Original Research Article

Fixed Deep Residual Networks with Multi-Objective Proximal Policy Optimization for Adaptive Trading

Xichen Song

University of California, Berkeley, Berkeley, CA 94720, United States

ABSTRACT

Adaptive trading systems must balance return generation with downside control while processing noisy, non stationary market data. This study evaluates a preference conditioned trading framework that combines a randomly initialized, permanently frozen deep residual network (DRN) with a single multi-objective proximal policy optimization (PPO) agent. Daily Bitcoin data from 2021 through 2025 were transformed into nine technical indicators and organized into 14 day windows. The frozen DRN mapped each window to a 32 dimensional representation, which was concatenated with profit and downside preference weights before entering the PPO actor critic policy. Four analyses were performed. A controlled seed 67 encoder ablation compared raw indicators, a fixed linear projection, a fixed non residual convolutional network, the fixed DRN, and a jointly trained DRN. The fixed DRN achieved the highest mean net profit across 20 preference evaluations (26.15%), whereas raw indicators produced the shallowest mean maximum drawdown (-5.15% versus -11.61% for the fixed DRN). Reward diagnostics showed that the directional profit and downside return components were of the same order of magnitude, with mean absolute values of 0.02194 and 0.01097, respectively. An ordered preference analysis showed measurable changes in policy probabilities and action composition, but deterministic actions were piecewise constant and maximum drawdown was unchanged across the seed 67 grid. In a common date seed 67 benchmark, the unified model produced 41.60% net profit, compared with 16.68% for a 20/50 day moving average strategy, 2.80% for the traditional fuzzy ensemble, and -0.06% for Buy-and-Hold. However, seven independently retrained seeds produced substantial uncertainty. At the neutral preference, the paired net profit difference favored the unified model by 7.73 percentage points, but the 95% bootstrap confidence interval included zero (-4.78 to 21.07), and the unified model exhibited deeper drawdowns on average. The results show that a fixed residual random projection can support profitable preference conditioned policies in some Bitcoin backtests, but they do not establish statistically robust superiority, improved maximum drawdown control, or a smooth continuous risk-return frontier.

Keywords: adaptive trading; proximal policy optimization; multi-objective reinforcement learning; deep residual network; random features; cryptocurrency trading; downside return control

INTRODUCTION

Developing a reliable quantitative trading system requires balancing the competing goals of generating returns and limiting capital loss. Higher return targets are often associated with greater volatility, larger drawdowns, and increased sensitivity to changing market regimes. A rigid single-objective strategy may

Corresponding author: Xichen Song, E-mail: xichen_song@berkeley.edu.

Accepted August 4, 2026
https://doi.org/10.70251/HYJR2348.44727739

therefore be unsuitable for investors or institutions whose preferences can change over time.

Financial markets are also non-stationary. The usefulness of a trading rule can vary with volatility, liquidity, participant behavior, and broader economic conditions (1). Reinforcement-learning policies face an especially difficult generalization problem because the policy is trained on one historical distribution but deployed on later market regimes. Proximal policy optimization (PPO) addresses part of the optimization problem by constraining large policy updates through a clipped surrogate objective (2). PPO and related actor-critic methods have been applied to automated trading, including ensemble systems that combine multiple reinforcement-learning algorithms (3). Other financial reinforcement-learning studies incorporate proportional transaction costs directly into portfolio evaluation (4).

The return control problem is naturally multi-objective. Multi-objective reinforcement learning (MORL) distinguishes methods that retain several objective-specific policies from methods that condition one policy on preferences or desired outcomes (5). Nguyen *et al*. developed a scalable framework supporting single-policy and multi-policy strategies (6), while Pareto Conditioned Networks demonstrated that a single network can condition behavior on a desired return vector (7). These studies motivate preference-conditioned policies, but a finite set of preference evaluations should not automatically be described as a Pareto frontier. Formal Pareto claims require explicit nondominance analysis and coverage metrics.

A separate question concerns market representation. Residual convolutional networks have provided strong time series baselines (8). Random feature methods show that a fixed randomized nonlinear map can provide a useful basis for a trained downstream model (9), and ROCKET applies random convolutional kernels to time series representation (10). Technical indicators have also shown out-of-sample forecasting value for daily Bitcoin returns (11). These findings provide a rationale for evaluating a frozen random residual network, but they do not imply that an untrained network is a learned financial feature extractor.

The framework examined here uses a randomly initialized DRN as a fixed nonlinear projection. Residual shortcuts preserve direct information paths within each block, while freezing prevents the coordinate system supplied to PPO from changing during policy optimization. This design concentrates all trainable capacity in the actor critic heads and exposes the policy to a fixed representation. Its limitations are equally important: the representation is not pretrained, is not optimized on market data, and can vary with random initialization.

The traditional comparison is a daily frequency adaptation of the fuzzy representation and deep direct reinforcement-learning approach introduced by Deng *et al*. (12). The adaptation uses the method's representation concept only; no table, figure, or numerical result from that publication is reproduced. The empirical analysis evaluates three linked questions: whether the residual random projection contributes beyond simpler representations, whether the learned categorical policy responds measurably to preference inputs, and whether any performance advantage remains stable across stronger baselines, independent training seeds, execution costs, market subperiods, and selected PPO hyperparameters.

METHODS AND MATERIALS

Data source and chronological partitioning

Daily BTC-USD open, high, low, close, and volume data were downloaded from Yahoo Finance through yfinance on July 28, 2026. The requested interval was January 1, 2021 through January 1, 2026; the final included market observation was December 31, 2025. No additional manual adjustment was applied beyond the behavior of the installed yfinance download function. Columns were flattened when returned as a multi index. Isolated missing observations were forward filled using only previously available information, and remaining incomplete rows were removed.

After the technical indicators were computed and leading lag induced missing values were dropped, the data contained 1,793 usable daily rows. The rows were divided chronologically into 70% training, 15% validation, and 15% testing partitions: 1,257 training rows, 268 validation rows, and 268 test rows. Fourteen day rolling windows produced 1,244 training states, 255 validation states, and 255 test states. The validation partition was used only for checkpoint selection, and the final testing partition was held out until evaluation.

For comparisons involving both the unified and traditional feature pipelines, the exact intersection of their test dates was used. This common interval ran from April 21 through December 31, 2025 and contained 255 prices and 254 realized return steps. All common date benchmark values were recomputed from one price sequence and one portfolio accounting function.

Feature engineering and preprocessing

Nine features represented direction, momentum, volatility, volume participation, and local channel position. Table 1 defines each feature. RSI, MACD difference, Bollinger-band width, ATR, and Donchian bands were computed with the ta package. The feature scaler was fitted exclusively on the training partition using StandardScaler; the resulting training means and standard deviations were then applied unchanged to validation and testing data.

Table 1. *Technical indicators used in the unified and raw indicator PPO models. All rolling quantities use only information available at or before day t.*

Feature	Definition
Return	$R_t = (P_t - P_{t-1})/P_{t-1}$, where P_t is the adjusted close.
RSI	Fourteen day relative strength index computed with the ta implementation.
MACD difference	Difference between the default 12/26 day MACD line and its 9 day signal line.
Bollinger width	Default 20 day Bollinger band width from the ta implementation.
ATR	Fourteen day average true range.
Bullish volatility	Fourteen day rolling standard deviation of $max(R_t, 0)$.
Bearish volatility	Fourteen day rolling standard deviation of $\lvert min(R_t, 0)\rvert$.
Log volume change	$log(1 + V_t) - log(1 + V_{t-1})$.
Donchian position	$(P_t - L_t^{(14)})/(H_t^{(14)} - L_t^{(14)} + 10^{-8})$, where $H_t^{(14)}$ and $L_t^{(14)}$ are the 14 day Donchian high and low.

Each trading state used a trailing 14 day feature window. Before convolution, the window was transposed into channel-by-time format. A batch of windows therefore had shape

$$X_{batch} \in R^{B\times 9\times 14},$$

where $B = 32$ during fixed representation extraction.

Fixed deep residual network

The DRN contained an initial one dimensional convolution from 9 to 16 channels, a 16-to-16 residual block, and a 16-to-32 residual block. Each residual block contained two kernel-size-3 convolutions with batch normalization and rectified linear units. A block was represented as

$$h_{\ell+1} = \sigma(F_\ell(h_\ell) + S_\ell(h_\ell)),$$

where S_ℓ was the identity when dimensions matched and a kernel-size-1 projection when channel dimensions changed. Adaptive average pooling collapsed the 14 day time dimension and produced a 32 dimensional vector z_t.

The DRN used the default PyTorch initialization. It was not pretrained and was not optimized with market data. After initialization, it was set to evaluation mode and applied once to the chronologically ordered training, validation, and testing windows under no_grad. The 32 dimensional outputs were then held fixed throughout PPO training. The representation should therefore be interpreted as a fixed nonlinear random projection rather than a learned financial embedding.

Multi-objective trading environment

At the start of each training episode, a profit preference weight was sampled uniformly:

$$w_{profit} \sim U(0, 1),\ w_{downside} = 1 - w_{profit}.$$

The same preference pair remained fixed throughout the episode. The PPO state concatenated the 32 dimensional DRN representation and the two weights:

$$s_t = [z_t, w_{profit}, w_{downside}] \in R^{34}.$$

The discrete actions were Short, Flat, and Long. Code values were mapped to position multipliers

$$m_t = a_t - 1 \in \{-1, 0, 1\}.$$

Let R_{t+1} be the next day BTC return, $\delta = 0.0005$ the transaction fee per action level turnover, and a_{t-1} the previous action. The profit component was

$$r_{profit,t} = m_t R_{t+1} - \delta|a_t - a_{t-1}|.$$

The second objective was

$$r_{downside,t} = min(0, m_t R_{t+1}).$$

This term penalizes one step negative position returns; it is not portfolio maximum drawdown. The scalar reward supplied to PPO was

$$r_t = w_{profit}\, r_{profit,t} + w_{downside}\, r_{downside,t}.$$

No explicit normalization was applied to the two reward components. A structural scale audit over the long and short directional returns in the training period found mean absolute magnitudes of 0.02194 for the gross directional profit term and 0.01097 for the downside return term, a ratio of approximately 2:1. This audit characterized the market data scale of the directional components; it did not represent the realized sequence of actions taken by the trained policy.

PPO policy and optimization

The actor and critic each used one 64 unit rectified linear hidden layer. The actor ended in a three way Softmax and defined a categorical distribution over the discrete actions:

$$\pi_\theta\,(a_t \mid s_t) = Categorical(p_t^{short},\, p_t^{flat},\, p_t^{long}).$$

During training, actions were sampled from this categorical distribution and PPO used the sampled action's log probability in the likelihood ratio policy gradient objective. During validation and testing, actions were selected deterministically by argmax. The action space remained discrete; only the probability vector varied continuously with the network output.

PPO used the clipped objective

$$L^{clip}(\theta) = E_t\,[min(\rho_t(\theta)\hat{A}_t,\, clip(\rho_t(\theta),\, 1-\epsilon,\, 1+\epsilon)\hat{A}_t)],$$

where $\rho_t(\theta)$ was the ratio of new to old action probabilities and $\epsilon = 0.20$. Full trajectory discounted Monte Carlo returns were used. Advantages were computed as returns minus rollout value estimates and then standardized. Generalized advantage estimation was not used, so a GAE parameter λ was not applicable.

The training episode contained 1,243 transitions, and the validation episode contained 254 transitions. Validation used the same preference pair as the corresponding training episode. Training always continued to 1,000 epochs even after the best checkpoint had been saved.

Experiments were executed in Google Colab with Python 3.12 and custom PyTorch PPO code. Logged packages included NumPy 2.0.2, pandas 2.2.2, yfinance 0.2.66, ta 0.11.0, Matplotlib 3.10.0, Gymnasium 1.3.0, Stable-Baselines3 2.9.0, and a PyTorch 2.x installation. The encoder-ablation, common-benchmark, and multi-seed runs selected CPU execution; the ordered-preference diagnostic selected a CUDA runtime. The exact GPU model was not recorded. Deterministic-algorithm mode was not enforced, so nominally identical seeds should not be interpreted as guarantees of bitwise identity across independent notebook executions.

Table 2. *PPO training and reproducibility configuration.*

Item	Value
Training epochs	1,000
Discount factor γ	0.99
GAE	Not used; λ not applicable
PPO update passes per trajectory	4
PPO clip coefficient ϵ	0.20
Learning rate	3×10^{-4}
Optimizer	Adam
Value loss coefficient	0.50
Entropy coefficient	0.01
PPO optimization batch	Full episode trajectory
Fixed representation extraction batch	32 windows; chronological; no shuffle
Validation frequency	Epoch 1 and every 10 epochs
Validation metric	Sum of scalar rewards over the validation episode
Checkpoint rule	Save when validation reward exceeds the previous best
Stopping rule	Fixed 1,000 epochs; no early stopping
Independent training seeds	12, 42, 45, 52, 67, 123, and 145
Initialization	PyTorch defaults for Linear, Conv1d, and BatchNorm layers

Baseline models

The traditional fuzzy baseline used 45 lagged daily returns and five momentum variables, producing 50 inputs. Each input was mapped to three learned Gaussian membership degrees, yielding a 150 dimensional fuzzy representation. Two PPO actor critic networks were trained separately: one used only the profit objective and one used only the downside objective. At inference, their action probability vectors were combined with the requested preference weights:

$p_t = w_{profit}\, p_t^{profit} + w_{downside}\, p_t^{downside}$, $a_t = \underset{a}{argmax} p_{t,a}$.

Both traditional agents used the same PPO learning rate, discount factor, clipping coefficient, update passes, training epochs, validation frequency, and checkpoint rule as the unified model. The unified model trained one policy network for 1,000 agent epochs and contained 4,740 trainable actor critic parameters, excluding the frozen DRN. The traditional ensemble trained two policy networks for 2,000 total agent epochs and contained 56,288 combined trainable parameters.

Additional controlled baselines were: 1) a preference conditioned PPO using the flattened 126 dimensional raw indicator window, 2) a single-objective profit PPO using the frozen 32 dimensional DRN state, 3) a fixed 20/50 day long-or-flat simple moving average crossover, and 4) Buy-and-Hold. The raw indicator and single-objective PPO models used 1,000 epochs and the same PPO hyperparameters and validation cadence.

Portfolio accounting and evaluation design

All common date strategies began with portfolio value $V_0 = 1$ and the Flat action. For strategy return $\tilde{R}_{t+1}$,

$$\tilde{R}_{t+1} = m_t R_{t+1} - (\delta + s)|a_t - a_{t-1}|,$$

where s was the slippage rate in sensitivity analyses. Portfolio value was

$$V_T = \prod_{t=0}^{T-1} (1 + \tilde{R}_{t+1})$$

Net profit was $100(V_T - 1)$. Annualized return was

$100(V_T^{252/T} - 1)$.

Maximum drawdown was the signed statistic

$$100 \min_t \left(\frac{V_t - \max_{\tau \le t} V_\tau}{\max_{\tau \le t} V_\tau} \right),$$

so more negative values indicate deeper losses. Sharpe and Sortino ratios were computed from daily net strategy returns with a $\sqrt{252}$ annualization factor.

Four analyses were performed. First, a seed 67 encoder ablation compared raw indicators, a fixed random linear projection, a fixed random non-residual CNN, the proposed fixed random DRN, and a jointly trained DRN. Second, the seed 67 unified checkpoint was evaluated on an ordered 21 point preference grid from 0.00 to 1.00 in increments of 0.05. Third, the unified model and stronger baselines were compared at the predeclared neutral preference (0.5, 0.5) on the common test interval. Fourth, unified and traditional models were retrained independently using seven seeds and evaluated on the ordered preference grid. The independent replication unit was the training seed; preference points were repeated evaluations of a trained policy and were not treated as independent runs.

The multi-seed analysis used 20,000 resample bootstrap confidence intervals across the seven seed level observations. Robustness tests reevaluated neutral preference action sequences at transaction fees of 0, 0.0005, 0.001, and 0.002; slippage charges of 0, 0.00025, 0.0005, and 0.001; three chronological out-of-sample subperiods; and seed 67 learning rate and PPO clipping alternatives.

RESULTS

Encoder ablation

Table 3 and Figure 1 summarize the seed 67 encoder ablation. Across 20 deterministic preference evaluations, the fixed random DRN produced the highest mean net

Table 3. *Seed 67 encoder ablation. Values summarize 20 preference evaluations of one trained policy per architecture; the preference evaluations are not independent training runs. Maximum drawdown (Max DD) is signed, and more negative values indicate deeper losses.*

Representation	Mean net profit (%)	Best net profit (%)	Mean Max DD (%)	Worst Max DD (%)	Mean Sharpe
Raw indicators	14.59	18.09	-5.15	-7.75	1.26
Fixed random linear projection	17.03	33.15	-22.12	-27.27	0.68
Fixed random non-residual CNN	3.03	25.83	-30.03	-43.07	0.24
Fixed random DRN	**26.15**	**56.94**	-11.61	-23.88	**1.29**
Jointly trained DRN	-14.81	-7.64	-26.27	-27.41	-0.39

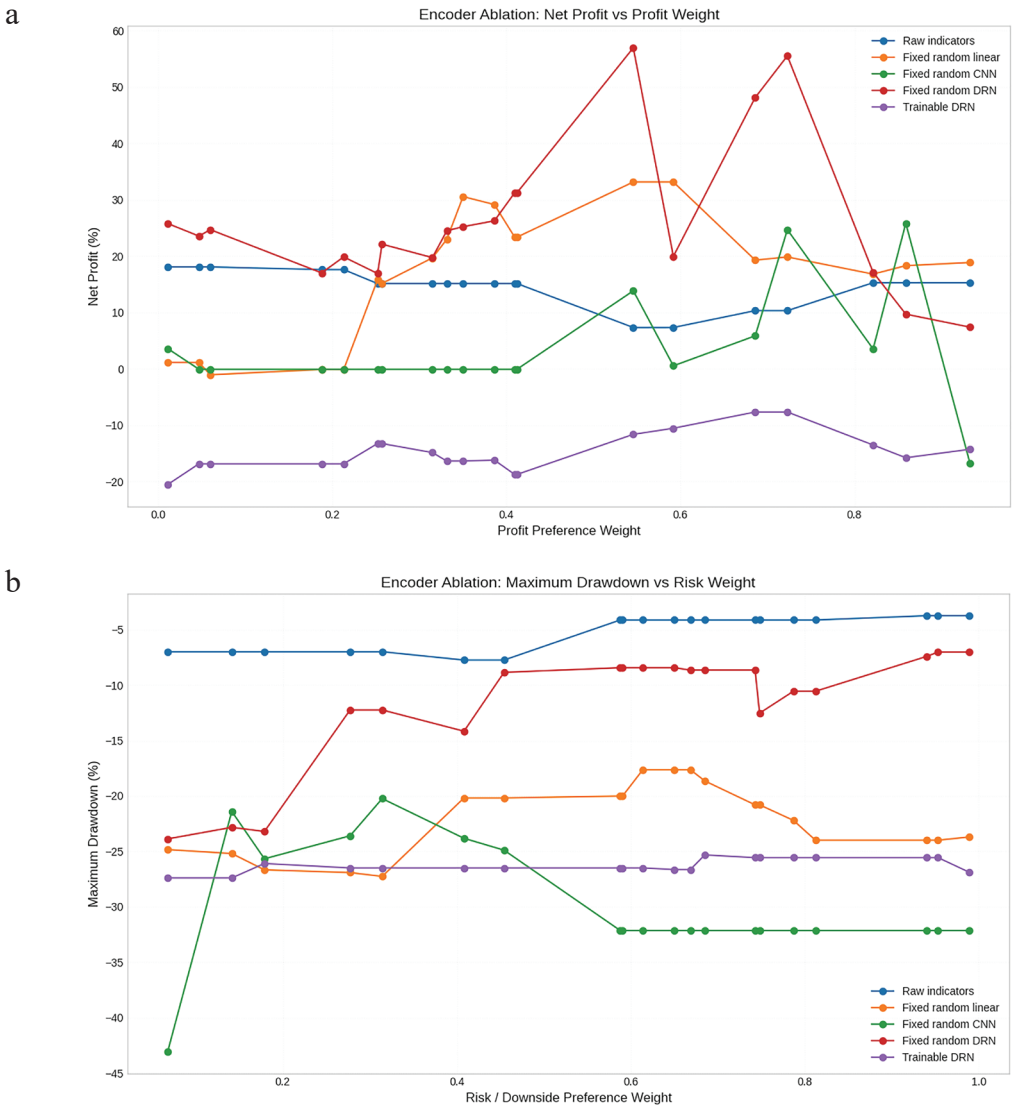

***Figure 1.* Encoder ablation over the seed 67 held out test period.** *(a) Net profit as a function of profit preference weight. (b) Maximum drawdown as a function of downside preference weight. All five conditions used the same BTC-USD data partitions, PPO hyperparameters, validation rule, initial portfolio value, and transaction fee of 0.0005. The fixed DRN produced the highest mean profit, whereas raw indicators produced the shallowest drawdown, followed immediately by the fixed DRN.*

profit (26.15%), followed by the fixed random linear projection (17.03%) and raw indicators (14.59%). The fixed random non-residual CNN produced 3.03%, and the jointly trained DRN lost 14.81% on average.

The drawdown ordering differed from the profit ordering. Raw indicators produced the shallowest mean maximum drawdown (-5.15%), followed by the fixed DRN (-11.61%). The fixed linear projection, trainable DRN, and non-residual CNN had deeper mean drawdowns of -22.12%, -26.27%, and -30.03%, respectively.

Reward scale and ordered preference response

The directional reward audit found a mean absolute magnitude of 0.02194 for the gross profit term and 0.01097 for the downside return term. The downside component was nonzero for half of the long-and-short directional cases because it penalized only negative directional returns. No normalization was applied.

When the same seed 67 checkpoint was evaluated at the ordered preference grid, net profit ranged from 12.77% to 38.96%. Maximum drawdown remained -12.25% at

all 21 weights. Action composition nevertheless changed across the grid. As increased from 0 to 1, the Short fraction decreased from 37.4% to 9.1%, the Flat fraction increased from 13.0% to 53.9%, and the Long fraction decreased from 49.6% to 37.0%.

Adjacent 0.05 changes in profit preference altered between 0% and 7.09% of deterministic actions. The mean total variation distance between adjacent categorical policy distributions ranged from approximately 0.00036 to 0.00047 and was nonzero at every test state. Thus, the underlying probability vector changed even across some intervals in which the argmax action remained unchanged.

Common date benchmark

Table 4 reports the controlled seed 67 benchmark over April 21 through December 31, 2025. The unified DRN-PPO model produced 41.60% net profit and -8.44% maximum drawdown. The 20/50 day moving average strategy produced 16.68% net profit, and the traditional fuzzy ensemble produced 2.80%. The raw indicator PPO lost 7.05%, the single-objective profit PPO was approximately flat, and Buy-and-Hold returned -0.06% after the common entry cost convention.

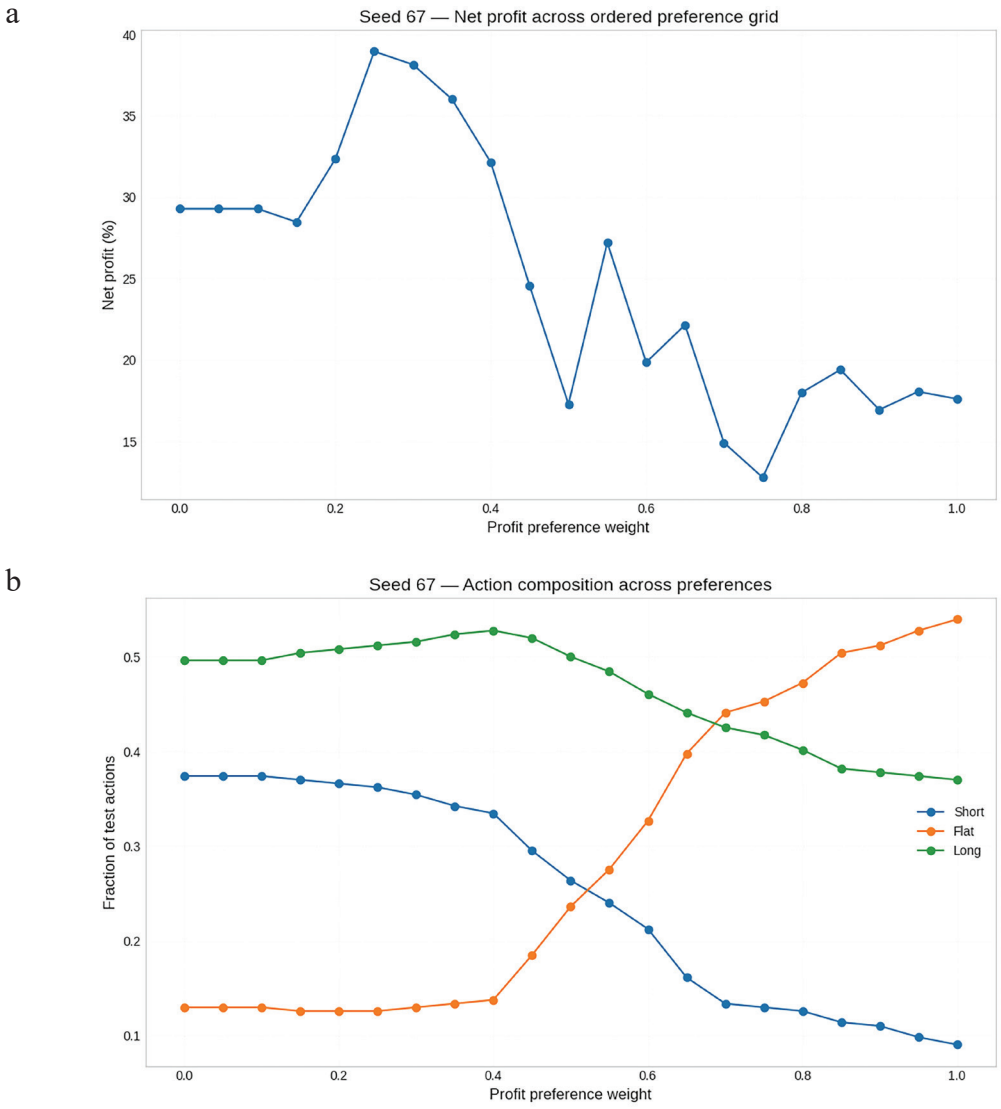

Figure 2. ***Ordered preference response of the seed-67 unified policy.*** *(a) Net profit for* $w_{profit} \in \{0, 0.05, \ldots, 1\}$*, with* $w_{downside} = 1 - w_{profit}$*. (b) Fractions of Short, Flat, and Long actions over the same grid. The same checkpoint was used at every weight; no retraining occurred between preference settings.*

Table 4. *Common date seed 67 benchmark. All methods used the same 255 BTC-USD prices, 254 return steps, initial portfolio value of 1.0, transaction fee of 0.0005 per action level turnover, zero slippage, and one metric implementation. Preference conditioned methods used* $(w_{profit}, w_{downside})=(0.5,0.5)$*. Annualized return uses 252 periods; Max DD is signed.*

Strategy	Net profit (%)	Annualized return (%)	Max DD (%)	Sharpe	Sortino	Profit factor
Unified DRN-PPO	**41.60**	**41.21**	-8.44	**1.84**	**2.49**	**1.60**
Traditional fuzzy ensemble	2.80	2.77	-9.49	0.31	0.19	1.21
Raw indicator PPO	-7.05	-7.00	-28.95	-0.22	-0.24	0.95
Single-objective profit PPO	0.04	0.04	-32.41	0.15	0.20	1.03
20/50 day SMA crossover	16.68	16.54	-14.66	0.82	0.96	1.21
Buy-and-Hold	-0.06	-0.06	-32.15	0.15	0.22	1.03

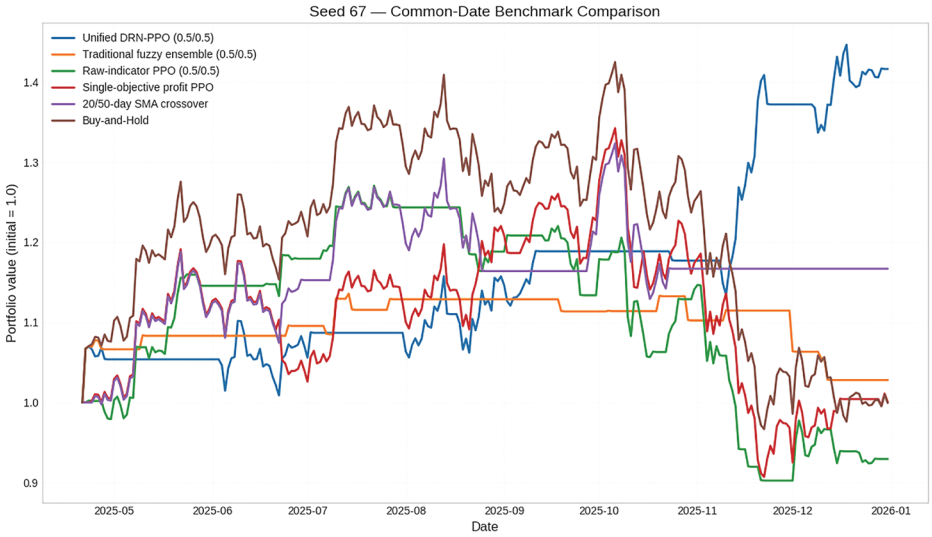

Figure 3. Common date seed 67 portfolio trajectories. *Every strategy begins at portfolio value 1.0 and uses the same BTC-USD close price sequence from April 21 through December 31, 2025. The unified, traditional, and raw indicator policies use the neutral preference (0.5, 0.5). Transaction cost is 0.0005 per action level turnover and slippage is zero. SMA denotes simple moving average.*

Seven seed robustness

Figure 4 shows means and 95% bootstrap confidence intervals across independently trained seeds 12, 42, 45, 52, 67, 123, and 145. The unified mean net profit curve was above the traditional mean over most of the preference grid, but the confidence bands were wider at lower preference weights. At the neutral preference, the paired mean net profit difference was 7.73 percentage points in favor of the unified model, with a 95% bootstrap confidence interval of -4.78 to 21.07.

The maximum drawdown result did not favor the unified model. At the neutral preference, the paired difference in maximum drawdown (Unified minus Traditional) was -11.44 percentage points, with a 95% bootstrap confidence interval of -23.04 to 2.08. The negative sign indicates deeper drawdown for the unified method.

Averaging each independently trained seed across its 21 preference evaluations, the unified model had mean net profit of 3.02% across seeds, compared with -3.44% for the traditional method. Mean maximum drawdown was -20.50% for the unified method and -12.68% for the traditional method. Seed level results are reported in Table 5.

Table 5. *Seed level summaries across the ordered 21 point preference grid. Each value is first averaged across preference evaluations within one trained seed. The final row reports the mean and sample standard deviation of those seed level means. Preference evaluations are not counted as independent replications.*

Seed	Unified mean profit (%)	Traditional mean profit (%)	Unified mean Max DD (%)	Traditional mean Max DD (%)
12	-0.56	5.58	-2.77	-13.02
42	0.96	-29.86	-15.84	-31.41
45	3.83	-0.85	-25.79	-3.59
52	14.40	3.94	-20.99	-10.80
67	-2.01	3.17	-27.01	-13.71
123	5.68	5.26	-25.04	-3.96
145	-1.18	-11.29	-26.03	-12.23
Across seeds, mean ± SD	**3.02 ± 5.73**	**-3.44 ± 13.05**	**-20.50 ± 8.73**	**-12.68 ± 9.26**

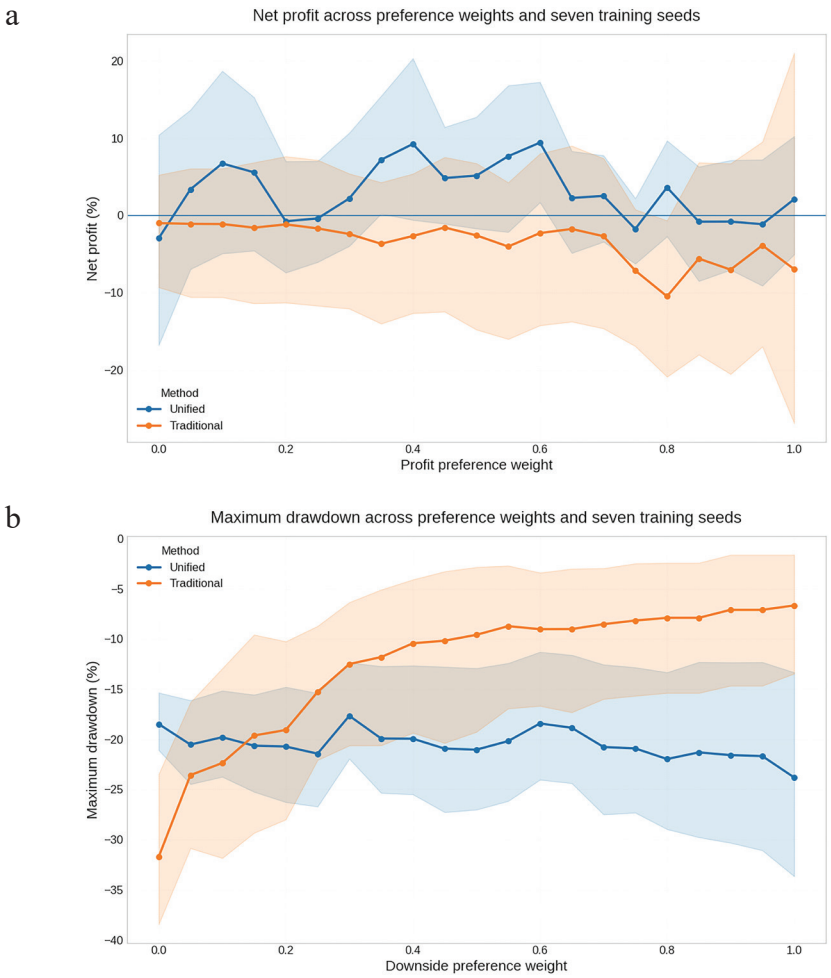

Figure 4. ***Preference responses across seven independently retrained seeds.*** *(a) Net profit versus profit preference weight. (b) Maximum drawdown versus downside preference weight. Points show means across seeds and shaded bands show 95% bootstrap confidence intervals. Each trained model was evaluated at 21 ordered preferences on the same held out dates. More negative maximum drawdown indicates a deeper loss.*

Transaction costs, slippage, market periods, and hyperparameters

Neutral preference action sequences from each seed were reevaluated at transaction fees of 0, 0.0005, 0.001, and 0.002. Mean profit declined approximately linearly for both methods as the fee increased. The unified mean remained above the traditional mean at the tested fee levels, but the bootstrap intervals were wide. Slippage entered the accounting equation as an additional symmetric per turnover charge. Under this simplified implementation, increasing slippage produced the same mathematical effect as increasing the transaction fee by the same amount.

The three chronological test subperiods produced different results. All three subperiods were drawn exclusively from the held out test partition and were not used for training, validation, checkpoint selection, or parameter tuning. The unified method was negative on average from April 21 to July 14, 2025, slightly positive from July 15 to October 7, and strongly positive from October 8 to December 31. The traditional method remained closer to zero and was negative in the first and third subperiods. Confidence intervals were broad in all three periods.

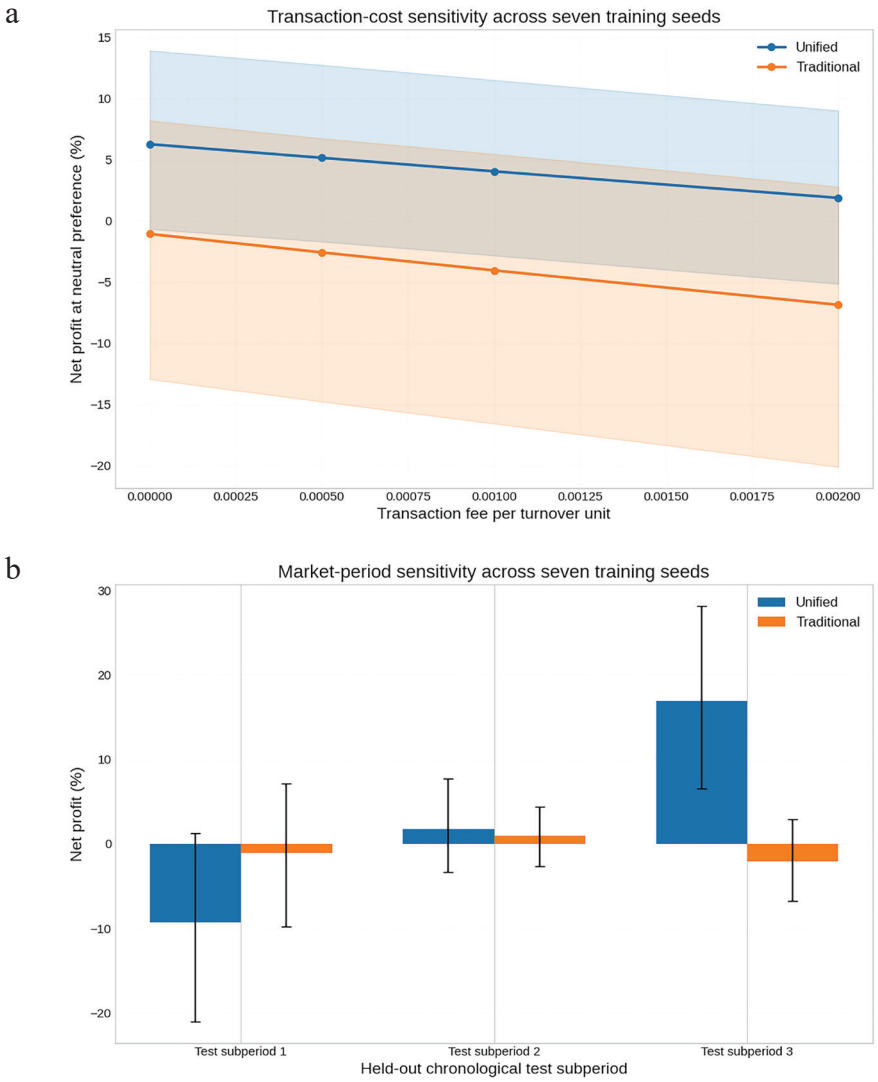

Figure 5. ***Execution cost and market period sensitivity across seven independent seeds.*** *(a) Mean neutral preference net profit under transaction fees of 0, 0.0005, 0.001, and 0.002 per action level turnover. Shaded bands are 95% bootstrap confidence intervals. (b) Mean net profit in three non overlapping held out periods: April 21-July 14, July 15-October 7, and October 8-December 31, 2025. Error bars are 95% bootstrap confidence intervals.*

Table 6. *Seed 67 hyperparameter sensitivity at the neutral preference. Each scenario was retrained from scratch. Max DD is signed.*

Scenario	Learning rate	Clip ∈	Method	Net profit (%)	Max DD (%)	Sharpe	Sortino
Baseline	3×10^{-4}	0.20	Unified	6.85	-27.16	0.37	0.54
Baseline	3×10^{-4}	0.20	Traditional	8.03	-9.35	0.71	0.52
Lower learning rate	1×10^{-4}	0.20	Unified	12.20	-15.28	0.57	0.73
Lower learning rate	1×10^{-4}	0.20	Traditional	-12.06	-26.84	-0.48	-0.58
Higher learning rate	1×10^{-3}	0.20	Unified	-1.69	-8.77	-0.11	-0.07
Higher learning rate	1×10^{-3}	0.20	Traditional	0.00	0.00	0.00	0.00
Lower PPO clip	3×10^{-4}	0.10	Unified	6.03	-10.16	0.46	0.49
Lower PPO clip	3×10^{-4}	0.10	Traditional	-6.81	-15.82	-0.40	-0.33
Higher PPO clip	3×10^{-4}	0.30	Unified	-8.62	-23.60	-0.41	-0.35
Higher PPO clip	3×10^{-4}	0.30	Traditional	8.03	-9.35	0.71	0.52

The one factor at a time seed 67 analysis also showed marked hyperparameter sensitivity (Table 6). Lowering the learning rate improved the unified run and harmed the traditional run. Raising the learning rate caused the unified run to lose money and the traditional run to select Flat throughout the evaluated interval. Lowering the PPO clip coefficient favored the unified run in this seed, whereas increasing it favored the traditional run.

DISCUSSION

Contribution and limits of the fixed DRN

The seed 67 ablation provides evidence that the fixed residual transformation contributed to profit in that controlled architecture comparison. The fixed DRN exceeded the fixed linear projection, non residual CNN, raw indicators, and jointly trained DRN in mean profit. This result is compatible with the random feature rationale: a nonlinear randomized basis can support a trained downstream model without being jointly optimized (9, 10). The comparison between the residual and non-residual random convolutional models also suggests that the shortcut structure mattered in this particular run.

The evidence is not sufficient to describe the DRN as a learned financial feature extractor. Its weights were never trained on market data. Moreover, raw indicators produced shallower drawdowns than the fixed DRN, and the ablation used only one independent training seed. The appropriate interpretation is that the residual random projection was useful for profit in one controlled experiment, not that it universally improved representation quality or downside control.

The poor jointly trained DRN result is consistent with the possibility that simultaneous encoder and policy optimization can be difficult in this setting, but the study did not measure latent state drift or training stability directly. It therefore cannot establish that freezing the encoder stabilized reinforcement learning.

Preference sensitivity and discrete action plateaus

The ordered preference analysis shows that the policy used its conditioning input. Action composition changed across the grid, deterministic actions changed at some adjacent weights, and the categorical probability vector changed at every adjacent pair. However, probability changes were small, selected actions were piecewise constant, realized profit was non-monotone, and maximum drawdown did not change across the seed 67 grid.

These patterns follow naturally from the three action policy. A continuously changing probability vector can remain on the same side of the argmax boundary over several adjacent weights, producing identical trades until a decision threshold is crossed. The results support the term *preference sensitive categorical policy*. They do not support the stronger descriptions *continuous risk return spectrum* or *Pareto frontier*.

The unchanged maximum drawdown also reflects an objective mismatch. The optimized downside term penalizes one step negative position returns, whereas maximum drawdown is a path dependent portfolio statistic. A preference weight on the former does not directly constrain the latter.

Baseline comparison and cross seed uncertainty

The common date benchmark demonstrates that one unified seed 67 run can outperform a broader set of RL and non-RL baselines under a matched accounting protocol. The seven seed experiment demonstrates why that isolated result cannot be generalized. The paired profit difference favored the unified method on average, but the confidence interval included zero. The unified model also had deeper drawdowns descriptively across the seven seed preference grid summaries.

This contrast is consistent with the model selection and backtest overfitting concerns described in financial evaluation research (13). A strong isolated run can coexist with high uncertainty across random initialization, market subperiod, and hyperparameter setting. Repeated validation checkpoint selection and inspection of several architectures and settings can further bias the apparent best configuration.

The traditional baseline should not be interpreted as a general rejection of ensemble reinforcement learning. It used a daily adaptation of a representation originally designed around higher frequency information (12), and its complete feature pipeline differed from the unified model. It also required two separately trained networks and post hoc probability blending. These differences are part of the complete system comparison, but they prevent attribution of every performance difference to preference conditioning alone.

Practical implications and limitations

The frozen encoder design reduces the number of trainable policy parameters and allows one policy to accept many preference inputs without retraining. This can simplify storage and inference compared with maintaining multiple objective specific policies. The empirical results do not justify interpreting the preference weight as a guarantee of realized maximum drawdown. A deployment system would require direct path dependent risk constraints, position sizing, realistic execution modeling, and ongoing regime monitoring.

The study has several limitations that constrain the interpretation and generalizability of the findings. First, the empirical analysis was limited to BTC-USD and to a common test interval spanning April through December 2025. The subperiod analysis showed that performance varied materially across market regimes, so the reported results may not generalize to other cryptocurrencies, asset classes, or time periods. The study also involved repeated inspection of architectures, checkpoints, preference settings, and hyperparameters, and no formal probability-of-backtest-overfitting correction was applied (13). In addition, the same validation interval was examined every 10 epochs and used for checkpoint selection, which may have introduced model-selection bias and increased the risk that the selected configurations were partially tailored to that validation period.

Several limitations also arise from the model design and training procedure. The DRN was a randomly initialized nonlinear projection rather than a learned financial encoder, and its performance remained sensitive to initialization. Deterministic PyTorch algorithms were not enforced, so separate notebook executions using the same nominal seed may not be bitwise identical. The optimized downside objective was a one-step negative-return penalty rather than portfolio maximum drawdown, which limits the extent to which the preference weight can be interpreted as directly controlling path-dependent downside risk. The previous action was also omitted from the state even though transaction costs depended on position changes, so the policy did not explicitly observe all information relevant to turnover costs.

The execution and evaluation framework was simplified. Slippage was represented as a symmetric per-turnover charge, while bid–ask spreads, order-book depth, liquidity, capacity constraints, and market impact were omitted. These assumptions may overstate real-world implementability, particularly during volatile or illiquid periods. Performance metrics were annualized using 252 periods even though cryptocurrency trades continuously, which may affect comparability with other studies using 365-day conventions. The traditional Deng-style fuzzy representation was adapted from intraday concepts to daily data, creating a frequency mismatch that may have disadvantaged that baseline. Finally, the ordered preference points were not filtered for nondominance, and no hypervolume or coverage metric was calculated. The results should therefore be interpreted as a preference-response analysis rather than as evidence of a formally established Pareto frontier.

CONCLUSION

This study evaluated a preference conditioned PPO policy supplied with a fixed random residual representation for daily BTC-USD trading. In a controlled seed 67 encoder ablation, the fixed DRN produced the highest mean profit among five representation conditions, but raw indicators produced shallower drawdowns. The ordered preference analysis showed measurable changes in policy probabilities and action composition, while

deterministic actions remained piecewise constant and realized maximum drawdown was not calibrated by the downside weight.

A matched seed 67 benchmark strongly favored the unified model, but seven independent retrainings produced wide uncertainty. The neutral preference profit difference favored the unified method on average, yet its confidence interval included zero, and the unified model exhibited deeper drawdowns descriptively. Transaction costs, market subperiods, and PPO hyperparameters materially affected results.

The evidence therefore does not establish that fixed residual representations stabilize reinforcement learning, that the proposed framework is generally superior, or that it generates a continuous risk return frontier. It supports a narrower empirical conclusion: a fixed residual random projection can support profitable preference conditioned policies in some Bitcoin backtests, while performance remains highly dependent on initialization, market regime, execution assumptions, and optimization settings. Future work should use multiple assets, walk forward evaluation, direct path dependent risk objectives, deterministic replication controls, realistic market impact simulation, and formal backtest overfitting analysis.

CONFLICT OF INTEREST

The author declares that there are no conflicts of interest related to this work.

REFERENCES

1. Lo AW. The adaptive markets hypothesis: market efficiency from an evolutionary perspective. *J Portf Manag.* 2004; 30 (5): 15-29. https://doi.org/10.3905/jpm.2004.442611
2. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv:1707.06347. 2017.
3. Yang H, Liu XY, Zhong S, Walid A. Deep reinforcement learning for automated stock trading: an ensemble strategy. In: *Proceedings of the First ACM International Conference on AI in Finance*; 2020. doi:10.1145/3383455.3422540.
4. Jiang Z, Xu D, Liang J. A deep reinforcement learning framework for the financial portfolio management problem. arXiv:1706.10059. 2017.
5. Hayes CF, Radulescu R, Bargiacchi E, *et al*. A practical guide to multi-objective reinforcement learning and planning. *Auton Agents Multi-Agent Syst.* 2022; 36: 26. doi:10.1007/s10458-022-09552-y.
6. Nguyen TT, Nguyen ND, Vamplew P, Nahavandi S, Dazeley R, Lim CP. A multi-objective deep reinforcement learning framework. *Eng Appl Artif Intell.* 2020; 96: 103915. doi:10.1016/j.engappai.2020.103915.
7. Reymond M, Bargiacchi E, Nowe A. Pareto Conditioned Networks. In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*; 2022. p. 1110-1118. https://doi.org/10.65109/NPXB3014
8. Wang Z, Yan W, Oates T. Time series classification from scratch with deep neural networks: a strong baseline. In: *2017 International Joint Conference on Neural Networks*; 2017. p. 1578-1585. doi:10.1109/IJCNN.2017.7966039.
9. Rahimi A, Recht B. Random features for large-scale kernel machines. In: *Advances in Neural Information Processing Systems 20*; 2007. p. 1177-1184.
10. Dempster A, Petitjean F, Webb GI. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min Knowl Discov.* 2020; 34: 1454-1495. doi:10.1007/s10618-020-00701-z.
11. Gradojevic N, Kukolj D, Adcock R, Djakovic V. Forecasting Bitcoin with technical analysis: a not-so-random forest? *Int J Forecast.* 2023; 39 (1): 1-17. doi:10.1016/j.ijforecast.2021.08.001.
12. Deng Y, Bao F, Kong Y, Ren Z, Dai Q. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Trans Neural Netw Learn Syst.* 2017; 28 (3): 653-664. doi:10.1109/TNNLS.2016.2522401.
13. Bailey DH, Borwein JM, Lopez de Prado M, Zhu QJ. The probability of backtest overfitting. *J Comput Finance.* 2017; 20 (4): 39-69. doi:10.21314/JCF.2016.322.