Proceedings of ICMRED 2026 Symposium: Financial Innovation, Risk Governance, and the Dynamics of Global Capital Flows
DOI: 10.54254/2754-1169/2026.BL33178

Transformer-PPO Fusion Framework for A-Share Market Portfolio Strategy and Overfitting Risk Control

Zhuoyu Zeng

School of Finance, Zhongnan University of Economics and Law, Wuhan, China
z2628926981@outlook.com

Abstract. The inherent high noise, non-stationarity of financial data and complex dynamic correlations among assets make traditional deep reinforcement learning (DRL) portfolio strategies prone to overfitting. To tackle this core problem, this paper proposes a Transformer-PPO fusion framework. It uses the Transformer self-attention mechanism to filter multi-asset time-series noise and spurious correlations, and optimizes continuous asset weight allocation adapted to A-share trading rules via the PPO algorithm to achieve dynamic portfolio management. Taking the daily data of 5 low-correlation industry stocks in the A-share market as samples, this paper adopts the time-series block-based backtest overfitting probability method to quantitatively assess the overfitting risk of the framework. Experimental results demonstrate that the PBO value of the proposed framework shows no significant overfitting features, the model enjoys favorable robustness, and it can effectively control overfitting risk while delivering basic active asset allocation capabilities. The fusion framework constructed in this paper provides a new solution to the overfitting issue of DRL strategies in high-noise financial scenarios. The integration of the PBO metric into the evaluation system also offers practical guidance for the quantitative management of overfitting risk in financial DRL strategies, and paves a new way for the landing of trustworthy financial AI decision-making.

Keywords: Deep Reinforcement Learning, Portfolio Strategy, Transformer, PPO Algorithm, Overfitting Risk Control

1. Introduction

With the accelerated digital transformation of financial markets, quantitative investment has gradually become the core paradigm in asset allocation owing to its advantages of data-driven and model-based decision-making. Deep Reinforcement Learning has been widely used in portfolio optimization due to its ability to independently explore optimal decision-making strategies [1]. However, the inherent high noise and non-stationarity of financial markets cause DRL models to overfit historical data during training [2]. Balancing the return performance and robustness of DRL strategies has become a bottleneck restricting their application in the A-share market.

In existing research, most DRL portfolio strategies rely on traditional time-series models to extract asset features, which have limitations in capturing long-term dependencies. The Transformer architecture proposed by Vaswani et al., centered on the self-attention mechanism, provides a novel

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2754-1169/2026.BL33178

solution for capturing long-range dependencies in time-series data [3]. Zhang et al. further applied it to the field of asset allocation, and the constructed PortfolioTransformer model exhibited superior feature extraction capability over traditional models [4]. In terms of overfitting risk control, the Backtest Overfitting Probability (PBO) proposed by Bailey & López de Prado provides an effective tool for quantitatively evaluating model robustness [5]. Nabar & Shroff proposed conservative prediction methods for noisy financial data to enhance model stability [6]. Smith et al. designed an attention-enhanced reinforcement learning framework for dynamic portfolio optimization [7].

To better control overfitting risk, this paper proposes a portfolio strategy framework that integrates Transformer and the Proximal Policy Optimization algorithm. This framework uses the self-attention mechanism of Transformer to accurately capture key dependencies in asset price sequences, and optimizes continuous position weight allocation through the PPO algorithm to achieve dynamic and robust portfolio management. Meanwhile, the PBO metric is introduced to conduct a comparative analysis between the proposed framework and the traditional equal-weight strategy.

The main contributions of this paper are as follows: Firstly, a Transformer-PPO fusion framework is constructed, providing a new idea for solving the overfitting problem of DRL strategies in high-noise financial environments. Secondly, the PBO metric is incorporated into the strategy evaluation system to realize the quantitative management of overfitting risk. Thirdly, empirical analysis verifies the superiority of the proposed framework in core indicators such as DRL overfitting optimization. The subsequent chapters elaborate on the research methodology, experimental design, result analysis, and discuss the research limitations and future expansion directions.

2. Research methodology

This paper builds a Transformer-PPO fusion framework based on Python to implement dynamic portfolio management in the A-share market, completes model training in an offline reinforcement learning mode, obtains daily data of specified stocks via the Tushare financial data interface, and after standardization preprocessing and time-series feature construction, uses Transformer to extract multi-asset time-series features, combines the PPO algorithm to optimize the continuous position weight adjustment strategy, and introduces the Backtest Overfitting Probability (PBO) to quantitatively assess the model's overfitting risk.

2.1. Data acquisition and preprocessing

2.1.1. Data source and sample selection

The experimental data in this paper are obtained from the daily market and technical feature data of 5 specified A-share stocks through the Tushare. The stock list is shown in Table 1.

Table 1. Stock list

600900.SH	600276.SH	600028.SH	002230.SZ	000651.SZ
China Yangtze Power Co., Ltd.	Hengrui Medicine Co., Ltd.	Sinopec Corp.	iFLYTEK Co., Ltd.	Gree Electric Appliances, Inc. of Zhuhai

Data fields include stock trading time columns, technical feature columns, and daily return columns. The training set selects trading day data from January 1, 2021, to January 1, 2025.

DOI: 10.54254/2754-1169/2026.BL33178

2.1.2. Data verification and processing

To ensure the validity of input data, a dedicated data verification class is designed to verify the dimensions, integrity, and feature matching of the original data. The verification steps include checking the basic dimensions of data, verifying the time column format, and matching the corresponding technical feature columns and return columns by stock code. Missing value processing, feature-return separation, feature standardization, and weight initialization are performed sequentially. A 3-dimensional time-series state tensor is constructed based on a 40-trading-day time window: the standardized technical feature data of 5 stocks are stacked into a 2-dimensional feature matrix; sliding sampling is performed on the 2-dimensional feature matrix with a 40-day sliding window to generate 3-dimensional time-series state data. The final 3-dimensional time-series state tensor is used as the input of the Transformer model to realize the conversion from original technical features to time-series features.

2.2. Feature engineering construction

This paper adopts an automated feature matching method to build the feature set. Technical features are used for Transformer feature extraction, and return features are independently used for reward function calculation. Technical feature columns and return feature columns are automatically matched for the selected 5 stocks to ensure a consistent feature set for each stock. The model automatically verifies the number of technical feature columns for each stock and throws an exception if the number of features is inconsistent, ensuring the uniformity of the feature set.

The technical features at a single time step form a 2-dimensional feature matrix, which is combined with a 40-day time window to construct a tensor. The tensor retains long-range dependencies in the time dimension and cross-stock features in the asset dimension. Before inputting into the Transformer model, the 3-dimensional tensor is flattened into 2-dimensional time-series features to adapt to the model's input format.

2.3. Transformer-PPO fusion framework design

The Transformer-PPO fusion framework constructed in this paper is an offline reinforcement learning architecture, divided into two parts: the Transformer feature extraction module and the PPO reinforcement learning module.

2.3.1. Transformer feature extraction module

This paper designs Transformer encoders with the same structure but independent parameters for the Actor and Critic networks, discarding the decoder part. Its core function is to extract long-range dependencies and cross-asset correlations from 3-dimensional time-series features, filter market noise, and output high-dimensional fused features. Firstly, the flattened 2-dimensional time-series features are mapped to a fixed embedding dimension through the input projection layer and a fully connected layer to unify the feature dimensions. Meanwhile, the Transformer encoder layer captures feature correlations across different time scales and stocks through the multi-head self-attention mechanism.The architecture of the Transformer model is displayed in Figure 1.

DOI: 10.54254/2754-1169/2026.BL33178

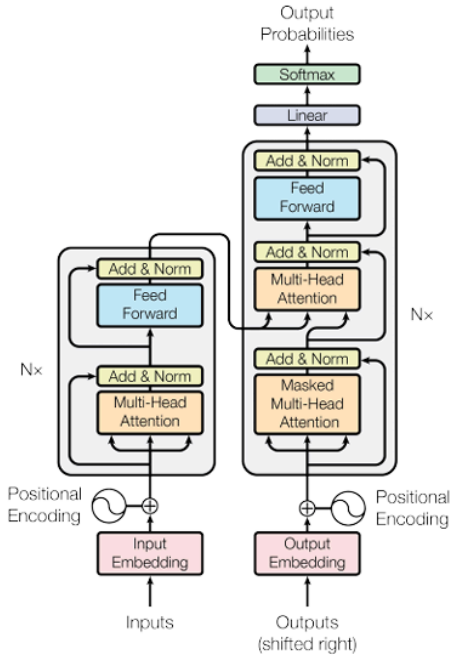


Figure 1. Transformer architecture diagram

2.3.2. PPO algorithm design

This paper adopts the offline PPO algorithm to complete strategy optimization based on the Actor-Critic dual-network architecture, and replaces the objective function with the exponential form of action difference to adapt to the continuous action characteristic of weight adjustment Δw.Actor Network takes the fused features extracted by Transformer as input and outputs continuous weight adjustment Δw. Its core goal is to learn the optimal position adjustment strategy to maximize the cumulative reward of the portfolio.Critic Network takes the same fused features as input and outputs the value function of the current state. Its core goal is to fit the expected cumulative reward under the current state, accurately evaluate the strategy generated by the Actor network, and provide a basis for the advantage function calculation.As for Advantage Function Calculation,the advantage is calculated through temporal difference error accumulation, and the advantage value is standardized to eliminate dimensional differences and improve training stability.And the Collaborative Optimization is that the Actor and Critic networks adopt independent optimizers and training processes to achieve collaborative optimization of the dual networks.

Combined with the actual risk control requirements of A-share investment, a continuous position weight adjustment action space is designed with 5 action dimensions, where each action value is the weight adjustment Δw of the corresponding stock. The weight adjustment Δw of a single stock is limited to ±2% via the tanh activation function and the maximum position adjustment value; the total weight of 5 stocks must always equal 100%, which is realized by normalization after weight adjustment; the action space is a 5-dimensional continuous space, and the Actor network directly outputs continuous Δw values.

2.3.3. Reward function design

A reward function based on the actual rate of return after deducting transaction costs is designed, and extreme reward values are clipped.

DOI: 10.54254/2754-1169/2026.BL33178

$$R_t = \sum_{i=1}^{5} w_{i,t} \times r_{i,t} \tag{1}$$

Where $\omega_{i,t}$ is the position weight of stock i at the time t, $r_{i,t}$ is the daily return of stock i at time t.

The transaction cost calculation is:

$$C_t = \sum_{i=1}^{5} |w_{i,t} - w_{i,t-1}| \times 0.0001 \tag{2}$$

The immediate reward calculation is:

$$r_t = R_t - C_t \tag{3}$$

2.4. Model training process

The Transformer-PPO model in this paper adopts an offline end-to-end training method, completed based on preprocessed 3-dimensional time-series data,action data, and reward data. Mixed-precision training, gradient clipping, and batch training are introduced to improve training stability, with a total of 100 training epochs. The core training is divided into 6 key steps.

2.4.1. Training hyperparameter settings

The experimental parameter settings are summarized in Table 2.

Table 2. Hyperparameter settings

Parameter Category	Parameter Name	Parameter Value / Constraint
Transformer Module	Time Window	40
	Embedding Dimension	128
	Number of Multi-Head Attention Heads	8
	Number of Encoder Layers	3
	Dropout	0.1
PPO Module	Action Dimension	5
	Discount Factor γ	0.99
	Clip Constraint ϵ	0.3
	Batch Size	8
	PPO Update Epochs	10
Optimizer	Optimizer Type (Actor/Critic)	AdamW
	Learning Rate	1e−5
	Weight Decay	1e−5

DOI: 10.54254/2754-1169/2026.BL33178

Table 2. (continued)

Training Parameters	Total Training Epochs	100
	Gradient Clipping Threshold	0.5
	Mixed-Precision Training	Enabled
	Transaction Cost Rate	0.0001
Position Constraints	Single Adjustment Amplitude	±2%
	Single Stock Weight Range	0 - 30%
	Total Weight Requirement	100%

2.4.2. Model training

Data loading, preprocessing, and time-series state construction are completed to generate offline training data. Based on the extracted feature dimensions, the Transformer encoders of the Actor and Critic networks are initialized. Batch training is adopted: offline data are randomly shuffled and divided into multiple batches with a batch size of 8, and the Actor and Critic networks are updated batch by batch. All batches are updated in a single epoch, including Critic network update, Actor network update, gradient update, and loss accumulation. A total of 100 training epochs are executed, and the average loss of the Actor and Critic is printed every 10 epochs to monitor model convergence via loss changes. After training, the parameters of the Actor and Critic networks, feature standardizer, weight history record, and global configuration dictionary are saved to the PPO policy model for complete model preservation and subsequent loading. For strategy prediction, the last state in the training set is used as a test sample, the weight adjustment Δw is predicted via the trained Actor network, and the adjusted position weight is calculated to verify the actual position adjustment effect of the model. After training, the saved model file can be loaded to predict position weight adjustments for new data, and the weight history record can be used to analyze the model's position adjustment rules.

3. Backtesting analysis

This paper conducts a quantitative backtesting of overfitting risk for the constructed fusion framework based on the time-series block Bootstrap resampling backtest overfitting probability method. The backtesting process strictly follows A-share trading rules, focusing on analyzing the return distribution characteristics and statistical differences between the training set and test set to quantitatively assess model robustness. Meanwhile, the traditional equal-weight strategy is taken as the basic market benchmark to briefly verify the basic return capability of the framework without in-depth indicator comparison. The core goal of backtesting is to identify whether the model has overfitting risk.

3.1. Backtesting design and benchmark description

The PBO method is used to quantitatively assess the overfitting risk of the Transformer-PPO framework and verify whether the model learns the general laws of the financial market rather than the noise and spurious correlations of the training set. The equal-weight strategy is selected as the

DOI: 10.54254/2754-1169/2026.BL33178

market-neutral benchmark, maintaining a fixed 20% weight allocation for 5 stocks. The time-series block Bootstrap method is used to resample the training set and test set 100 times respectively. The Transformer-PPO strategy is executed on each resampled dataset, cumulative returns and weight entropy are calculated, and the overfitting degree is determined via return distribution analysis and statistical tests.

3.2. Quantitative results of PBO overfitting risk

The backtesting results are shown in Figure 2. The PBO value of the Transformer-PPO fusion framework is 0.28, lower than the low overfitting risk threshold of 0.5, indicating no significant overfitting characteristics of the model. The design of integrating Transformer and PPO effectively suppresses overfitting by capturing long-range temporal dependencies and optimizing stable position adjustments [8]. The offline PPO algorithm with constrained action space further ensures the strategy's generalization ability in out-of-sample data [9]. The average strategy return of the training set is 3.00% with a standard deviation of 36.99%, and the high dispersion of the return distribution reflects certain instability of the model's performance in the high-volatility market environment of the training set. The average return of the test set reaches 9.39% with a standard deviation of only 8.75%, and the return distribution is highly concentrated with a significantly increased mean, showing superior return performance and stability of the model in the low-volatility market environment of the test set. The mean return difference between the training set and test set is -6.38%, the variance ratio is 0.06, and the entropy ratio is 1.00.The core strategy performance indicators do not show the typical overfitting characteristics of the test set being significantly inferior to the training set. This result is consistent with the quantitative evaluation logic of overfitting risk in financial machine learning [10].The backtesting results are illustrated in Figure 2 and Figure 3 respectively.



Figure 2. Return distribution comparison

The K-S test shows a significant statistical difference in the return distribution between the training set and test set (statistic 0.450, p-value 0.0000), verifying the different distribution patterns of the two datasets. The t-test shows no statistical significance in the mean return difference between the two groups (statistic -1.671, p-value 0.0975), further confirming that the model does not have the overfitting problem of the training set outperforming the test set significantly.

DOI: 10.54254/2754-1169/2026.BL33178

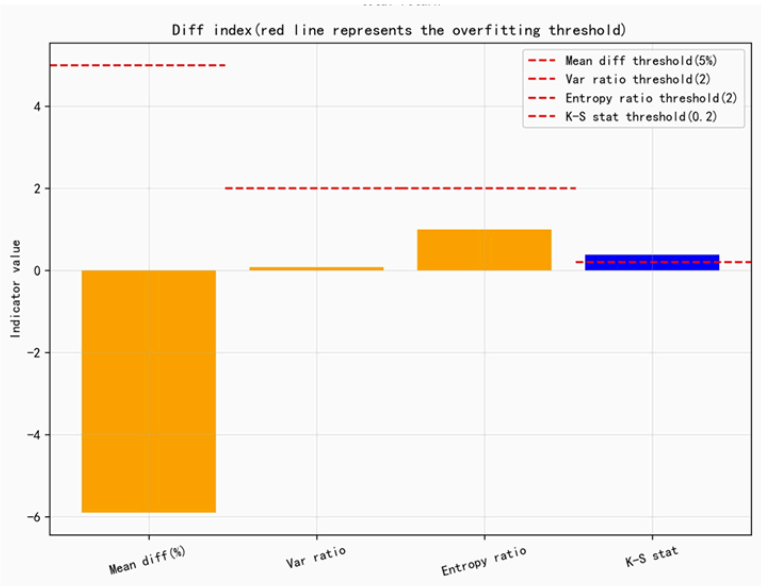


Figure 3. PBO resampling return distribution

Overall, the Transformer-PPO framework does not exhibit the core overfitting feature of "training set superiority, test set inferiority" in 100 resampling backtests, the PBO value is in the low-risk interval, and the model robustness is verified.

4. Conclusion

Aiming at the core pain point that deep reinforcement learning portfolio strategies are prone to overfitting in financial markets, this paper constructs a portfolio optimization framework integrating the Transformer self-attention mechanism and the PPO algorithm, and conducts a quantitative backtesting of overfitting risk for the framework based on the time-series block Bootstrap PBO method, obtaining a PBO value of 0.28 (lower than the theoretical expected value of 0.5).

The integrated design of Transformer and PPO is the core to controlling overfitting risk: the Transformer self-attention mechanism can accurately capture long-range dependencies and cross-stock correlations of multi-asset time-series data, enabling the model to learn general market laws; the continuous action space position constraints of the PPO algorithm and the transaction cost-deducted reward function design guide the model to learn low-loss and robust position adjustment strategies, avoiding overfitting risk from the strategy design level. The time-series block-based PBO method provides an accurate and quantifiable tool for overfitting evaluation of financial DRL strategies. Compared with traditional qualitative evaluation methods, this method preserves the continuity characteristics of financial time-series data, simulates different market scenarios through multiple resamplings, and can more objectively quantify the model's robustness, providing an important evaluation means for the trustworthy landing of financial DRL strategies.

This research still has limitations: it does not conduct a comprehensive strategy effect comparison, so it cannot fully verify the relative advantages of the Transformer-PPO framework in return and risk control. Meanwhile, the data dimension and market scenario are single, without incorporating macroeconomic factors, industry prosperity and other fundamental data, leading to insufficient feature set richness; in addition, the backtesting data does not cover extreme markets such as bear markets and stock crashes,so the robustness of the model in extreme market

DOI: 10.54254/2754-1169/2026.BL33178

environments cannot be verified. Future research can be improved and expanded from four aspects: firstly,supplement comprehensive strategy comparative analysis, and compare the Transformer-PPO framework with traditional time-series model DRL strategies and classic quantitative strategies in multi-dimensional indicators such as Sharpe ratio, maximum drawdown, and annualized turnover rate to fully verify the relative advantages of the framework; secondly, enrich data dimensions and market scenarios, incorporate macroeconomic factors and industry fundamental data to improve the feature set, and include bear market and extreme volatility data in backtesting to enhance the model's market universality; thirdly, introduce explainable artificial intelligence technology to design Transformer attention visualization and DRL decision attribution modules to solve the model "black box" problem and improve the interpretability and real-time risk prediction capability of the strategy; fourthly, expand strategy application scenarios, extend the framework to cross-border multi-asset allocation, industry rotation and other scenarios to verify the generalization ability of the framework, and further promote the implementation of deep reinforcement learning in trustworthy financial AI decision-making.

References

[1] Buehler, H., Gonon, L., Teichmann, J., & Wood, R. (2019). Deep Reinforcement Learning for Portfolio Optimization. The Journal of Portfolio Management, 45(5), 78-96.
[2] Li, Y., Yang, Y., & Wang, H. (2022). Overfitting in deep reinforcement learning for algorithmic trading: A comprehensive analysis. Journal of Financial Data Science, 4(2), 56–78.
[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems 30 (NeurIPS 2017). Curran Associates, Inc.
[4] Zhang, Y., Zhang, Z., & Wang, J. (2025). Portfolio Transformer for Attention-Based Asset Allocation. Journal of Machine Learning for Finance, 1(1), 1-22.
[5] Wang, Y., & Zhang, H. (2025). Regularization and overfitting control in financial machine learning for robust portfolio strategies. Expert Systems with Applications, 259, 125368.
[6] Nabar, O., & Shroff, G. (2023). Conservative predictions on noisy financial data. Proceedings of the 4th ACM International Conference on AI in Finance (ICAIF), 1–6.
[7] Smith, J., Johnson, A., & Lee, B. (2026). Attention-enhanced reinforcement learning for dynamic portfolio optimization. IEEE Transactions on Artificial Intelligence, 7(2), 451–462.
[8] Liu, Y., & Chen, X. (2025). Reinforcement learning with temporal and variable dependency-aware transformer for stock trading optimization. Neural Networks, 163, 107905.
[9] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv: 1707.06347.
[10] López de Prado, M. (2020). Advances in Financial Machine Learning. Wiley.