benchmark strategies on the four test datasets are given in Table II. Besides, the daily PnLs of these strategies on the FU datasets are depicted in Fig. 3. The comparison results on these four datasets indicate that the proposed approach significantly outperforms the benchmarks.

As demonstrated in Table II, on the RB dataset, the proposed method PIMMA achieves the highest terminal wealth and the best risk-adjusted-return PnLMAP, at the cost of a slightly higher inventory risk compared to the expert LTIIC and the two RL-based benchmarks. Meanwhile, the two multi-price level RL-based agents, PIMMA and DRL_{OS}, outperform the single-price level method RL_{DS}, indicating the superiority of the multi-price level strategy.

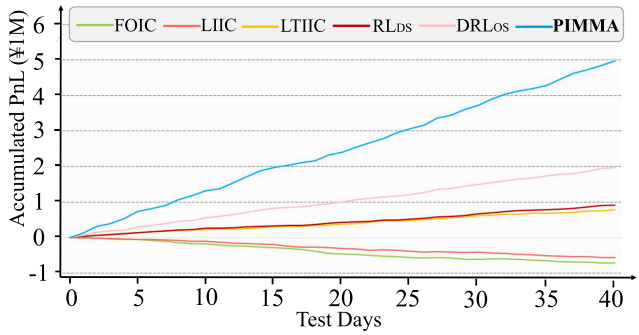

Fig. 3. The accumulated daily PnL of the proposed approach and the benchmarks on the FU dataset.

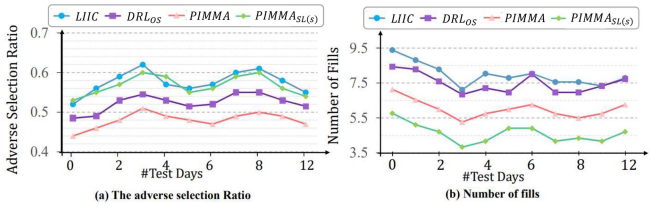

Fig. 4. Adverse_ratio and numbers of fills.

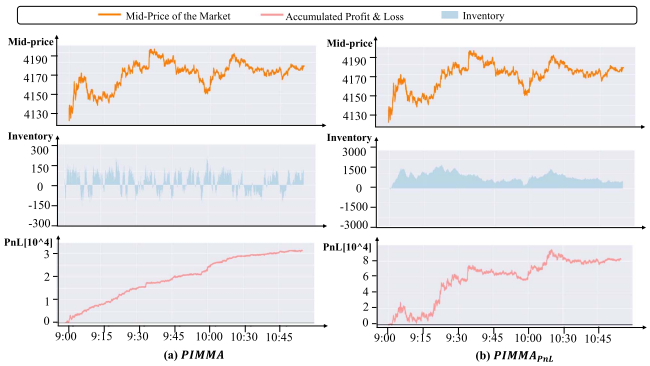

Fig. 5. The intra-day performance of the $PIMMA$ and $PIMMA_{PnL}$ strategy on FU on Jun. 14th, 2022.

On the FU dataset, PIMMA achieves not only the highest terminal wealth, but also the most favorable return-to-risk performance and the most favorable spread-capturing ability, while maintaining the second-lowest MAP. As shown in Fig. 5(a), the proposed approach learns an effective MM strategy: The agent achieves stable dividends with profits (pink line) while maintaining the inventory at an acceptable low level (small area of the blue region) in such a violate market. The inventory constantly fluctuates around zero, indicating that the MM agent behaves nicely as expected. Since the proposed approach does not force the agent to place opposite-side orders to clear its inventory, it is quite appealing that the proposed method achieves automatic inventory control according to the state. In most trading days, the expert strategy LTIIC and the RL-based DRL_{OS} strategy also achieve decent performance. The other two rule-based strategies

TABLE V
VARIATIONS OF STATE REPRESENTATIONS FOR ABLATION STUDY

Variation	quote infos	signal	TCSA	RL	BC
$PIMMA_{BC(0)}$	O	O	O	O	X
$PIMMA_{BC(1)}$	O	O	O	X	O
$PIMMA_{SL(m)}$	O	O	X	O	O
$PIMMA_{SL(s)}$	O	X	O	O	X
$PIMMA_{SL(q)}$	X	O	O	O	O

TABLE VI
COMPARISON RESULTS OF ABLATION STUDY ON FU DATASET

	EPnL[10^3]	MAP[unit]	PnLMAP	SR
$PIMMA_{BC(0)}$	14.67 ± 5.11	85 ± 5	172 ± 57	**2.87**
$PIMMA_{BC(1)}$	8.22 ± 3.70	**51 ± 4**	156 ± 61	2.22
$PIMMA_{SL(m)}$	10.57 ± 8.63	74 ± 41	142 ± 39	1.22
$PIMMA_{SL(s)}$	7.83 ± 3.64	49 ± 5	159 ± 46	2.15
$PIMMA_{SL(q)}$	10.20 ± 9.72	74 ± 47	104 ± 56	1.05
$PIMMA$	**28.10 ±10.27**	103± 15	**274 ± 89**	2.80

without signals (i.e. FOIC and LIIC) fail to make profits on most trading days.

It's difficult for all these methods to optimize MM on CU and AG future markets. The reason might be the lack of market liquidity in these two markets. However, the proposed method still significantly outperforms the benchmarks in terms of the terminal wealth in these markets. PIMMA exhibits favorable performance across other evaluation metrics, solidifying its capability to mitigate liquidity risks. These collective findings indicate the effectiveness of PIMMA. Readers seeking further substantiation can refer to the subsequent ablation studies for additional evidence.

E. Ablation Study

To investigate the effectiveness of the proposed PIMMA, we compare the proposed approach with its five variations including $PIMMA_{BC(0)}$, $PIMMA_{BC(1)}$, $PIMMA_{SL(m)}$, $PIMMA_{SL(s)}$ and $PIMMA_{SL(q)}$. They are described in Table V.

Here $PIMMA_{BC(0)}$ refers to method that trains the policy using TD3 instead of the imitative RL unit, i.e. in (17), $\lambda = 0$, while $PIMMA_{BC(0)}$ refers to the method only using BC, in (17), $\lambda = 1$. $PIMMA_{SL(m)}$ denotes removing the market representation learning part TCSA from the proposed method; $PIMMA_{SL(s)}$ denotes the strategy without using BC and trend signal; $PIMMA_{SL(q)}$ denotes not including information of the resting quotes in the private variables $\boldsymbol{s}_t^p$. As shown in Table VI, the performance of the proposed method is superior to the other variant methods.[4] After that, we vary the reward functions to prove the goodness of our proposed reward function for PIMMA in Section V-E3.

1) Effectiveness of State Representation Learning Unit: The comparison results of PIMMA and other variations indicate that introducing multi-granularity signals as auxiliary observations is a promising way to improve the performance of the learned strategy. The multi-granularity predictive signals provide effective information about the market conditions, and enable a more flexible trade-off between spread-capturing and trend-chasing.

The comparison results of $PIMMA_{SL(\cdot)}$ and PIMMA indicate that introducing multi-granularity signals as auxiliary observations is quite important to improve the performance of the learned strategy. To further illustrate the role of auxiliary signals in mitigating adverse selection risk, we calculated the adverse selection ratio as:

$$adverse\_ratio = \frac{\#\text{adverse fills in the last time interval}}{\#\text{fills in the last time interval}}. \tag{23}$$

Here, "adverse fills" refer to limit bid (ask) orders that are executed shortly before the best bid (ask) price moves down (up) [20]. If the best bid price has gone down, it might have been better to wait for the next bid price level. We also show the number of transactions (*#T*) in Fig. 4 to examine the impact of quote information on liquidity risk. Based on the findings from Fig. 4, we can conclude that multi-granularity predictive signals are useful in avoiding adverse selections. They provide effective information about market conditions and enable a more flexible trade-off between capturing spreads and chasing trends. The order quote information also contributes to an increased number of fills by reducing frequent cancellations and maintaining queue positions.

2) Effectiveness of the Imitative RL Unit: The empirical comparison between $PIMMA_{BC(\cdot)}$ and PIMMA demonstrates the crucial requirement of extracting additional knowledge from experts and exploring efficiently, particularly in challenging tasks. As MM evolves into a complex high-frequency trading problem, which is characterized by noisy data, it becomes difficult for RL agents to identify an effective trading mode and consistently follow it during the early stages of training. Utilizing the behavior cloning objective during training assists the agent in achieving positive rewards and drawing valuable lessons from these experiences.

3) Effects of Reward Functions: To investigate the effects of different reward functions, we also have conducted a set of experiments using three variants of PIMMA that are trained with different rewards: The $PIMMA_{PnL}$ method trains PIMMA using the PnL reward, i.e. $\eta, \beta = 0$; The $PIMMA_{PnL+C}$ method trains PIMMA with the combination of the PnL and compensation reward, i.e. $\eta = 0, \beta > 0$; Similarly, The $PIMMA_{PnL+IP}$ method trains PIMMA with the combination of the PnL and truncated inventory penalty reward, i.e. $\eta > 0, \beta = 0$. We select the hyper-parameters that result in the maximum PnLMAP value on the validation dataset for these models. The results on the FU

TABLE VII
THE PERFORMANCE OF PIMMA TRAINED WITH DIFFERENT REWARD FUNCTIONS ON FU

	EPnL[10^3]	MAP[unit]	PnLMAP	#T
$PIMMA_{PnL}$	58.76 ± 94.43	2156 ± 655	31 ± 48	4.43 ± 0.94
$PIMMA_{PnL+C}$	42.86 ± 123.04	20414 ± 465	27 ± 68	4.85 ± 1.09
$PIMMA_{PnL+IP}$	**73.07 ± 53.83**	756 ± 289	90 ± 46	4.42 ± 0.96
PIMMA	28.10 ± 10.27	**103 ± 15**	**274 ± 89**	**5.15 ± 1.19**

[4]Here SR denotes the Sharpe ratio, see A for further explanation

dataset are listed in Table VII. Here the metric #T refers to the number of transactions normalized by the episode length.

The experimental results validate the effectiveness of the components of the proposed reward formulation. As shown in Table VII, the $PIMMA_{PnL}$ strategy tends to have the largest inventory risk exposure. We depict an example of the intra-day performance of the $PIMMA_{PnL}$ policy in Fig. 5(b) to analyze the reasons. We observe that the $PIMMA_{PnL}$ agent learns to chase trends through maintaining a large inventory (> 1000). This results in poor out-of-sample performance with large variance. Therefore, the truncated inventory penalty term is very important to deactivate blind trend-chasing.

The $PIMMA_{PnL+C}$ strategy also suffers from a high inventory risk but has a larger number of transactions #T compared to $PIMMA_{PnL}$. The $PIMMA_{PnL+IP}$ strategy achieves the largest average terminal wealth and the return per trade metric while having the lowest #T, which is unfavorable for risk-averse market makers. The strategy trained with the proposed PnL+IP+C reward significantly improves the return-to-risk performance with the lowest MAPs, as well as a larger #T, compared to the $PIMMA_{PnL+IP}$ strategies. Although having the lowest average terminal wealth, the proposed PIMMA strategy acts very stably and might be the most favorable policy among these four policies for a risk-averse market maker. Besides, note that the proposed strategy has the largest #T, it could receive more compensation from the exchange.

F. Time Cost

The experiments are carried out on a server with a single NVIDIA GeForce RTX 3090 with 24GB RAM, 64GB system RAM, and Intel(R) Xeon(R) Platinum 8350C CPU @2.60GHz with 16 cores. The training times (in minutes) for the proposed approach PIMMA are as follows: 190 (50 for SRLU + 140 for policy learning). Despite the additional pre-training time of 50 minutes for SRLU, PIMMA enhances the efficiency of policy learning. In this work, the lightGBM model is executed on a CPU with an inference time of 2 to 5 million seconds. The TCSA is executed on a GPU with an inference time of 3 to 4 million seconds. The total inference time of the PIMMA algorithm is 8 to 15 million seconds. Therefore, the latency of the proposed approach remains low, effectively meeting the requirements of most actual transaction scenarios.

VI. Related Work

Market-making strategies have been studied extensively across disciplines, including finance, economics, and machine learning. Classic approaches [1], [2], [3], [34] in the finance literature consider market making as a stochastic optimal control problem that can be solved analytically. For example, the Avellaneda—Stoikov (AS) model [1] uses the Hamilton-Jacobi-Bellman equations to derive closed-form approximations to the optimal quotes of a market maker. However, such models are typically predicated upon a set of strong assumptions and employ multiple parameters that need to be laboriously calibrated on historical market data. In contrast, the proposed approach adopts the RL techniques, thus enabling learning directly from data in a model-free fashion.

Another line of prominent approaches focus on Agent-Based Modeling (ABM) [16], [35], [36], [37]. However, these MM methods are typically evaluated in simulated markets without using real market data. On the contrary, the proposed approach is devoted to developing realistic market-making strategies. Methods evolve over time. As neural networks gain their popularity in handling large amounts of high-dimensional data, [38] presents an approach based on a Recurrent Neural Network (RNN) to forecast the opening price, the closing price, and the difference between them. Ref. [5] also proposes a temporal-aware neural bag-of-features model to learn for predicting mid-price movements using LOB data.

Reinforcement learning (RL) is a powerful machine learning approach with adaptability to complex environments, generalization ability, and autonomous learning capability. Lately, the popularity of deep reinforcement learning (DRL) has grown immensely due to a series of outstanding successes in various complex sequential decision-making tasks. Recent years have witnessed a strong popularity of (D)RL in the field of quantitative trading, including portfolio management [39], option hedging [40], optimal execution [21], and market making [14], [19], [20], [41], [42], [43], [44]. Several researchers employ non-deep RL algorithms to solve market-making. Ref. [10] developed a market-making agent using temporal-difference RL. The authors experimented with different reward functions and state components to exhibit superior risk-adjusted performance. Furthermore, non-deep approaches include additional features such as latency effects [45], state space representation consisting of technical indicators [46], and echo state networks (ESNs) [47]. As for DRL methods for market making, [13] provided an end-to-end market-making framework using proximal policy optimization (PPO) and advantage actor—critic (A2C) to train the agent. Gueant and Manziuk [14] addressed multi-asset MM in over-the-counter markets and proposed a model-based actor—critic-like algorithm for approximating optimal quotes in an AS-like multi-asset model. Ref. [48] considered an MM trader simultaneously trading in the dark and lit pools of an exchange. Gašperov and Kostanjčar [11] proposed a non-gradient DRL framework for MM using trend signals as market representations, with a focus on the interpretability of the learned controls. The main difference between the proposed SRLU in PIMMA and this method is that in addition to signals, we train a TCSA network to learn useful representations from the rewards. Ref. [12] introduced a new action representation for MM and made some efforts towards establishing a high-fidelity order-driven market.

VII. Conclusion and Future Work

In this paper, we propose PIMMA, a DRL-based approach incorporated with imitation learning techniques and predictive representation learning to deal with the challenge of high-frequency MM. To leverage the RL techniques to address MM, we introduce a novel action space and reward formulation. At first, PIMMA pre-trains an SL-based prediction model to

TABLE VIII
THE REST COMPARISON RESULTS OF THE PROPOSED METHOD AND THE BENCHMARKS

	RB		FU		CU		AG	
	#T	Sharpe Ratio	#T	Sharpe Ratio	#T	Sharpe Ratio	#T	Sharpe Ratio
FOIC	6.52 ± 1.61	0.74	**7.44 ± 1.65**	-0.84	**5.39 ± 1.43**	-1.20	**4.47 ± 1.25**	-1.68
LIIC	**6.52 ± 1.57**	0.68	7.26 ± 1.61	-1.03	5.21 ± 1.39	-1.63	4.24 ± 1.17	-1.48
LTIIC	4.34 ± 1.17	1.88	4.98 ± 1.25	**3.13**	3.88 ± 1.01	-1.06	3.34 ± 0.95	-1.43
RL_{DS}	5.19 ± 1.31	**2.66**	5.92 ± 1.36	1.36	4.52 ± 1.31	-1.16	3.66 ± 0.65	-1.07
DRL_{OS}	4.34 ± 1.17	2.22	4.90 ± 1.23	0.80	3.79 ± 1.09	-2.36	3.41 ± 1.00	-1.02
PIMMA	4.05 ± 1.06	1.81	5.15 ± 1.19	2.74	4.22 ± 1.30	**-0.48**	3.65 ± 0.79	**-0.72**

generate multi-granularity trend signals as effective auxiliary observations to describe the state. Additionally, PIMMA utilizes a TCSA network to handle the temporal and spatial relationships among multi-granularity market features. Through abstracting trading knowledge from a sub-optimal expert meanwhile interacting with the environments, PIMMA explores the state and action spaces efficiently and learns a practical MM strategy. Experiments on four future markets demonstrate that PIMMA outperforms the benchmarks, and further ablation studies verify the effectiveness of the components in the proposed method.

As it is important for market makers to preempt order priorities in inactive markets, we would like to take order cancellations into account and investigate the automatic MM strategies for less liquid markets in future work. Furthermore, including more market information into the market state, such as the order flow data at the micro level and macroeconomic indicators at the macro level, would be an interesting future direction as well. Another promising future direction is to develop more advanced action space for MM, which supports iceberg orders, stop-loss orders, etc.

APPENDIX A
ADDITIONAL EXPERIMENTAL RESULTS

Due to the space limitation of paper writing, we put some experimental results here. Table VIII list the comparison of PIMMA and benchmarks in terms of other two financial criteria. Here $\#T$ denotes the number of transactions while the Sharpe ratio is a mathematical expression of the insight that excess returns over a period of time may signify more volatility and risk, rather than investing skill. It was proposed by Economist William F. Sharpe in 1966, with its formulation:

$$SharpeRatio = \frac{R_p - R_f}{\sigma_p} \tag{24}$$

where: $R_p = $ *return of portfolio*,
$R_f = $ *risk-free rate*,
$\sigma_p = $ *standard deviation of the portfolio's excess return*.
The Sharpe ratio can be calculated using our metric PnL:

$$SR = Mean(PnL)/Std(PnL). \tag{25}$$

A higher Sharpe ratio indicates that the investment distribution achieves higher returns per unit of risk taken. As shown in Table VIII, the proposed PIMMA method behaves more decently than others in the two harder tasks - CU and AG, while still dealing well in RB and FU markets.

TABLE IX
THE PRECISION OF THE PRE-TRAINED LARA MODELS

Labels	1-minute	5-minute	15-minute	30-minute
RB	0.6712	0.6324	0.5921	0.5316
FU	0.6503	0.6213	0.5834	0.5471

APPENDIX B
RESULTS OF THE PREDICTION MODEL

The prediction precision of the pre-trained LightGBM models on validation datasets is given in Table IX.

Notice that it is hard to obtain an accurate long-term prediction signal, therefore the long-term trend chasing is a more risky trading behavior compared to the short-term spread capturing.

REFERENCES

[1] M. Avellaneda and S. Stoikov, "High frequency trading in a limit order book," *Quantitative Finance*, vol. 8, pp. 217–224, 2008.
[2] O. Guéant, C.-A. Lehalle, and J. Fernandez-Tapia, "Dealing with the inventory risk: A solution to the market making problem," *Math. Financial Econ.*, vol. 7, no. 4, pp. 477–507, Sep. 2012.
[3] L. R. Glosten and P. R. Milgrom, "Bid, ask and transaction prices in a specialist market with heterogeneously informed traders," *J. Financial Econ.*, vol. 14, pp. 71–100, 1985.
[4] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3815895
[5] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 6, pp. 774–785, Dec. 2020.
[6] H. Zhao, C. Dong, J. Cao, and Q. Chen, "A survey on deep reinforcement learning approaches for traffic signal control," *Eng. Appl. Artif. Intell.*, vol. 133, 2024, Art. no. 108100.
[7] G. Papadopoulos et al., "Deep reinforcement learning in service of air traffic controllers to resolve tactical conflicts," *Expert Syst. Appl.*, vol. 236, 2024, Art. no. 121234.
[8] L. d. A. Takara, A. A. P. Santos, V. C. Mariani, and L. d. S. Coelho, "Deep reinforcement learning applied to a sparse-reward trading environment with intraday data," *Expert Syst. Appl.*, vol. 238, p. 121897, 2024.
[9] D. Li, F. Zhu, J. Wu, Y. D. Wong, and T. Chen, "Managing mixed traffic at signalized intersections: An adaptive signal control and CAV coordination system based on deep reinforcement learning," *Expert Syst. Appl.*, vol. 238, 2024, Art. no. 121959.
[10] T. Spooner, J. Fearnley, R. Savani, and A. Koukorinis, "Market making via reinforcement learning," in *Proc. AAMAS 2018: Proc. 17th Int. Conf. Auton. Agents MultiAgent Syst.*, 2018, pp. 434–442.
[11] B. Gašperov and Z. Kostanjčar, "Market making with signals through deep reinforcement learning," *IEEE Access*, vol. 9, pp. 61611–61622, 2021.
[12] J. Jerome, G. Palmer, and R. Savani, "Market making with scaled beta policies," in *Proc. 3rd ACM Int. Conf. AI Finance*, 2022, pp. 214–222.
[13] J. Sadighian, "Deep reinforcement learning in cryptocurrency market making," Trading and Market Microstructure, 2019, *arXiv:1911.08647*.

[14] O. Gu'eant and I. Manziuk, "Deep reinforcement learning for market making in corporate bonds: Beating the curse of dimensionality," *Appl. Math. Finance*, vol. 26, pp. 387–452, 2019.
[15] Á. Cartea, S. Jaimungal, and J. S. Penalva, *Algorithmic and High-Frequency Trading*. New York, NY, USA: Cambridge Univ. Press, 08 2015.
[16] J. Jumadinova and P. Dasgupta, "A comparison of different automated market-maker strategies," in *Proc. 12th Workshop Agent-Mediated Electron. Commerce*, 2010, pp. 141–154.
[17] Y.-S. Lim and D. Gorse, "Reinforcement learning for high-frequency market making," in *Proc. 26th Eur. Symp. Artif. Neural Netw., 2018*, Bruges, Belgium, 2018, pp. 521–526. [Online]. Available: http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-50.pdf
[18] J. D. Abernethy and S. Kale, "Adaptive market making via online learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2013.
[19] T. Chakraborty and M. Kearns, "Market making and mean reversion," in *Proc. ACM Conf. Econ. Computation*, 2011, pp. 307—314.
[20] G. Y. Chung, M. Chung, Y. Lee, and W. C. Kim, "Market making under order stacking framework: A deep reinforcement learning approach," in *Proc. 3rd ACM Int. Conf. AI Finance*, 2022, pp. 223–231. [Online]. Available: https://api.semanticscholar.org/CorpusID:253022156
[21] Y. Fang et al., "Universal trading for order execution with oracle policy distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 107–115.
[22] R. Mendonca, A. Gupta, R. Kralev, P. Abbeel, S. Levine, and C. Finn, "Guided meta-policy search," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019.
[23] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE Int. Conf. Robot. Automat.*, 2017, pp. 6292–6299.
[24] W. Sun, J. A. Bagnell, and B. Boots, "Truncated horizon policy search: Combining reinforcement learning & imitation learning," in *Proc. Int. Conf. Learn. Representations*, 2018.
[25] J. Fernandez-Tapia, "Modeling, optimization and estimation for the on-line control of trading algorithms in limit-order markets," Ph.D. dissertation, Université Pierre et Marie Curie, Paris, France, 2015.
[26] M. Avellaneda and S. Stoikov, "High-frequency trading in a limit order book," *Quantitative Finance*, vol. 8, pp. 217–224, 2008.
[27] W. Huang, C.-A. Lehalle, and M. Rosenbaum, "Simulating and analyzing order book data: The queue-reactive model," *J. Amer. Stat. Assoc.*, vol. 110, no. 509, pp. 107–122, 2015, doi: 10.1080/01621459.2014.982278.
[28] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.
[29] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, in Series NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6000–6010.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[31] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 12, 1999.
[32] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*. PMLR, 2018, pp. 1587–1596.
[33] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*. PMLR, 2014, pp. 387–395.
[34] Y. Amihud and H. Mendelson, "Dealership market: Market-making with inventory," *J. Financial Econ.*, vol. 8, pp. 31–53, 1980.
[35] S. Das, "A learning market-maker in the Glosten–Milgrom model," *Quantitative Finance*, vol. 5, pp. 169–180, 2005.
[36] S. Das, "The effects of market-making on price dynamics," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2008, pp. 887–894.
[37] E. Wah and M. P. Wellman, "Welfare effects of market making in continuous double auctions," *J. Artif. Intell. Res.*, vol. 59, pp. 613–650, 2015.
[38] Y.-F. Lin, T.-M. Huang, W.-H. Chung, and Y.-L. Ueng, "Forecasting fluctuations in the financial index using a recurrent neural network based on price features," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 5, pp. 780–791, Oct. 2021.
[39] H. Niu, S. K. Li, and J. Li, "MetaTrader: An reinforcement learning approach integrating diverse policies for portfolio optimization," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 1573—1583.
[40] J. Cao, J. Chen, J. Hull, and Z. Poulos, "Deep hedging of derivatives using reinforcement learning," *J. Financial Data Sci.*, vol. 3, pp. 10–27, 2020.
[41] N. T. Chan and C. R. Shelton, "An electronic market-maker," MIT AI Lab, Tech. Rep., pp. 1–43, 2001.
[42] Á. Cartea, R. F. Donnelly, and S. Jaimungal, "Enhancing trading strategies with order book signals," *Appl. Math. Finance*, vol. 25, no. 1, pp. 1–35, 2015.
[43] Y. Zhong, Y. Bergstrom, and A. R. Ward, "Data-driven market-making via model-free learning," in *Proc. 29th Int. Conf. Int. Joint Conferences Artif. Intell.*, 2020, pp. 4461–4468.
[44] Y. Patel, "Optimizing market making using multi-agent reinforcement learning," 2018, *arXiv:1812.10252*.
[45] X. Gao and Y. Wang, "Electronic market making and latency," 2018, *arXiv:1806.05849*.
[46] K. Lokhacheva, D. I. Parfenov, and I. P. Bolodurina, "Reinforcement learning approach for market-maker problem solution," in *Proc. Int. Session Factors Regional Extensive Develop.*, 2020, pp. 256–260.
[47] A. G. Hart, K. R. Olding, A. M. G. Cox, O. Isupova, and J. H. P. Dawes, "Echo state networks for reinforcement learning," 2021, *arXiv:2102.06258*.
[48] B. Baldacci, I. Manziuk, T. Mastrolia, and M. Rosenbaum, "Market making and incentives design in the presence of a dark pool: A deep reinforcement learning approach," 2019, *arXiv:1912.01129*.

Siyuan Li received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2022. She is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. Her research interests include deep reinforcement learning, multi-agent learning, and their applications in robotics.

Yafei Chen received the master's degree from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2011. She is currently working toward the Ph.D. degree in artificial intelligence with the Harbin Institute of Technology, Harbin, China. Her research interests include reinforcement learning and multi-agent learning.

Hui Niu received the B.S. degree from the School of Mathematical Science, Zhejiang University, Hangzhou, China, in 2017. She is currently working toward the Ph.D. degree with the Institute for Interdisciplinary Information Science, Tsinghua University, Beijing, China, under the supervision of Prof. Jian Li. Her research interests include deep reinforcement learning, machine learning, and AI finance.

Jiahao Zheng received the bachelor's degree from South China Agricultural University, Guangzhou, China, and the master's degree from the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He is currently working with AI Finance & Deep Learning Department, International Digital Economy Academy. His research interests include remote sensing image processing, video anomaly detection, multilevel optimization with deep reinforcement learning, and AI finance.

Zhouchi Lin received the B.Eng. degree from Sun Yat-sen University, Guangzhou, China, in 2012, the M.Eng. degree from Cornell University, Ithaca, NY, USA, in 2013, and the Ph.D. degree from the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, under the supervision of Prof. S. C. Chan. He is currently working with AI Finance and & Deep Learning Department, International Digital Economy Academy. His research interests include image processing, computer vision, pattern recognition, and AI finance.

Jian Li received the B.Sc. degree from Sun Yat-sen Zhongshan University, Zhongshan, China, and M.Sc. degree in computer science from Fudan University, Shanghai, China, and the Ph.D. degree from the University of Maryland, College Park, MD, USA. He is currently an Associate Professor with the Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China, headed by Prof. Andrew Yao. His main research interests include theoretical computer science, machine learning, and databases. He is interested in learning theory, and applying machine learning to financial applications. He coauthored several research papers that have been published in major computer science conferences and journals. He was the recipient of the Best Paper Awards at VLDB 2009 and ESA 2010 the Best Newcomer Award with ICDT 2017.

Jian Guo received the undergraduate degree (major in mathematics and applied mathematics) from Tsinghua University, Beijing, China, and the Ph.D. degree (major in machine learning) from the Department of Statistics, University of Michigan,, Ann Arbor, MI, USA. He is currently the Executive President of the International Digital Economy Academy, and the ChiefScientist of AI finance and deep learning with IDEA research. His research interests include deep learning and reinforcement learning for quantitative trading.

Zhen Wang (Fellow, IEEE) received the Ph.D. degree from Hong Kong Baptist University, Hong Kong, China, in 2014. He is currently a Distinguished Professor with Northwestern Polytechnical University, Xi'an, China. He has authored or coauthored more than 100 scientific papers, his total citations are more than 16000 times and H-index is 58. His research interests include artificial intelligence complex networks, complex systems, Big Data, evolutionary game theory, behavioral decision-making, and cognition.