An Ensemble Method of Deep Reinforcement Learning for Automated Cryptocurrency Trading

***Abstract*—We propose a novel ensemble method for deep reinforcement learning based trading strategies that improves generalization in cryptocurrency trading. The ensemble method utilizes a mixture distribution to provide a distributional way of combining multiple stochastic polices from models that perform well in different trading periods instead of simply averaging the model outputs. The proposed method is tested on a 4-year historical period that encompasses various cryptocurrency market conditions, including bullish and bearish markets. The results empirically demonstrate the effectiveness of the ensemble method by its robust performance compared to a deep reinforcement learning benchmark and a passive investments strategy measured by annualized returns and risk-adjusted returns.**

***Index Terms*—algorithmic trading, deep reinforcement learning, cryptocurrency trading, ensemble method**

I. Introduction

Cryptocurrencies have emerged as an alternative opportunity for investors to diversify their portfolio. The cryptocurrency market is vastly different from the traditional financial market, characterized by its large fluctuations and the lack of the fundamental data to conduct valuation and risk factor analyses, making it particularly difficulty to reliably predict prices of cryptocurrencies. Moreover, the cryptocurrency market is open around the clock and the trading activities are affected by the blockchain transactions, making the intraday trading strategies for cryptocurrencies fundamentally different from those for traditional assets. Therefore, it is important to understand the dynamics of the cryptocurrency market and design intraday trading strategies for cryptocurrencies for investors to maximize the returns while minimizing the risks due to large volatility.

The recent breakthrough of deep reinforcement learning (DRL) algorithms has demonstrated its potential to learn good policies for complex control tasks in stochastic environments [1], [2]. The success has inspired researchers to leverage the power of the DRL algorithms to find automated trading strategies. We adopt the reinforcement learning paradigm to develop intraday trading strategies that dynamically adjust the portfolio composition based on the observation of the prices and the current holdings. In the reinforcement learning paradigm, we apply DRL algorithms to optimize the long term return instead of myopically maximizing the immediate return to avoid unnecessary turnovers in the highly volatile cryptocurrency market. The DRL agent learns from the rewards of expected future returns for each trading decision made as we roll out a large number of episodes using historical cryptocurrency prices.

The attempts of applying reinforcement learning in trading show some promising progress with evidence of profitable trading strategies in backtesting [3]–[7]. Nevertheless, the nonstationary and chaotic nature of financial markets poses severe challenges to the generalization performance of DRL trading strategies on out-of-sample data, undermining their potential of deployment in live trading. The generalization of DRL trading strategies is an active research area. Numerous attempts have been made to adapt DRL algorithms to accommodate dynamic financial markets, including improvements of feature extraction [8], [9], noise reduction in reward signals [10], and ensemble methods [11], [12]. While some existing works simplify the trading environment by allowing only finite actions or a single asset to study improvements in generalization, we explore in a realistic setting that supports continuous actions of trading multiple assets simultaneously while focusing on the out-of-sample trading performance.

The challenge for a reliable generalization of DRL trading strategies is two-fold. The noisy nature of the market hinders the extraction of informative patterns from the historical price and volume data, while the nonstationary property amplifies the overfitting issue in training and leads to unreliable performance on out-of-sample data. On the feature extraction side, we employ the long short-term memory [13] (LSTM) module to capture the temporal relationships in sequences of historical observations and enrich the features by technical indicators commonly utilized by traders; on the generalization side, we propose a novel mixture distribution policy to effectively ensemble the best performance models from multiple validation periods. By constructing a mixture distribution, we enable a distributional way of combining stochastic polices from multiple models rather than simply averaging the model outputs. By tracking the moving average of the validation metrics, we stabilize the evaluation of the model during training and retain models that exhibit persistently high performance instead of those with spurious sudden spikes in performance. To address the difficulty of training with nonstationary financial time series, we break down the 4-year test data into granular periods and use a rolling window to repeat the training and testing procedure.

The enhancement of the generalization performance is measured by the annualized return and the risk-adjusted returns on 4-year out-of-sample data, and the robustness of the proposed method is demonstrated by a distributional view of the performance on granular test periods consisting of various market conditions. The granular performance evaluation helps mitigate the critical issue of false positive reporting of

trading strategies that are highly overfitted to specific market conditions despite impressive cumulative returns. We compare our proposed method with the FinRL-Meta [14] framework that supports continuous trading actions of a cryptocurrency portfolio as our DRL strategy benchmark, and a buy and hold strategy as our passive investment benchmark. Our ensemble method outperforms both benchmarks in annualized and risk-adjusted returns on 4-year out-of-sample data, showing a promising DRL approach to train reliable automated trading strategies for intraday cryptocurrency portfolio trading.

Our contributions are as follows.

- We propose a novel mixture distribution policy for DRL trading strategies to effectively ensemble the best performance models on multiple validation periods, and achieve high returns on out-of-sample data.
- We design a model selection schema that tracks the moving average of the validation performance to select robust models during training.
- We provide a distributional view of the out-of-sample performance on granular periods to demonstrate the robustness of DRL strategies in evolving market conditions.

II. Related Work

The task of portfolio trading can be formulated as a dynamic programming problem where at each time step, the portfolio manager makes a trading decision to maximize the expected future return. Many attempts have been made to apply DRL algorithms to find profitable trading strategies. Some works simplify the problem by discretizing the action space [3], [8], [9], [15]–[20]. They use DRL algorithms for a discrete action space where the buy or sell amount of each trade is restricted to a finite set of predefined values. We relax the assumption and study continuous actions of trading a portfolio, which is closer to the realistic trading environment.

Among the works of reinforcement learning method of trading multiple assets with continuous action space, [21] use a reward scaling method that incorporates modern portfolio theory and outperforms the momentum strategies in future contracts market, while we explore trading in the more volatile cryptocurrency market. [22] focus on the explanability of DRL strategies, while our work is focused on the generalization performance. FinRL-Meta [14] proposes a generic framework for trading portfolios in different markets with continuous actions. We use models trained by FinRL-Meta as a benchmark to demonstrate the superiority of our proposed ensemble method.

We supplement the literature on the important issue of reliable generalization of DRL trading strategies in changing market conditions. [11] design an ensemble method that combines models trained by three DRL algorithms to find optimal trading strategies, which incurs high computational cost. We identify the issue that hinders generalization of a model to be overfitting to specific market conditions and resolve poor performance by ensembling models that perform well on different validation periods without training multiple models. [23] address the nonstationarity issue in trading a foreign exchange pair. Their procedure consists of training a set of reinforcement learning experts on different market regimes and averaging the discrete action outputs from each expert. Our method provides a distributional way of combing multiple stochastic policies for continuous actions for portfolio trading. [24] design a generative adversarial market simulator to improve the generalization of DRL trading strategies. [25] adopt a rule-based mechanism according to the market environment that explicitly controls the actions taken by the trading strategy for reliable generalization. Our work utilizes the power of DRL algorithms without additional expert knowledge. [26] take an explicit approach to address the overfitting issue of DRL trading strategies. They add a module of hypothesis testing that rejects overfitted agents to ensure higher chance of good performance on the out-of-sample. Our ensemble method implicitly addresses the overfitting issue by effectively combining models that exhibit good performance on different periods with a novel mixture distribution policy.

Our work is in line with the emerging trend of literature on DRL trading in the cryptocurrency market. Compared with the traditional financial markets, the cryptocurrency market is characterized by its high volatility and fluctuation without reliable fundamental analysis tools. Researchers attempt to find robust DRL strategies amidst these additional challenges for cryptocurrencies. On trading a single cryptocurrency, [27] study trading points recommendation by DRL and show profitability; [28] apply a double deep Q-learning algorithm to train the model for discrete actions, with rules to stop losses and guarantee gains; [29] explore using a multi-objective algorithm to improve trading performance for a single asset. On trading a cryptocurrency portfolio, [4] train a convolutional neural network to learn portfolio weights as actions; [30] experiment adding a self-attention layer to the DRL model architecture. We use LSTM as the feature extraction module and focus on the ensemble method for DRL trained models. A DRL approach has also been taken to study cryptocurrency market making using the order book data [31].

There have been attempts of using alternative data or approaches to improve upon existing DRL strategies. Some [15], [17], [32], [33] augment the data by market sentiments and show improvement of performance, while we aim to utilize only the technical data to find profitable strategies; [34] propose a supervised learning approach to trading a single asset; [18] obtain promising results of a risk curiosity-drive learning framework. Our work demonstrates the power of DRL algorithms in training reliable trading strategies when a novel ensemble method is adopted.

III. Background

A. MDP Formulation

We formulate the problem of trading a portfolio of cryptocurrencies as a Markov Decision Process (MDP). The objective is to maximize the return at the end of the trading period through a sequence of decisions of buying, selling or holding each asset at each time step $t \leq T$. We assume that the market is liquid enough with negligible bid ask spread.

The formulation of MDP and the constraints are described as follows, where D is the number of cryptocurrencies.

State space: At time t, the state $s_t = [P_t, I_t, H_t, b_t]$; P_t represents the open, high, low, close prices and trading volume (OHLCV) of time t, and I_t consists of additional technical indicators for each cryptocurrency; $H_t \in \mathbb{R}^D, H_t \geq 0$ denotes the holding of each asset and $b_t \geq 0$ denotes the balance in USD on the account.

Action Space: At time t, the agent takes an action $a_t \in \mathbb{R}^D$ that changes the holding of the cryptocurrency d by $a_{t,d}$, subject to constraints of non-negative holdings and non-negative balance on the account. The agent takes actions according to a policy $\pi_t = \pi(s_t)$ that maps the states to the probability of selecting the actions. Additionally, the trade size is scaled by a predefined value $hmax_d$ such that $|a_{t,d}| \leq hmax_d$ for all $d \in [D]$.

Transition: $\Pr(s_{t+1}|s_t, a_t)$ denotes the probability of transitioning from state s_t to s_{t+1} by taking action a_t. The trading environment determines P_{t+1} and I_{t+1}, and the action modifies H_{t+1}. The balance b_{t+1} is adjusted after executing the trades at close prices at time t. A real number reward r_t is collected after the transition.

B. Technical Indicators

We use the following technical indicators to supplement the OHLCV data as additional features in the states. Technical indicators are widely adopted by traders to detect trend and momentum in financial time series and provide important guidance for their trading decisions.

- Simple moving average (SMA) tracks the price trend using the arithmetic average of the prices at p recent time steps.
- Relative strength index (RSI) is a momentum oscillator in the range of $[0, 100]$ that is used to validate the price trends. An RSI below 30 signals an oversold condition while an RSI above 70 is usually considered an overbought condition.
- Commodity channel index (CCI) is an unbounded oscillator that measures the difference between the current price and the average of past prices in a typical lookback window of 20 time periods. CCI increasing above 100 can indicate an emerging bullish trend while CCI dropping below -100 can signal a new bearish trend.
- Moving average convergence/divergence (MACD) is a momentum indicator that calculates the difference between the exponential moving averages of 26 periods and 12 periods.
- Average true range (ATR) is a non-negative volatility indicator that measures the average price movement range. A larger value indicates a larger volatility.
- Average directional index (ADX) is a trend strength indicator in the range of $[0, 100]$. It signals the presence of a strong trend in either direction when ADX is above 25.

C. Long Short-Term Memory Network

The LSTM architecture is designed to address the vanishing and exploding gradient issues of recurrent neural networks [13]. The model uses gates to control the read and write of the memory and state cells, so that the model is capable of learning long-term dependencies in a sequence. We use LSTM to extract the patterns in a sequence of observations.

D. Proximal Policy Optimization

We use the Proximal Policy Optimization (PPO) algorithm [35] to train the DRL agent to maximize the discounted return $R = \mathbb{E}\sum_t \gamma^t r_t$, where the reward r_t is the change in the portfolio value at time step t of an episode and $\gamma \leq 1$ is the discount factor. An episode extends from the beginning to the end of the trading period. PPO is an actor-critic algorithm where the actor generates a policy compatible with continuous action spaces, and the critic learns the value of the current state. PPO is built on the trust region policy optimization algorithm and uses a clipped surrogate objective function to encourage conservative policy updates. It achieves state-of-the-art or comparable performance while enjoying the benefit of a relatively easy implementation.

IV. Methodology

A. Model Selection

We propose a model selection schema based on multiple validation sets that consist of different market conditions. We randomly select K in-sample validation periods and evaluate the model performance on each validation set. In-sample validation may exacerbate overfitting in supervised learning, but for reinforcement learning, the issue can be less severe. The state vectors differ in the holdings of assets and account balance for the same period visited in training and validation stages, which leads to potentially different policy outputs and subsequent trajectories. Since the training performance is measured on the entire training period while the test performance is measured on much shorter periods, we use validation periods of the same length as the test periods to provide a more accurate assessment of the model performance over the testing time horizon. Additionally, we track the moving average of validation returns and select models with the highest smoothed return. This model selection method implicitly favors models that are robust to weight perturbations by optimization steps and demonstrate consistently high profitability.

B. Ensemble Policy

We model the stochastic policy by a $\tanh$ transformed Gaussian distribution $TanhGaussian(\mu, \Sigma)$ where Σ is a $D \times D$ diagonal matrix with σ_d^2 on the diagonal. The LSTM model outputs μ_d, σ_d for each $d \in [D]$. Sampling from a $\tanh$ transformed Gaussian distribution can be interpreted as sampling an auxiliary value $\tilde{a}_d$ from a Gaussian distribution $\mathcal{N}(\mu_d, \sigma_d^2)$, applying $\tanh$ to map the sampled value to the interval of $(-1, 1)$, and scaling the result by a predefined factor $hmax_d$. The sampled action satisfies the constraints $|a_d| < hmax_d$.

For evaluating the model performance on out-of-sample periods, we propose an ensemble policy using a novel mixture of $\tanh$ transformed Gaussian distributions to effectively combine the stochastic policies from K models, where each model achieves the highest smoothed return on one of the K validation sets. At each time step t of the test episode, the agent takes an action according to the ensemble policy $\pi_t(a) = \frac{1}{K}\sum_{k=1}^{K} TanhGaussian_k(a|\mu_k(s_t), \Sigma_k(s_t))$.

Models that achieve superior performance on more validation periods demonstrate better capacity to maintain profitability in different market conditions, and have higher weights in the mixture of equal weighted individual distributions. While for models that are overfitted, their individual effects can be reduced by the ensemble policy. Given a specific environment observation, actions that are favored by more models, including the ones overfitted to their corresponding validation periods, represent an approximate consensus among all policies. The ensemble policy can be viewed as a continuous counterpart to the majority vote in discrete cases. With little additional computational overhead, we efficiently leverage the various market conditions in the history and the models along the training trajectory to enhance the generalization ability of DRL trading strategies.

C. Evaluation on Granular Test Periods

We break down the out-of-sample data into granular periods and evaluate the model performance on each test period to gain a distributional understanding of the generalization performance in evolving market conditions. We retrain the model periodically using a rolling window to adapt the model to recent market conditions. By comparing the quantiles of the test performance with those of the market performance, we assess the robustness of DRL strategies behind an overshadowing return on the entire test data. Extreme values in the distribution can indicate either severe failures in certain conditions or rare successes that obscure overall poor performance. The identification of the extreme cases also helps mitigate the false positive reporting of trading strategies that are highly overfitted to specific market conditions despite an overall high return, which is a crucial step before the deployment of trading strategies in live trading.

V. Computational Experiments

A. Data Processing

We use historical hourly OHLCV data from the online cryptocurrency exchange Kraken for the period from 01/01/2018 to 06/30/2022. We consider a portfolio constituted by USD and 5 largest cryptocurrencies by market capitalization in the month of December 2017, namely Bitcoin (XBT), Ethereum (ETH), Bitcoin Cash (BCH), Ripple (XRP) and Litecoin (LTC). For each cryptocurrency, we detrend the time series by taking the percentage change of the raw OHLCV data, to avoid serious failures in generalization when the absolute price range enters a different regime in out-of-sample data. The price trend and momentum information is extracted and preserved by the 7 technical indicators, including SMA_{30} and SMA_{60} of close price for 30 and 60 periods, RSI, CCI, $MACD$, ADX and ATR.

For each time step t, the price and technical features of the 5 cryptocurrencies form a vector $f_t \in \mathbb{R}^{60}$. The features are concatenated with the holdings of the assets and the balance in USD to form a vector $\tilde{s}_t \in \mathbb{R}^{66}$. Since the financial time series is generally considered non-Markovian, we define the states $s_t \in \mathbb{R}^{12\times 66}$ as a sequence $[\tilde{s}_t, \tilde{s}_{t-1}, ..., \tilde{s}_{t-11}]$ and assume that the Markovian property is satisfied for the constructed states s_t.

We apply feature-wise normalization to the states s_t. We standardize the percentage changes of OHLCV, SMA_{30}, SMA_{60}, CCI, $MACD$ and ATR. RSI and $ADX \in [0, 100]$ are linearly mapped to the interval of unit length centered at 0. For holdings and balance, we apply min-max normalization with a predefined maximum value for all t based on the initial portfolio value and initial prices.

The code has been open sourced.[1]

B. Model Architecture

The inputs s_t are of dimension 12×66. We use a fully connected layer followed by a 2-layer LSTM with hidden dimension of 64 for each layer to learn the representation of the sequential data. The sequential feature extraction module is shared by the policy and the value networks. Th policy network uses fully connected layers to output vectors μ, σ as parameters of the stochastic policy, and the value network uses a fully connected layer to output the value of the state. The network architecture is detailed in Table I and the weights are initialized by the Xavier initialization method.

C. Training

We split the data from 01/01/2018 to 06/30/2022 into 48 rolling windows, where each rolling window consists of 6 months of training data and 1 month of test data. For each rolling window, we roll out 400 episodes on the training data, where each episode starts with 1 million USD on the account and zero holdings of cryptocurrencies. The episodes vary only by stochasticity of the policy. We sample 9 weekly periods within the training period as validation sets and evaluate on each validation set every 2 epochs of training. The reward is calculated by the change of the portfolio value after each action and is normalized by the running maximum of past absolute values of rewards. The size of the trading actions is specified by $hmax$ for BTC and scaled for other cryptocurrencies. Based on the immediate rewards and the value function outputs, we apply generalized advantage estimation [36] to calculate the estimated advantages $\hat{A}_t$. We use a discount factor of $\gamma = 0.99$ for the rewards and an exponential weight discount factor $\lambda = 0.95$ for the extended advantage estimators. The objective function for the PPO algorithm consists of the clipped surrogate policy objective, the mean squared error of the value function loss and an entropy term to control exploration and exploitation, weighted by hyperparameters

[1] https://anonymous.4open.science/r/ensemble_DRL_trading-841B

TABLE I
NEURAL NETWORK ARCHITECTURE

Layer	Description	Output shape	Activation
Fully Connected	16 hidden units	(12, 16)	sigmoid
LSTM	2 layers of 64 hidden units	64	tanh
Bacth Normalization	-	-	-
Fully Connected	policy network: mean	5	tanh
Fully Connected	policy network: standard deviation	5	sigmoid
Fully Connected	value network	1	linear

c_1, c_2 and c_3. The loss is calculated in batches and minimized by the Adam [37] optimizer with a linear decaying learning rate schedule. The set of hyperparameters used is summarized in Table II.

D. Metrics

For each rolling window, we backtest the DRL trading strategies on weekly periods on the 1-month out-of-sample data. In total, we backtest on 208 weekly periods of the 4-year test data. For each test period, we calculate the annualized and cumulative returns to directly measure the profitability of the strategy. We also assess the risk-adjusted returns of the trading strategy by the Sortino and Sharpe ratios, where the Sortino ratio measures the return adjusted by downside risks while the Sharpe ratio factors in both upside and downside risks. We use maximum drawdown as an indicator of the downside risk and the standard deviation of hourly returns as a proxy of volatility. The annualized return serves as the primary evaluation metric for the trading performance over a given period, while the Sortino ratio is utilized as an important supplementary metric where the undesired downside risks are factored in.

E. Benchmarks

FinRL-Meta: The framework allows for continuous action spaces and intraday trading of a cryptocurrency portfolio. We include the same technical indicators in addition to the hourly OHLCV data as the input price features of the five cryptocurrencies. The FinRL model is trained by the PPO algorithm for 400 episodes using learning rate of $5 \cdot 10^{-6}$ and scale of the action of $hmax_d = 70$ for all d. Similarly, we use 48 rolling windows to retrain the FinRL model periodically and backtest on 208 weeks of the 4-year test period. These hyperparameters used have been tuned.

Buy and hold: We use the performance of the buy and hold strategy to benchmark our DRL trading strategy with the market performance. For each of the 208 test weeks, the initial fund of 1 million USD is allocated equally among the five cryptocurrencies at the beginning of the period, and the positions are held until the end of the period. The strategy can be regarded as a passive investment strategy that rebalances on a weekly basis to maintain equal allocation of funds among portfolio constituents. Its performance on the 4-year test data closely tracks the market performance of the five cryptocurrencies.

VI. RESULTS

A. Backtesting Results

We present the backtesting results on the 4-year test period from 07/01/2018 to 06/30/2022 in Table III for the ensemble method and the benchmarks. Our ensemble method shows superior profitability in both returns and risk-adjusted returns. The ensemble policy method outperforms the FinRL model in the annualized return by 47% and the buy and hold strategy by 23%. FinRL achieves an overall low risk level with the lowest maximum drawdown and volatility, but our ensemble method still outperforms FinRL in the Sortino ratio by 18% and the buy and hold strategy by 29%.

The cumulative returns for the 4-year test period are presented in Fig. 1. We observe that the ensemble method follows the market trend in the early stage before 2020. The strategy takes a notable advantage of the bullish trend during the cryptocurrency market rally from 2020 to 2021, which results in cumulative returns of up to 25 times until the market experienced a downturn at the end of 2021. The performance is affected by the market downturn reflected in the 69.53% maximum drawdown, but it is not as severe as the 74.30% drawdown of the buy and hold strategy. The ensemble method excels at exploiting the upward trend in the market, and shows limited protection during market downturns.

B. A Distributional View of Returns

In addition to a qualitative description of a strategy's performance from the cumulative returns plots, we take a distributional view of the out-of-sample performance on granular test periods. We present the quantiles of the 208 weekly returns in Table IV. The 100% quantile is lower than the annualized return in Table III since the raw weekly returns without annualization are used to calculate these quantiles. By examining the extreme values, we observe that returns of our proposed method have less extreme returns on both positive and negative sides comparing with the buy and hold returns. The ensemble method shows robust performance relative to the market performance. Returns of our proposed strategy also dominate at most quantile levels at or above 50% except being slightly behind the buy and hold strategy at 70%, showing a high concentration on the large positive returns comparing with the benchmarks. While the ensemble method underperforms the FinRL model in preventing losses, it still outperforms the buy and hold strategy at all quantile levels at

TABLE II
HYPERPARAMETERS

input sequence length	episodes	learning rate	batch size	hmax	γ	λ	c_1	c_2	c_3
12	400	$5 \cdot 10^{-6}$	6,000	70	0.99	0.95	100	-1	-1

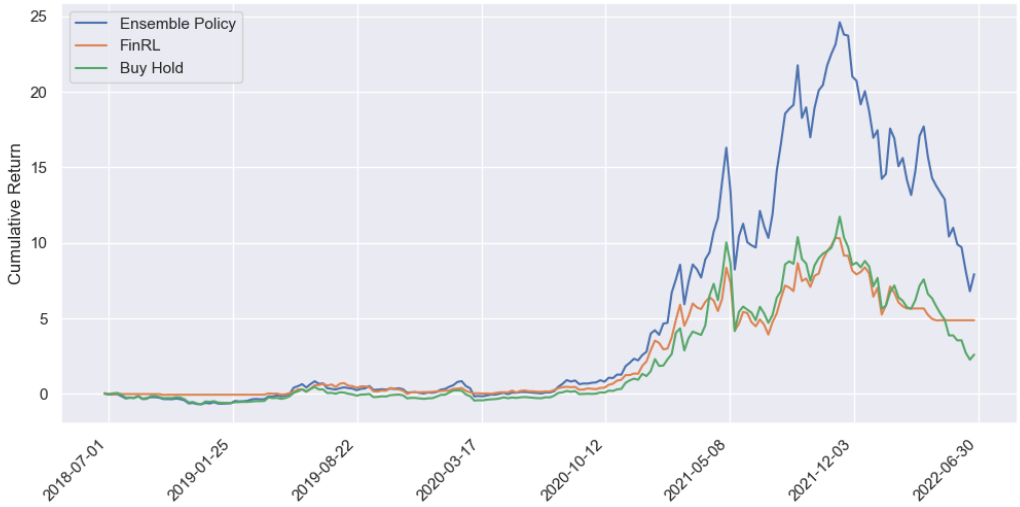

Fig. 1. Cumulative returns on 4-year out-of-sample period

TABLE III
BACKTESTING SUMMARY

	Ensemble policy	FinRL	Buy-hold
Annualized return	**0.9319**	0.6320	0.7571
Cumulative return	**7.9361**	4.8851	2.6088
Sortino	**1.6218**	1.3762	1.2560
Sharpe	**1.0724**	1.0401	0.8119
Max drawdown	0.6953	**0.4818**	0.7430
Volatility	0.8690	**0.6076**	0.9325

TABLE IV
QUANTILES OF WEEKLY RETURNS DISTRIBUTION

Quantile	Ensemble Policy	FinRL	Buy-hold
100%	0.4254	0.2825	**0.5630**
90%	**0.1480**	0.1243	0.1470
80%	**0.1085**	0.0693	0.0958
70%	0.0644	0.0240	**0.0646**
60%	**0.0411**	0.0061	0.0313
50%	**0.0210**	0.0000	0.0144
40%	-0.0112	**-0.0000**	-0.0118
30%	-0.0307	**-0.0066**	-0.0368
20%	-0.0647	**-0.0284**	-0.0705
10%	-0.1218	**-0.0835**	-0.1274
0%	-0.3882	**-0.3774**	-0.4633

or below 60%. Our ensemble method also has the highest median return among the three strategies. The distributional view shows that the returns of the ensemble method concentrate at higher values compared with both the FinRL and the buy and hold strategy, and demonstrates some extent of downside risk control compared to more cases of large losses incurred by the passive investment strategy. The results corroborate the finding that the ensemble method achieves much higher positive returns in a bullish market from the cumulative returns plot.

We can also take a distributional view of the performance of the 48 models trained on the rolling window periods. Since the model is retrained every month, we aggregate the weekly returns obtained by the same model trained on the preceding 6-month rolling window to calculate 48 monthly returns. The distribution of the monthly returns reflect the robustness of the performance of the trained model to different training processes. We present the empirical cumulative density function (eCDF) plots for our proposed method and the two benchmarks in Fig. 2. The monthly returns of our proposed models dominate the buy and hold returns at all levels for negative returns, and dominate the FinRL model at all levels for positive returns. As we can see from the histograms of the monthly returns in Fig. 3, the FinRL model lacks the capacity to produce high returns and the returns are highly concentrated on the insignificant values around zero, while the the buy and hold returns take more extreme values on both positive and negative sides. The ensemble method strikes a balance that avoids severe negative returns but also generates more significant positive returns.

Fig. 2. Empirical CDF of monthly returns

Fig. 3. Histograms of monthly returns

C. *Effectiveness of the Ensemble*

We present the evidence of the effectiveness of the ensemble policy by comparing with the model at the last epoch of training and with the average performance of individual models. The model at the final epoch of training is susceptible to overfitting the training data, and there is little guarantee that the market will exhibit similar trends in the subsequent month. Therefore, in the event of a regime shift, such models may cause serious failures on out-of-sample data. Our ensemble policy accounts for the consensus among different models saved during training to overcome the overfitting issue of the last epoch model. As shown in Table V, the ensemble model outperforms the final epoch model in all metrics considered, including an improvement of 15% in annualized return, 17% in Sortino ratio, and 13% less in maximum drawdown. The results provide evidence that the ensemble method mitigates the overfitting issue at the end of training.

To show that our proposed method effectively ensembles the individual models, we verify that the ensemble policy produces better results than the expectation of the performance

TABLE V
PERFORMANCE OF MODEL AT LAST EPOCH

	Final epoch	Ensemble policy
Annualized return	0.8092	**0.9319**
Cumulative return	3.7503	**7.9361**
Sortino	1.3804	**1.6218**
Sharpe	0.8865	**1.0724**
Max drawdown	0.8005	**0.6953**
Volatility	0.9128	**0.8690**

of individual constituent models. We conclude from the aggregated results of the individual models in Table VI that the ensemble method achieves better performance in all metrics considered comparing with the mean and median values of the performance of individual models. The results support the effectiveness of the ensemble policy compared to the average of individual models selected based on a single validation period.

TABLE VI
PERFORMANCE OF INDIVIDUAL MODELS

	Average	Median	Standard deviation
Annualized return	0.8651	0.8698	0.0802
Cumulative Return	5.8103	5.9666	1.7859
Sortino	1.5485	1.5864	0.1534
Sharpe	0.9713	0.9877	0.0714
Max Drawdown	0.7662	0.7756	0.0257
Volatility	0.8895	0.8938	0.0224

VII. CONCLUSION AND FUTURE WORK

In this work, we focus on improving the out-of-sample performance of DRL trading strategies using an ensemble method. We propose a novel mixture distribution policy that effectively ensembles multiple models selected by the smoothed validation performance on different periods during training. Our method efficiently leverages the historical market conditions and the models along the optimization trajectory to mitigate the overfitting issue for the model at the last epoch of training. Compared with the DRL strategy benchmark FinRL-Meta and the buy and hold strategy, our proposed method demonstrates a significant improvement in out-of-sample returns. We evaluate the model performance on granular test periods and present a distributional view of the returns. The distribution of the returns shows robustness of our ensemble strategy compared with market performance. Our returns show a higher concentration on the large returns compared with the FinRL returns, and also exhibit less extreme positive and negative returns compared with the buy and hold strategy.

The results open up a promising direction to design robust DRL trading strategies using a novel ensemble policy. Future work can incorporate the covariance matrix to the $\tanh$ transformed Gaussian policy to capture the relationships among the portfolio constituents. We conduct an experiment where the policy network also learns a lower triangular matrix L to form a positive semi-definite matrix $Cov = LL^\top$ as the covariance

matrix. We train the model with the modified policy network and backtest on the 208 weekly periods. The annualized return is 0.6832 and the Sortino ratio is 1.2620. The modified model underperforms our proposed model, but outperforms the buy and hold strategy in the Sortino ratio by a small margin. Future work can improve the learning of the covariance matrix using the covariance of historical asset returns as a warm start.

Future work can also study the effect of transaction costs and modify the objective to be maximizing of the expected return while minimizing the transaction costs. We can also relax the assumption of nonnegative holdings to allow both long and short positions and explore hedging strategies by DRL algorithms.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[3] L. Chen and Q. Gao, "Application of deep reinforcement learning on automated stock trading," in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2019, pp. 29–33.

[4] Z. Jiang and J. Liang, "Cryptocurrency portfolio management with deep reinforcement learning," in *2017 Intelligent Systems Conference (IntelliSys)*. IEEE, 2017, pp. 905–913.

[5] C. Y. Huang, "Financial trading as a game: A deep reinforcement learning approach," *arXiv preprint arXiv:1807.02787*, 2018.

[6] X.-Y. Liu, H. Yang, J. Gao, and C. D. Wang, "Finrl: Deep reinforcement learning framework to automate trading in quantitative finance," in *Proceedings of the Second ACM International Conference on AI in Finance*, 2021, pp. 1–9.

[7] T. Théate and D. Ernst, "An application of deep reinforcement learning to algorithmic trading," *Expert Systems with Applications*, vol. 173, p. 114632, 2021.

[8] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, and H. Fujita, "Adaptive stock trading strategies with deep reinforcement learning methods," *Information Sciences*, vol. 538, pp. 142–158, 2020.

[9] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, vol. 7, pp. 108 014–108 022, 2019.

[10] K. S. Zarkias, N. Passalis, A. Tsantekidis, and A. Tefas, "Deep reinforcement learning for financial trading using price trailing," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3067–3071.

[11] H. Yang, X.-Y. Liu, S. Zhong, and A. Walid, "Deep reinforcement learning for automated stock trading: An ensemble strategy," in *Proceedings of the First ACM International Conference on AI in Finance*, 2020, pp. 1–8.

[12] S. Carta, A. Corriga, A. Ferreira, A. S. Podda, and D. R. Recupero, "A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning," *Applied Intelligence*, vol. 51, pp. 889–905, 2021.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] X.-Y. Liu, Z. Xia, J. Rui, J. Gao, H. Yang, M. Zhu, C. Wang, Z. Wang, and J. Guo, "Finrl-meta: Market environments and benchmarks for data-driven financial reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1835–1849, 2022.

[15] A. Nan, A. Perumal, and O. R. Zaiane, "Sentiment and knowledge based algorithmic trading with deep reinforcement learning," in *International Conference on Database and Expert Systems Applications*. Springer, 2022, pp. 167–180.

[16] M. Taghian, A. Asadi, and R. Safabakhsh, "Learning financial asset-specific trading rules via deep reinforcement learning," *Expert Systems with Applications*, vol. 195, p. 116523, 2022.

[17] T. Kabbani and E. Duman, "Deep reinforcement learning approach for trading automation in the stock market," *IEEE Access*, vol. 10, pp. 93 564–93 574, 2022.

[18] B. Hirchoua, B. Ouhbi, and B. Frikh, "Deep reinforcement learning based trading agents: Risk curiosity driven learning for financial rules-based policy," *Expert Systems with Applications*, vol. 170, p. 114553, 2021.

[19] Y. Li, P. Ni, and V. Chang, "Application of deep reinforcement learning in stock trading strategies and stock forecasting," *Computing*, vol. 102, no. 6, pp. 1305–1322, 2020.

[20] X.-Y. Liu, Z. Xiong, S. Zhong, H. Yang, and A. Walid, "Practical deep reinforcement learning approach for stock trading," *arXiv preprint arXiv:1811.07522*, 2018.

[21] Z. Zhang, S. Zohren, and R. Stephen, "Deep reinforcement learning for trading," *The Journal of Financial Data Science*, 2020.

[22] M. Guan and X.-Y. Liu, "Explainable deep reinforcement learning for portfolio management: an empirical approach," in *Proceedings of the Second ACM International Conference on AI in Finance*, 2021, pp. 1–9.

[23] A. Riva, L. Bisi, P. Liotet, L. Sabbioni, E. Vittori, M. Pinciroli, M. Trapletti, and M. Restelli, "Addressing non-stationarity in fx trading with online model selection of offline rl experts," in *Proceedings of the Third ACM International Conference on AI in Finance*, 2022, pp. 394–402.

[24] C.-H. Kuo, C.-T. Chen, S.-J. Lin, and S.-H. Huang, "Improving generalization in reinforcement learning–based trading by using a generative adversarial market model," *IEEE Access*, vol. 9, pp. 50 738–50 754, 2021.

[25] V. Kochliaridis, E. Kouloumpris, and I. Vlahavas, "Combining deep reinforcement learning with technical analysis and trend monitoring on cryptocurrency markets," *Neural Computing and Applications*, pp. 1–18, 2023.

[26] B. J. D. Gort, X.-Y. Liu, X. Sun, J. Gao, S. Chen, and C. D. Wang, "Deep reinforcement learning for cryptocurrency trading: Practical approach to address backtest overfitting," *arXiv preprint arXiv:2209.05559*, 2022.

[27] O. Sattarov, A. Muminov, C. W. Lee, H. K. Kang, R. Oh, J. Ahn, H. J. Oh, and H. S. Jeon, "Recommending cryptocurrency trading points with deep reinforcement learning approach," *Applied Sciences*, vol. 10, no. 4, p. 1506, 2020.

[28] G. Lucarelli and M. Borrotti, "A deep reinforcement learning approach for automated cryptocurrency trading," in *Artificial Intelligence Applications and Innovations: 15th IFIP WG 12.5 International Conference, AIAI 2019, Hersonissos, Crete, Greece, May 24–26, 2019, Proceedings 15*. Springer, 2019, pp. 247–258.

[29] F. Cornalba, C. Disselkamp, D. Scassola, and C. Helf, "Multi-objective reward generalization: Improving performance of deep reinforcement learning for applications in single-asset trading," *arXiv preprint arXiv:2203.04579*, 2022.

[30] C. Betancourt and W.-H. Chen, "Reinforcement learning with self-attention networks for cryptocurrency trading," *Applied Sciences*, vol. 11, no. 16, p. 7377, 2021.

[31] J. Sadighian, "Deep reinforcement learning in cryptocurrency market making," *arXiv preprint arXiv:1911.08647*, 2019.

[32] Y. Ye, H. Pei, B. Wang, P.-Y. Chen, Y. Zhu, J. Xiao, and B. Li, "Reinforcement-learning based portfolio management with augmented asset movement prediction states," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1112–1119.

[33] P. Koratamaddi, K. Wadhwani, M. Gupta, and S. G. Sanjeevi, "Market sentiment-aware deep reinforcement learning approach for stock portfolio allocation," *Engineering Science and Technology, an International Journal*, vol. 24, no. 4, pp. 848–859, 2021.

[34] L. K. Felizardo, F. C. L. Paiva, C. de Vita Graves, E. Y. Matsumoto, A. H. R. Costa, E. Del-Moral-Hernandez, and P. Brandimarte, "Outperforming algorithmic trading reinforcement learning systems: A supervised approach to the cryptocurrency market," *Expert Systems with Applications*, vol. 202, p. 117259, 2022.

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[36] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.