QF634 Applied Quantitative Research Methods (AY2021/2022)

Research Project: Deep Reinforcement Learning in Stock Trading

Prepared By:
Brandon Lee
Dani Surya Pangestu
Gabriel Tan
Gabriel Woon
Leon Cai
Nicole Lim
Pak Lam Chan

Table of Contents

Abstract

Deep reinforcement learning has been envisioned to have a competitive edge in quantitative finance. There were multiple studies conducted previously that looked at the performance of various individual algorithms and an ensemble of strategies on multi-stock trading with a small number of stocks (usually the Dow Jones Industrial Average of 30 stocks). However, there was no research conducted to compare the different algorithms and ensemble strategy in a larger dataset.

Hence, the efficacy of applying various algorithms and the ensemble strategy to trade the underlying constituents of the Nasdaq-100 Index will be studied and their performances will be measured relative to one another, the index benchmark and the traditional minimum-variance portfolio allocation strategy. In addition, the algorithms and ensemble strategy will be tested on different trading environments to test their robustness.

1 Introduction

With the advent of technology, trading has evolved from an open outcry format where one communicated trade orders through shouting and gesturing with hand signals, to the faster and more accurate electronic order systems nowadays that involve less human intervention. More recently, advancements in artificial intelligence (AI) have enabled quantitative hedge funds to carry out trading with even less reliance on human intervention. New techniques are applied in machine learning and deep reinforcement learning to crunch more data, analyse markets, and ultimately achieve an alpha, all at an astonishing speed.

Although few papers report on the performance of actual strategies deployed by hedge funds due to proprietary reasons as explained in Li (2017), many open-source code packages and deep reinforcement learning environments are available to experiment on.

TABLE I
SUMMARY OF ALGORITHMS AND TRADING ENVIRONMENTS

Trading Environments		
Normal (Control Env)	**Cash Penalty (Less Strict Env)**	**Stop Loss (Stricter Env)**
DDPG	DDPG	DDPG
TD3	TD3	TD3
PPO	PPO	PPO
A2C	A2C	A2C
SAC	SAC	SAC
Ensemble Strategy		

This paper aims to compare and contrast the 5 latest and broadly adopted algorithms (Deep Deterministic Policy Gradient [DDPG], Twin Delayed Deep Deterministic Policy Gradient [TD3], Proximal Policy Optimization [PPO], Advantage Actor-Critic [A2C] and Soft Actor-Critic [SAC]) and the ensemble strategy on various trading environments (normal, cash penalty and stop loss) to trade the Nasdaq-100 index constituents. The performances are then measured relative to one another, the index benchmark and traditional minimum-variance portfolio allocation strategy.

2 Literature Review

2.1 OpenAI Five

OpenAI Five became the first AI system to defeat the world champions of the game Dota 2 in 2019. Dota 2 was studied because it presented key challenges for AI systems such as long-time horizons, imperfect information, and complex, continuous state-action spaces, that were eventually overcome. OpenAI Five leveraged on PPO to learn from batches of approximately 2 million frames every 2 seconds. By having a winning rate of over 90% for all the public games played and successfully defeating the Dota 2 world champion “Team OG”, OpenAI Five demonstrated that self-play deep reinforcement learning can achieve superhuman performance on a difficult task (Berner et al., 2019).

The success of OpenAI Five highlighted deep reinforcement learning techniques’ ability to solve complex sequential decision-making problems. This brings to mind the application of deep reinforcement learning to the complex and dynamic stock market that generally have a sequential nature and are highly stochastic, with an environment partially observable and potentially adversarial (Théate & Ernst, 2021). Furthermore, deep reinforcement learning is perfectly aligned with trading objectives such as maximizing cumulative rewards of expected returns (Wilcox, 2019).

2.2 Deep Reinforcement Learning Algorithms

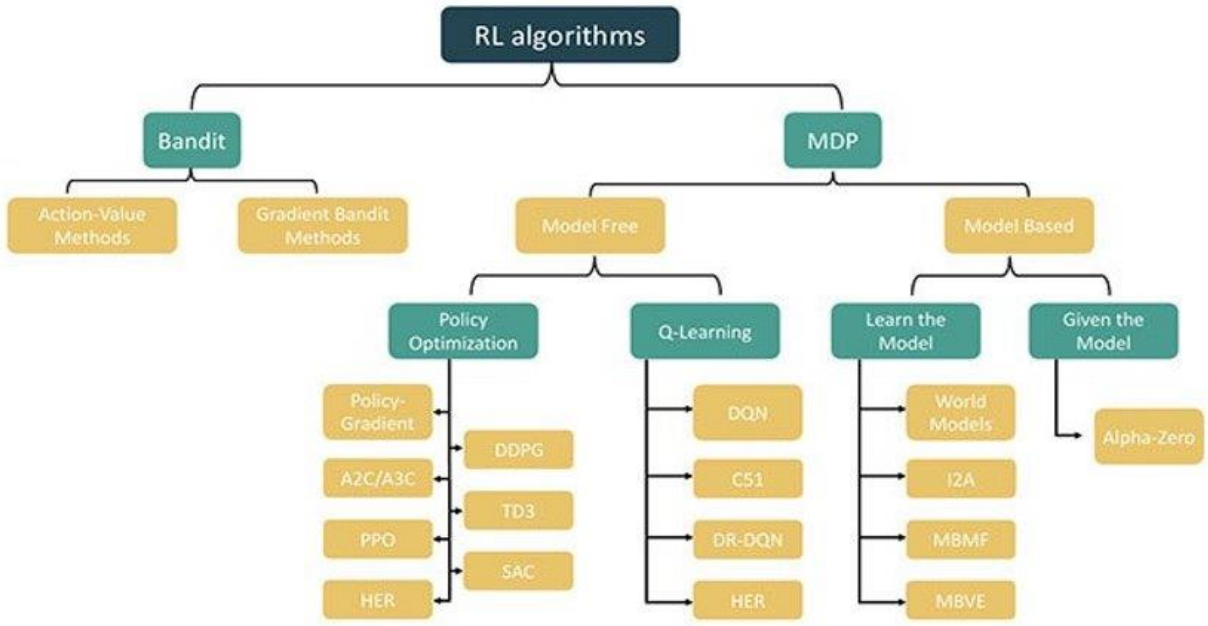

Fig 1: Hierarchy Table of Deep Reinforcement Learning Algorithms
Source: https://substance.etsmtl.ca/en/exploring-maze-reinforcement-learning

As shown in the diagram above, many different deep reinforcement learning algorithms are suitable for different situations. For example, a multi-armed bandit process is suitable for situations that are stateless and have a fixed reward. On the other hand, the Markov decision process is suitable for situations that have states and each reward influences future rewards. This paper will focus on model-free algorithms under the Markov decision process.

Markov Decision Process (MDP)

MDP is the fundamental framework used in deep reinforcement learning where it is used for modelling a sequential decision-making process under uncertainty (i.e., outcomes are partially stochastic and partially under the control of the decision-maker).

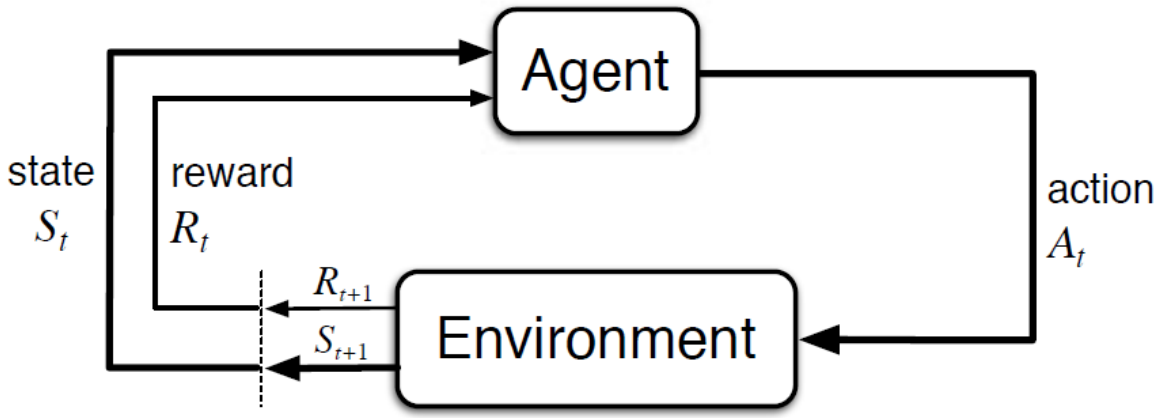

Fig 2: Markov Decision Process
Source: https://www.analyticsvidhya.com/blog/2020/11/reinforcement-learning-markov-decision-process

MDPs are defined as controlled stochastic processes satisfying the Markov property and assigning reward values to state transitions. It relies on the notions of state, describing the current situation of the agent, action (or decision), affecting the dynamics of the process, and reward, observed for each transition between states. Such a process describes the probability of triggering a transition to the next state and receiving a certain reward when making a decision/taking an action in the current state (Sigaud & Buffet, 2013). In addition, the transition probability from the current state to when the agent is to make the next decision is only dependent on the current state and not on earlier states (White & White, 1989).

In a MDP framework, the agent acts in the environment which will influence the environment that will subsequently bring the agent to a new state. In return, the environment will return some information to the agent, the next state and the corresponding reward.

As such, MDP is used to model the stochastic nature of the dynamic stock market with portfolio value maximization as the trading goal (Yang et al., 2020). In fact, MDP has been used before on the stock market (Chang & Lee, 2017), but with different algorithms from the ones used in this paper.

2.2 Model Free Deep Reinforcement Learning

2.2.1 Value-Based Approach (Q-Learning)

The value-based learning approach solves a discrete action space problem and trains an agent on a single stock or asset, predicting the number of shares, action strategies, and transfer learning (Jeong & Kim, 2019). The value-based approach tries to learn the optimal policy through state-action value function (Q-functions) aimed to maximize the expected future reward given the current state (Neuneier, 1996). The major limitation of the value-based learning approach is that it only works with discrete and finite state and action spaces, it is hence not practical for a portfolio of stocks and because prices are continuous.

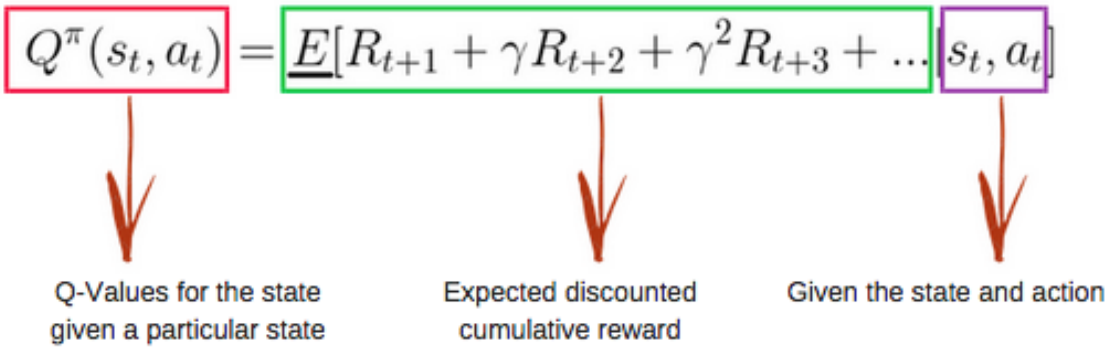

Fig 3: Q-function Equation
Source: https://www.freecodecamp.org/news/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc/

As shown above, a Q-function uses the Bellman equation taking two inputs which are the states(s) and actions, then returning the output (maximum future reward) which is the reward that the agent received for entering the current state s in addition to the maximum future reward for the next state s'.

TABLE II
VARIOUS IMPROVEMENTS IMPLEMENTED ON Q-FUNCTION

Q-Learning Network	Deploy "state–action–reward–state" function by using artificial neural networks to determine the maximum Q functions (Watkins & Dayan, 1992).
Deep Q-Learning (DQN)	Combines Q-learning with deep neural networks to allow for complex high-dimensional environments such as video games and robotics. Aim to minimize the error between estimated Q-value and target Q-value over a transition and use convolutional neural networks (CNN) to perform function approximation (Chen et al., 2019).
Double DQN	Q-learning can sometimes overestimate the action values, causing the agent to become bias and slowing the learning. Hence, two separate value functions are trained in a mutually symmetric fashion using separate experiences to resolve this (Hasselt et al., 2016).
Dueling DQN	Splits the neural networks into two parts. One learns to estimate the value at every state whereas the other calculates the potential advantages of each action. Both parts are then combined to form a single action-advantage Q function. (Wang et al., 2016).

2.2.2 Policy-Based Approach (Policy Gradient)

The policy-based approach trains an agent to directly learn the optimal policy itself by using deep neural networks to approximate the optimal policy (Sutton et al., 2000). Policy gradient allows stochastic policies to be factored in which handles the exploration/exploitation trade-off as it outputs a probability distribution over actions resulting in a possibility of different actions for the same/similar states. Another advantage of policy gradient is that it has better convergence properties compared to a value-based approach as it is guaranteed to converge on local maximum or global maximum.

However, one of the biggest disadvantages of the policy-based approach is that as it tends to follow the policy gradient, it will usually converge to a local maximum instead of the optimal value.

2.2.3 Actor-Critic Approach

The actor-critic approach incorporates the advantages of value-based and policy-based approaches. It utilizes both the actor neural network and the critic neural network. The actor-network is used to train the agent to learn the optimal policy whereas the critic-network is used to estimate the state-action function. Both actor and critic networks can be trained simultaneously and over time actor learns how to make better trading decisions and the critic learns how to evaluate the trading decision.

Deep Deterministic Policy Gradient (DDPG)

DDPG combines both the frameworks of Q-learning and policy gradient by using neural networks as function approximators. In contrast with DQN that learns indirectly through Q-values tables and suffers the curse of dimensionality problem, DDPG learns directly from the observations through policy gradient. It is proposed to deterministically map states to actions to better fit the continuous action space environment (Yang et al., 2020). It requires only a straightforward actor-critic architecture and learning algorithm with very few “moving parts”, making it easy to implement and scalable to more complex problems and larger networks (Lillicrap et al., 2016).

Twin Delayed Deep Deterministic Policy Gradient (TD3)

One of DDPG algorithm weaknesses is that it is unstable and heavily dependent on getting the correct hyperparameters for a particular task. So TD3 was created to overcome these pitfalls of DDPG algorithm (Lillicrap et al., 2016). TD3 draws inspiration from the previous Double DQN (Hasselt, 2010) algorithm where it takes the smallest value of two critic networks to prevent overestimation of Q value. Preventing the overestimation of Q value will improve the stability of the algorithm. It also has a delayed update where the policy network updated much less frequently than the value network to give a less varying value.

Proximal Policy Optimization (PPO)

With the policy gradient approach, if the step size was too small, the training process would be too slow, and if the step size was too high, there would be too much variability in the training. PPO deals with this problem by limiting the policy update at each training step through the introduction of the clipped surrogate objective function. This function constrains the policy change in a small range using a clip, which would avoid the problem of having too large a policy update, by using a probability ratio between the old policy and the new policy and clipping the ratio between 0.8 and 1.2. This improved the stability of the algorithm, decreased the variance, and was simple to implement and tune (Schulman et al., 2015).

Advantage Actor-Critic (A2C)

A2C is derived from the Asynchronous Advantage Actor-Critic (A3C) model developed by Google Deepmind. The A3C implements parallel training where multiple agents update gradients independently with different data samples in the same parallel environment (Mnih et al., 2016). After all the agents have finished updating the gradients, a coordinator passes the average gradients to a global network, which then increases the diversity of the training data. This makes A3C more stable, faster, and more cost-effective, as well as allowing for larger batch sizes. The A2C is the same as A3C, except that it is synchronous. Based on studies conducted, the OpenAI team found that their synchronous A2C implementation performed better than A3C and did not observe any evidence that noise introduced by asynchrony provided additional benefits (Wu et al., 2017).

Soft Actor-Critic (SAC)

SAC was created by researchers from UC Berkeley where its purpose is to improve its predecessor by being more efficient with the usage of samples and improve the robustness of its result to combat brittleness in convergence (Haarnoja et al, 2018). SAC modified the objective of deep reinforcement learning by adding entropy maximization in addition to the traditional reward maximization. The higher entropy encourages the model to do more exploration and it forces the model not to put a very high probability in a part of the action. The second benefit is as it prevents the model to select only a few sets of action, it prevents the model to exploit the inconsistency of the Q function.

3 Methodology

3.1 Stock Data Pre-Processing

The daily stock data (open, high, low, close and volume) from "Yahoo! Finance" for the constituents of Nasdaq-100 (as of 1st Jan 2019) was used for the period 1st Jan 2000 to 1st Oct 2021. The technical and risk indicators are added to the database (more information about the indicators in the next section). Any incomplete stock data (due to delisting or listing halfway) was not included, resulting in 66 stocks remaining. Lastly, the data was split into in-sample training set (1st Jan 2000 to 1st Jan 2020 to train on the Dot Com Bubble and Global Financial Crisis) and out-of-sample trading set (1st Jan 2020 to 1st Oct 2021 to trade on the COVID19 Crisis).

3.2 Environment Design

The state space, action space and stock trading constraints are similar for all the 3 different trading environments. Only the reward function will differ for each of them.

State Space

A 661-dimensional vector (66*10 + 1) consists of 10 parts of information to represent the state space of multiple stock trading environment:
$[b_t, p_t, h_t, M_t, R_t, C_t, X_t, B_t, SMA30_t, SMA60_t]$

Each component is defined as follows:

$b_t \in R_+$: available balance at current time step t.

$p_t \in R_+^{66}$: adjusted close price of each stock.

$h_t \in Z_+^{66}$: shares owned of each stock.

$M_t \in R^{66}$: Moving Average Convergence Divergence (MACD) is calculated using close price. MACD is one of the most used momentum indicators that identifies moving averages (Chong et al., 2014).

$R_t \in R_+^{66}$: Relative Strength Index (RSI) is calculated using close price. RSI quantifies the extent of recent price changes. If the price moves around the support line, it indicates the stock is oversold, and we can perform the buy action. If the price moves around the resistance, it indicates the stock is overbought, and we can perform the selling action (Chong et al., 2014).

$C_t \in R_+^{66}$: Commodity Channel Index (CCI) is calculated using high, low, and close prices. CCI compares current price to average price over a time window to indicate a buying or selling action (Maitah et al., 2016).

$X_t \in R^{66}$: Average Directional Index (ADX) is calculated using high, low, and close price. ADX identifies trend strength by quantifying the amount of price movement (Gurrib, 2018).

$B_t \in R^{66}$: Bollinger Bands (BB) is calculated using close price. BB comprise a volatility indicator that measures the relative high or low of a security's price in relation to previous trades. Volatility is measured using standard deviation, which changes with increases or decreases in volatility. The bands widen when there is a price increase, and narrow when there is a price decrease (Bollinger, 2002).

$SMA30_t$ and $SMA60_t \in R^{66}$: Simple Moving Average (SMA) of 30 days and 60 days respectively.

Action Space

For a single stock, the action space is defined as {−*k*, ..., −1, 0, 1, ..., *k*}, where *k* and −*k* presents the number of shares the agent can buy or sell, and $k \leq h_{max}$ (h_{max} is a predefined parameter set at 100 that acts as the maximum number of shares for each buying or selling action). Therefore, the size of the entire action space is $(2*100 + 1)^{66}$.

Stock Trading Constraints

The following assumption and constraints reflect concerns for practice: market liquidity, nonnegative balance, transaction costs and risk aversion etc.

Market liquidity: the orders can be rapidly executed at the close price. It is assumed that the stock market will not be affected by our reinforcement trading agent.

Nonnegative balance $b_t > 0$: the allowed actions should not result in a negative balance and short selling.

Transaction cost: to account for the different types of transaction costs such as exchange fees, execution fees, SEC fees, brokerage commission fees and slippage costs. It is assumed to be 0.1% of the value of each trade (either buy or sell).

Risk-aversion for market crash: due to events that may cause sudden shock to the stock market such as wars, the collapse of stock market bubbles, sovereign debt default, and financial crisis, the VIX index is employed to control the risk in a worst-case scenario like Global Financial Crisis. It measures the stock market's expectation of volatility based on S&P 500 index options

When the VIX index is higher than a threshold, which indicates extreme market conditions, the agent will halt buying and immediately sell all shareholdings. This minimizes the negative change of the portfolio value by selling all held stocks because during a bear market. As all stock prices are expected to fall and the agent is unable to capitalise during a bear market by short selling, it is illogical for the agent to take any buy action (i.e., trying to catch a falling knife). The agent resumes trading once the VIX index returns under the threshold.

The VIX index should not be included in training because it is not a part of model training, only the trading environment should include the risk aversion signal.

4 Normal Environment

4.1 Reward Function

The reward function goal is to design a trading strategy that maximizes the change of the portfolio value:

$$r(s, a, s') = v' - v$$

where:
v' *and* v *represent the portfolio values at state* s' *and* s
a *is the action taken at state* s

4.2 Ensemble Strategy

The purpose of the ensemble strategy is to be a highly robust trading strategy. It will automatically select the best performing agent among DDPG, TD3, PPO, A2C, and SAC to trade based on the Sharpe ratio.

The core principle behind this setup is that each algorithm is sensitive to different type of trends. One agent may perform well in a bullish trend but badly in a bearish trend. Whereas another agent may be more suitable for volatile market conditions. The higher an algorithm Sharpe ratio, the better its returns have been relative to the amount of investment risk it has taken (better performance). Therefore, the trading agent that can maximize the returns adjusted to the increased risk is selected.

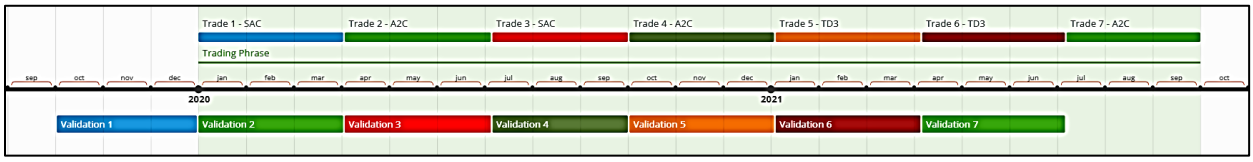

Fig 4: Ensemble Strategy Timeline

The ensemble process is described as follows:

Step 1: Use a growing window of n months to retrain the 5 agents concurrently. In this paper, the 5 agents are retrained every 3 months (63 days).

Step 2: Validate all 5 agents by using a 3-month validation (63 days) rolling window to pick the best performing agent with the highest Sharpe ratio.

Step 3: After the best agent is picked, use it to trade for the next 3 months (63 days).

TABLE III
ENSEMBLE STRATEGY SHARPE RATIO & MODEL SELECTION

	Iter	*Val Start*	*Val End*	*Model Picked*	*DDPG Sharpe*	*TD3 Sharpe*	*PPO Sharpe*	*A2C Sharpe*	*SAC Sharpe*
1	126	3/10/2019	3/01/2020	SAC	0.378823	0.528301	0.391671	0.332707	0.623039
2	189	3/01/2020	3/04/2020	A2C	-0.41911	-0.41861	-0.32846	-0.31634	-0.37338
3	252	3/04/2020	6/07/2020	SAC	0.429154	0.476688	0.440794	0.344554	0.479630
4	315	6/07/2020	2/10/2020	A2C	0.115733	0.113784	-0.01673	0.203629	0.101520
5	378	2/10/2020	4/01/2021	TD3	0.328222	0.371970	0.209871	0.222040	0.349700
6	441	4/01/2021	6/04/2021	TD3	0.177734	0.205645	0.017392	0.121242	0.040542
7	504	6/04/2021	6/07/2021	A2C	0.199771	0.342633	0.126254	0.417919	0.201515

4.3 Performance Comparisons

TABLE IV
PERFORMANCE EVALUATION COMPARISON
(NORMAL ENVIRONMENT)

	DDPG	TD3	PPO	A2C	SAC	Ensemble	Min-Var	NDX
Cumulative Returns	38.69%	39.91%	31.42%	**87.29%**	27.87%	52.02%	26.29%	66.28%
Annual Returns	20.55%	21.15%	16.90%	**43.13%**	15.08%	27.04%	14.27%	33.81%
Annual Volatility	26.71%	28.29%	**25.52%**	31.61%	25.62%	31.70%	22.39%	30.11%
Sharpe Ratio	0.84	0.82	0.74	**1.30**	0.68	0.91	0.71	1.12
Sortino Ratio	1.14	1.12	1.04	**1.91**	0.94	1.40	1.00	1.57
Calmar Ratio	0.64	0.80	0.65	**1.63**	0.53	0.85	0.54	1.21
Max Drawdown	-31.87%	-26.53%	**-26.03%**	-26.43%	-28.52%	-31.86%	-26.56%	-28.03%

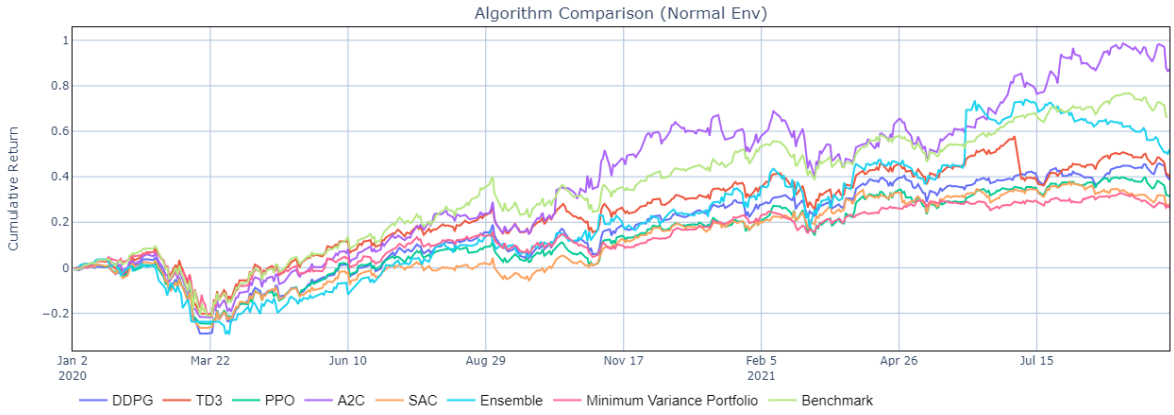

Fig 5: Cumulative Returns – Normal Environment (Initial Capital of $1,000,000)

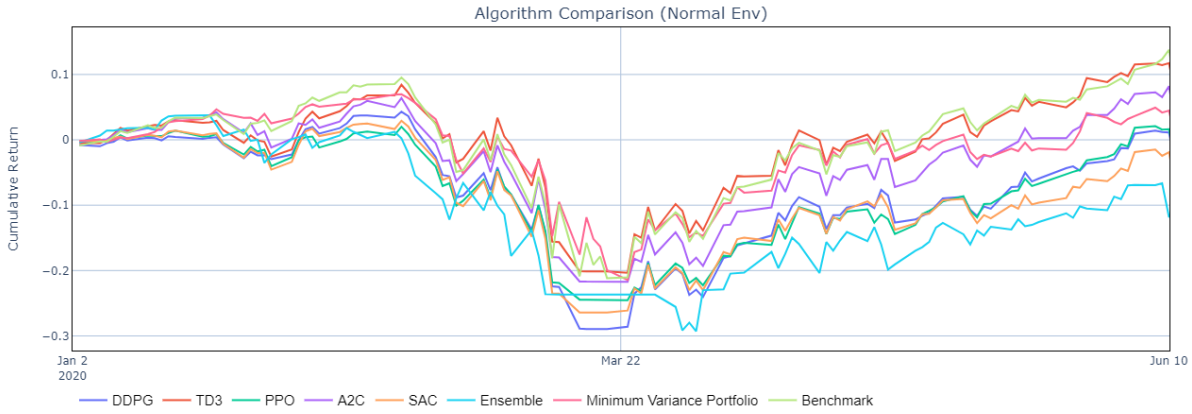

Fig 6: Cumulative Returns – Normal Environment – Stock Market Crash (Initial Capital of $1,000,000)

Analysis of Algorithm Performance

It can observe that the PPO is more adaptive to risk. It has the lowest annual volatility of 25.52% and a max drawdown of −26.03%. In other words, it is good at handling a bearish market. A2C is good at following trends and acts well in generating high returns, it has the highest annual return of 41.13% and the highest cumulative return of 87.29%. Hence, A2C is preferred when facing a bullish market.

TD3 also performs relatively well but was not as good as A2C, it can be used as a complementary strategy to A2C in a bullish market. During the final quarter of the trading phase, it made a critical and major prediction error which caused the cumulative returns to drop sharply. This is a stark reminder that an agent is not full proof and there will be needs for risk control and limits in place.

Using annual return and cumulative return as performance metrics, A2C was the only agent whose performance outperform both the NDX and min-variance portfolio allocation. All other agents manage to beat the min-variance portfolio allocation of NDX.

Performance under Market Crash

By incorporating the VIX index, the agents were able to cut losses early and successfully survive the stock market crash in March 2020. However, the timing to cut loss came rather late. The VIX index threshold could be tuned lower for higher risk aversion. Alternatively, the financial turbulence index could be employed instead. The turbulence index is an indicator when prices movements violate the existing correlation structure; the “decoupling of correlated assets” and the “convergence of uncorrelated assets” (Kritzman & Li, 2010).

Ensemble Comparison

The ensemble strategy performed reasonably well, achieving an annual return of 27.04% and a cumulative return of 52.02%. Its Sharpe ratio of 0.91 was much higher than the Sharpe ratio of 0.71 for the min-variance portfolio allocation but lower than the Sharpe ratio of 1.12 for NDX.

Based on the model picked by the agent during the trading phase, both DDPG and PPO were not chosen and hence indicate that they are inferior choices based on the Sharpe ratio. This could imply that on a quarterly basis, both models are not good at balancing risk and return. However, when looking at a long-term perspective both models beat SAC Sharpe ratio of 0.68.

These findings demonstrate that the proposed ensemble strategy can effectively be used to develop a trading strategy that performs well. However, the validation interval could be tuned lower to allow for more flexibility.

5 Cash Penalty Environment

5.1 Reward Function

On top of the constraints of the normal environment, the cash penalty environment includes a penalty for not maintaining a reserve of cash. This constraint makes it more realistic as funds are commonly mandated to carry cash reserves of about 3% to 5%. In some cases, certain jurisdictions have laws in place to ensure that not all capital is exposed to market risk. This additional constraint enables the agent to manage cash reserves in addition to performing trading procedures. To prevent the agent from simply hoarding cash, it is penalized for each lapsed day of not trading as the reward gets smaller.

The reward function is defined as:

$$r(s,a,s') = \frac{(cash + asset) - initial\ cash - \max\left[0, \left((cash + asset) * cash\ penalty\ proportion - cash\right)\right]}{days\ elapsed}$$

In this paper, the cash penalty proportion is set at 0.05 (i.e., if cash reserve is less than 5% of total asset value, the cash penalty will be applied.

5.2 Performance Comparison

TABLE V
PERFORMANCE EVALUATION COMPARISON
(CASH PENALTY ENVIRONMENT)

	DDPG	TD3	PPO	A2C	SAC	Min-Var	NDX
Cumulative Returns	26.84%	**32.51%**	3.11%	2.75%	34.41%	26.29%	66.28%
Annual Returns	14.59%	**17.49%**	1.77%	1.56%	18.45%	14.27%	33.81%
Annual Volatility	12.84%	14.56%	**1.64%**	1.65%	13.64%	22.39%	30.11%
Sharpe Ratio	1.13	**1.18**	1.08	0.95	1.31	0.71	1.12
Sortino Ratio	1.62	**1.71**	1.52	1.34	1.89	1.00	1.57
Calmar Ratio	2.05	**2.01**	1.82	1.56	2.54	0.54	1.21
Max Drawdown	-7.01%	-7.79%	**-0.97%**	-1.00%	-7.28%	-26.56%	-28.03%

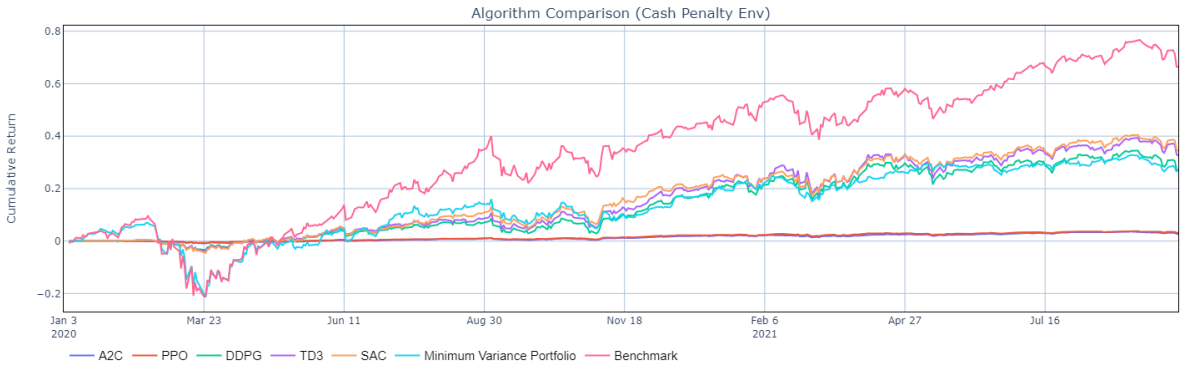

Fig 7: Cumulative Returns – Cash Penalty Environment (Initial Capital of $1,000,000)

Analysis of Algorithm Performance

Using annual return and cumulative return as performance metrics, none of the agents managed to beat the NDX. However, DDPG, TD3 and SAC managed to outperform the min-variance portfolio allocation.

TABLE VI
REMAINING CASH – A2C & PPO
(CASH PENALTY ENVIRONMENT)

PPO					
EPISODE	STEPS	TERMINAL_REASON	CASH	TOT_ASETS	CASH_PROPORTION
1	440	Last Date	$903,462	$1,031,150	87.62%

A2C					
EPISODE	STEPS	TERMINAL_REASON	CASH	TOT_ASETS	CASH_PROPORTION
1	440	Last Date	$887,136	$1,027,519	86.34%

Specifically, special attention needs to be brought to PPO and A2C which had a miserable annual return of 1.77% and 1.56% respectively, and a cumulative return of 3.11% and 2.75% respectively. This could be explained by the agent highly risk-averse nature due to the penalty imposed in the cash penalty environment which led to only a small portion of available cash being invested during the trading period.

DDPG, TD3 and SAC had Sharpe ratio, Sortino ratio and Calmar ratio that were better than their normal environment counterparts. This can be explained by the cash reserve constraint which meant that lesser risk can be taken and instead hurt both their annual return and cumulative return.

This was however not true for TD3 which had an annual return of 17.49% and a cumulative return of 32.51%, about 20 basis points lower than its normal environment counterpart. The agent surprisingly performed better when additional constraints are placed on it.

These findings show that a simple small change in the reward function that penalizes the agent for not carrying sufficient reserve cash can drastically change the behaviour of the agent to become more risk-averse, and in one instance perform better.

6 Stop Loss Environment

6.1 Reward Function

Building on the cash penalty environment, the stop loss environment penalizes the agent if it exceeds the stop-loss threshold, selling assets below the expected profit margin, and for not maintaining a reserve of cash. This creates a realistic trading environment where funds are mandated to cut losses once a certain threshold of loss is reached. The reward function accounts for a profit/loss ratio constraint, liquidity requirement, and long-term accrued rewards. It also forces the agent to trade only when it's confident to do so.

The reward function is defined as:

$$r(s,a,s') = [(cash + asset) + additional\ reward - total\ penalty - initial\ cash)]/initial\ cash/days\ elapsed$$

where:

$$total\ penalty = cash\ penalty + stop\ loss\ penalty + low\ profit\ penalty$$
$$cash\ penalty = \max[0, (cash + asset) * cash\ penalty\ proportion - cash]$$
$$stop\ loss\ penalty = -1 * (holdings * negative\ closing\ diff\ avg\ buy)$$
$$low\ profit\ penalty = -1 * (holdings * negative\ profit\ sell\ diff\ avg\ buy)$$
$$additional\ reward = holdings * positive\ profit\ sell\ diff\ avg\ buy$$

In this paper, the cash penalty proportion is set at 0.5 same as before, stop loss at 0.75 (i.e., the agent will cut loss if a particular asset drops more than 25% below the average purchase price) and an expected profit loss ratio of 1.5 (i.e., the high profit and lower profit remarks is determined by this ratio).

6.2 Performance Comparison

TABLE VII
PERFORMANCE EVALUATION COMPARISON
(STOP LOSS ENVIRONMENT)

	DDPG	TD3	PPO	A2C	SAC	Min-Var	NDX
Cumulative Returns	22.87%	**25.74%**	2.06%	2.03%	4.12%	26.29%	66.28%
Annual Returns	12.52%	**14.02%**	1.18%	1.16%	2.34%	14.27	33.81%
Annual Volatility	12.68%	12.14%	**1.31%**	1.47%	3.13%	22.39%	30.11%
Sharpe Ratio	1.00	**1.14**	0.90	0.79	0.76	0.71	1.12
Sortino Ratio	1.40	**1.63**	1.27	1.10	1.05	1.00	1.57
Calmar Ratio	1.77	**1.90**	1.64	1.43	1.28	0.54	1.21
Max Drawdown	-7.07%	-7.40%	**-0.72%**	-0.81%	-1.82%	-26.56%	-28.03%

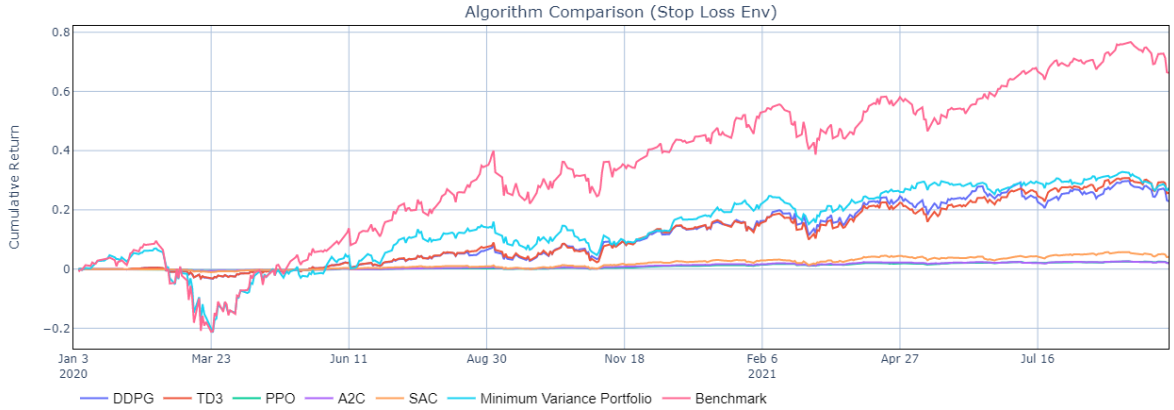

Fig 8: Cumulative Returns – Stop Loss Environment (Initial Capital of $1,000,000)

Analysis of Algorithm Performance

Using annual return and cumulative return as performance metrics, none of the agents managed to beat NDX and min-variance portfolio allocation. This was expected given that the stop loss environment has much more constraints than the cash penalty environment.

TABLE VIII
REMAINING CASH – A2C, PPO & SAC
(STOP LOSS ENVIRONMENT)

A2C					
EPISODE	STEPS	TERMINAL_REASON	CASH	TOT_ASETS	CASH_PROPORTION
1	440	Last Date	$903,129	$1,020,300	88.52%

PPO					
EPISODE	STEPS	TERMINAL_REASON	CASH	TOT_ASETS	CASH_PROPORTION
1	440	Last Date	$914,556	$1,020,611	89.61%

SAC					
EPISODE	STEPS	TERMINAL_REASON	CASH	TOT_ASETS	CASH_PROPORTION
1	440	Last Date	$657,588	$1,041,227	63.16%

In the stop loss environment, in addition to PPO and A2C, SAC suffered the same fate of being risk-averse and invested only a small portion of available cash during the trading phase. The level of risk aversion increased as interpreted from the higher cash proportion as compared to their cash penalty environment counterparts.

TABLE IX
PERFORMANCE REMARKS – DDPG & TD3
(STOP LOSS ENVIRONMENT)

	DDPG	TD3
STOP LOSS	7	7
LOW PROFIT	90	0
HIGH PROFIT	25	0
HIGH PROFIT/ LOW PROFIT RATIO	0.27	NA

It would be expected that if the agent trades with higher confidence, the downside risk measured by the Sortino ratio and Calmar ratio would be higher. However, this was not the case, both ratios dropped as compared to their cash penalty environment counterparts across all the agents. Upon closer inspection of DDPG and TD3 (which utilized the available cash to the fullest), it was discovered that the ratio of “high profit to low profit” was very low for DDPG at 0.27, and not applicable to TD3 (no special performance remarks at all). This indicates that the trades were not executed with high confidence.

One possible explanation for these observations was that the agents were not trained enough to build up enough competency and trade confidently. If enough training was given, the level of risk aversion would be lowered significantly along with the available cash surplus, while the Sortino ratio and Calmar ratio would be significantly higher.

7 Conclusion

7.1 Key Findings

In the normal trading environment that had no constraint, A2C significantly outperformed all other algorithms and benchmarks for nearly all performance metrics. Despite the sound core principles in the ensemble strategy design, it did not perform as well as expected. It was however, still an effective strategy overall. Interestingly in the other more constrained environments of cash penalty and stop loss, TD3 consistently emerged as the best performing agent in cumulative returns and risk-adjusted returns. It does suggest that in a complex constrained trading environment, TD3 could be the best algorithm to consider deploying.

The results are conclusive that no single algorithm can perform equally well in different trading environments, and it was not a “one size fits all” situation. This was evident in the performance comparison for each trading environment and the fact that the ensemble strategy was as effective as expected.

Like OpenAI Five deciding to use PPO as its algorithm, many factors need to be considered before making a final decision, for example, resource availability (budget, time and computing power), the size and complexity of the environment design (actions space and state space) and the size of data (daily vs hourly data and SPY vs NDX vs DJI).

7.2 Future Research

To value add to the current findings, further modifications can be made to the existing ensemble strategy to focus on A2C and TD3 since they are the best in their respective environments. By reducing the ensemble strategy selection choices from 5 to 2 algorithms, it would also significantly reduce the time needed to train the model. For non-institutional machine learning enthusiasts who face limited computing resources, this should be taken into consideration. In addition, modifying the ensemble strategy to choose the best performing agent based on the highest Sortino ratio instead of the Sharpe ratio may reduce the downside risk to the portfolio.

Further research can be done to optimise the parameters in the cash penalty and stop loss environment to prevent the agents from being too risk-averse. Currently, some of the algorithms are holding too much available cash which reduces the potential upside of the portfolio.

Other than stock trading, this research can be replicated on portfolio optimization. It would be insightful to analyse the difference in effectiveness in the application of deep reinforcement learning in stock trading and portfolio optimization. The ensemble strategy can also be deployed in portfolio optimization.

References

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J. W., Petrov, M., de Oliveira Pinto, H. P., Raiman, J., Salimans, T., Schlatter, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wolski, F. & Zhang, S. (2019). Dota 2 with Large Scale Deep Reinforcement Learning. ArXiv, abs/1912.06680.

Bollinger, J. (2002). Bollinger on Bollinger Bands. McGraw-Hill.

Chang, Y.-H., & Lee, M.-S. (2017). Incorporating Markov decision process on genetic algorithms to formulate trading strategies for stock markets. Applied Soft Computing, 52, 1143–1153. https://doi.org/10.1016/j.asoc.2016.09.016

Chen, S.-A., Tangkaratt, V., Lin, H.-T., & Sugiyama, M. (2019). Active deep Q-learning with demonstration. Machine Learning, 109(9-10), 1699–1725. https://doi.org/10.1007/s10994-019-05849-4

Chong, T., Ng, W.-K., & Liew, V. (2014). Revisiting the performance of MACD and RSI Oscillators. Journal of Risk and Financial Management, 7(1), 1–12. https://doi.org/10.3390/jrfm7010001

Gurrib, I. (2018). Performance of the average directional index as a market-timing tool for the most actively traded USD based Currency Pairs. Banks and Bank Systems, 13(3), 58–70. https://doi.org/10.21511/bbs.13(3).2018.06

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. ICML.

Hasselt, H.V. (2010). Double Q-learning. Advances in Neural Information Processing Systems 23 (NIPS 2010), Vancouver, British Columbia, Canada, pp. 2613–2622.

Hasselt, H.V., Guez, A., & Silver, D. (2016). Deep Reinforcement Learning with Double Q-Learning. ArXiv, abs/1509.06461.

Jeon, G., & Kim, H. Y. (2019). Improving financial trading decisions using Deep Q-Learning: Predicting the number of shares, Action Strategies, and transfer learning. Expert Systems with Applications, 117, 125–138. https://doi.org/10.1016/j.eswa.2018.09.036

Kritzman, M., & Li, Y. (2010). Skulls, financial turbulence, and risk management. Financial Analysts Journal, 66(5), 30–41. https://doi.org/10.2469/faj.v66.n5.3

Li, Y. (2017). Deep Reinforcement Learning: An Overview. ArXiv, abs/1701.07274.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N.M., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. CoRR, abs/1509.02971.

Maitah, M., Procházka, P., Cermak, M., & Šrédl, K. (2016). Commodity channel index: Evaluation of trading rule of agricultural commodities. International Journal of Economics and Financial Issues, 6(1), 176-178.

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T.P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. ICML.

Neuneier, R. (1996). Optimal asset allocation using adaptive dynamic programming. Advances in Neural Information Processing Systems, 952-958.

Schulman, J., Levine, S., Abbeel, P., Jordan, M.I., & Moritz, P. (2015). Trust Region Policy Optimization. ArXiv, abs/1502.05477

Sigaud, O., & Buffet, O. (2010). Markov decision processes in Artificial Intelligence: Mdps, beyond mdps and applications. ISTE.

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems (pp. 1057-1063).

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. Expert Systems with Applications, 173, 114632. https://doi.org/10.1016/j.eswa.2021.114632

Wang, Z., Schaul, T., Hessel, M., Hasselt, H.V., Lanctot, M., & Freitas, N.D. (2016). Dueling Network Architectures for Deep Reinforcement Learning. ArXiv, abs/1511.06581.

Watkins, C. J., & Dayan, P. (1992). Q-learning. Machine Learning, 8(3-4), 279–292. https://doi.org/10.1007/bf00992698

White, C. C., & White, D. J. (1989). Markov decision processes. European Journal of Operational Research, 39(1), 1–16. https://doi.org/10.1016/0377-2217(89)90348-2

Wilcox, P. (2019, January 5). Why deep reinforcement learning (DRL) matters for trading. Lucena Research. Retrieved December 4, 2021, from https://lucenaresearch.com/2019/01/05/deep-reinforcement-learning-for-investment-professionals/.

Wu, Y., Mansimov, E., Liao, S., Radford, A., & Schulman, J. (2020, June 29). OpenAI Baselines: ACKTR & A2C. OpenAI. Retrieved December 5, 2021, from https://openai.com/blog/baselines-acktr-a2c/.

Yang, H., Liu, X.-Y., Zhong, S., & Walid, A. (2020). Deep Reinforcement Learning for Automated Stock Trading: An ensemble strategy. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3690996

Appendix

Appendix A (Normal Environment)

1. DDPG

Backtest	
Annual return	20.55%
Cumulative returns	38.69%
Annual volatility	26.71%
Sharpe ratio	0.84
Calmar ratio	0.64
Stability	0.84
Max drawdown	-31.87%
Omega ratio	1.17
Sortino ratio	1.14
Skew	NaN
Kurtosis	NaN
Tail ratio	0.85
Daily value at risk	-3.28%
Alpha	-0.04
Beta	0.7

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	31.87	19/02/2020	17/03/2020	6/07/2020	99
1	12.32	2/09/2020	21/09/2020	25/11/2020	61
2	8.25	29/04/2021	12/05/2021	6/07/2021	49
3	7.58	19/02/2021	4/03/2021	1/04/2021	30
4	5.74	21/01/2021	29/01/2021	11/02/2021	16

Table A1 DDPG Backtest Stats

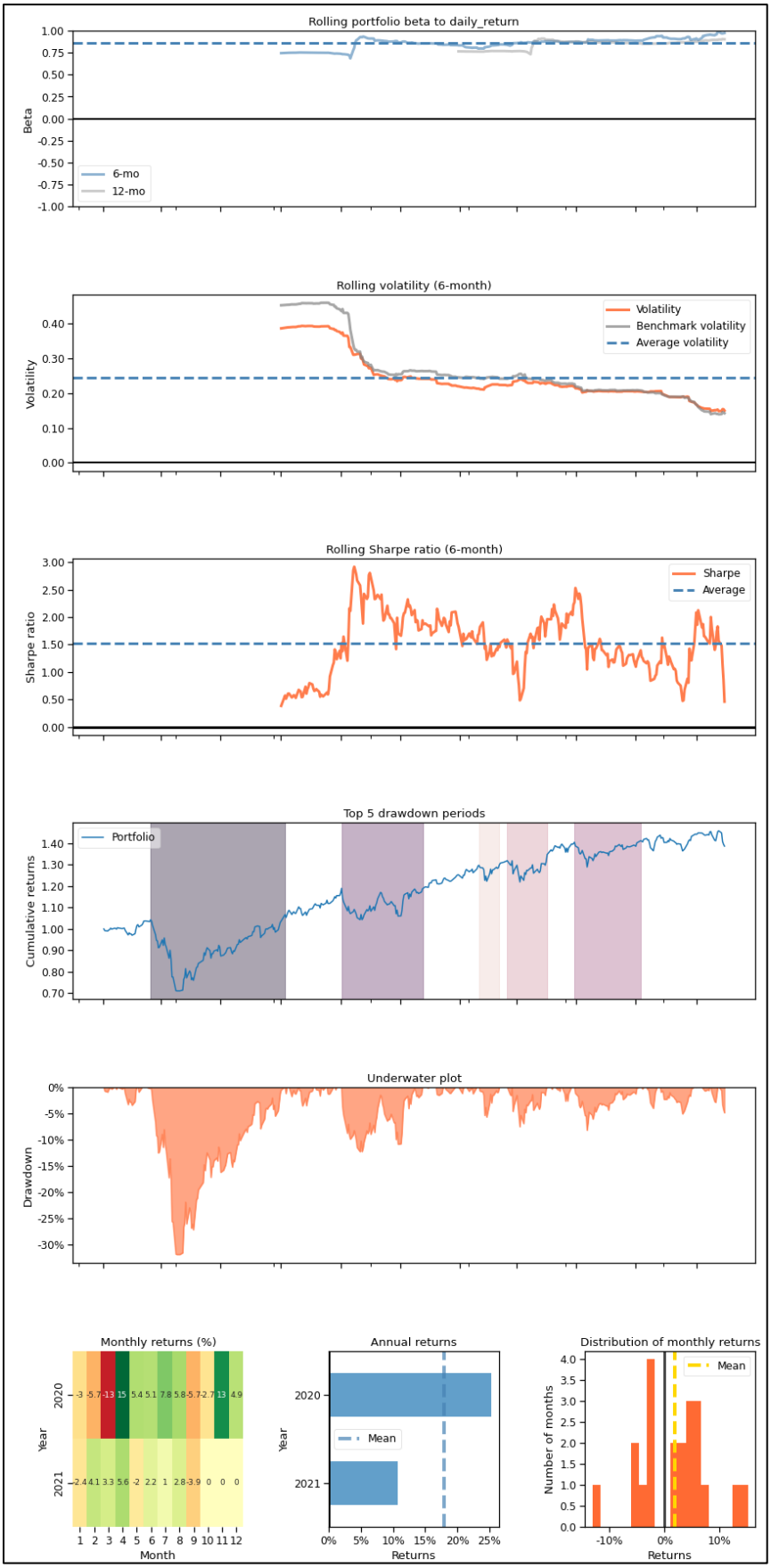

Fig A1 DDPG Backtest Plots

2. TD3

Backtest	
Annual return	21.15%
Cumulative returns	39.91%
Annual volatility	28.29%
Sharpe ratio	0.82
Calmar ratio	0.8
Stability	0.86
Max drawdown	-26.53%
Omega ratio	1.17
Sortino ratio	1.12
Skew	NaN
Kurtosis	NaN
Tail ratio	0.93
Daily value at risk	-3.47%
Alpha	-0.03
Beta	0.78

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	26.53	19/02/2020	23/03/2020	29/05/2020	73
1	13.73	2/07/2021	19/07/2021	NaT	NaN
2	11.36	16/02/2021	8/03/2021	5/04/2021	35
3	10	2/09/2020	8/09/2020	13/10/2020	30
4	9.72	13/10/2020	30/10/2020	1/12/2020	36

Table A2 TD3 Backtest Stats

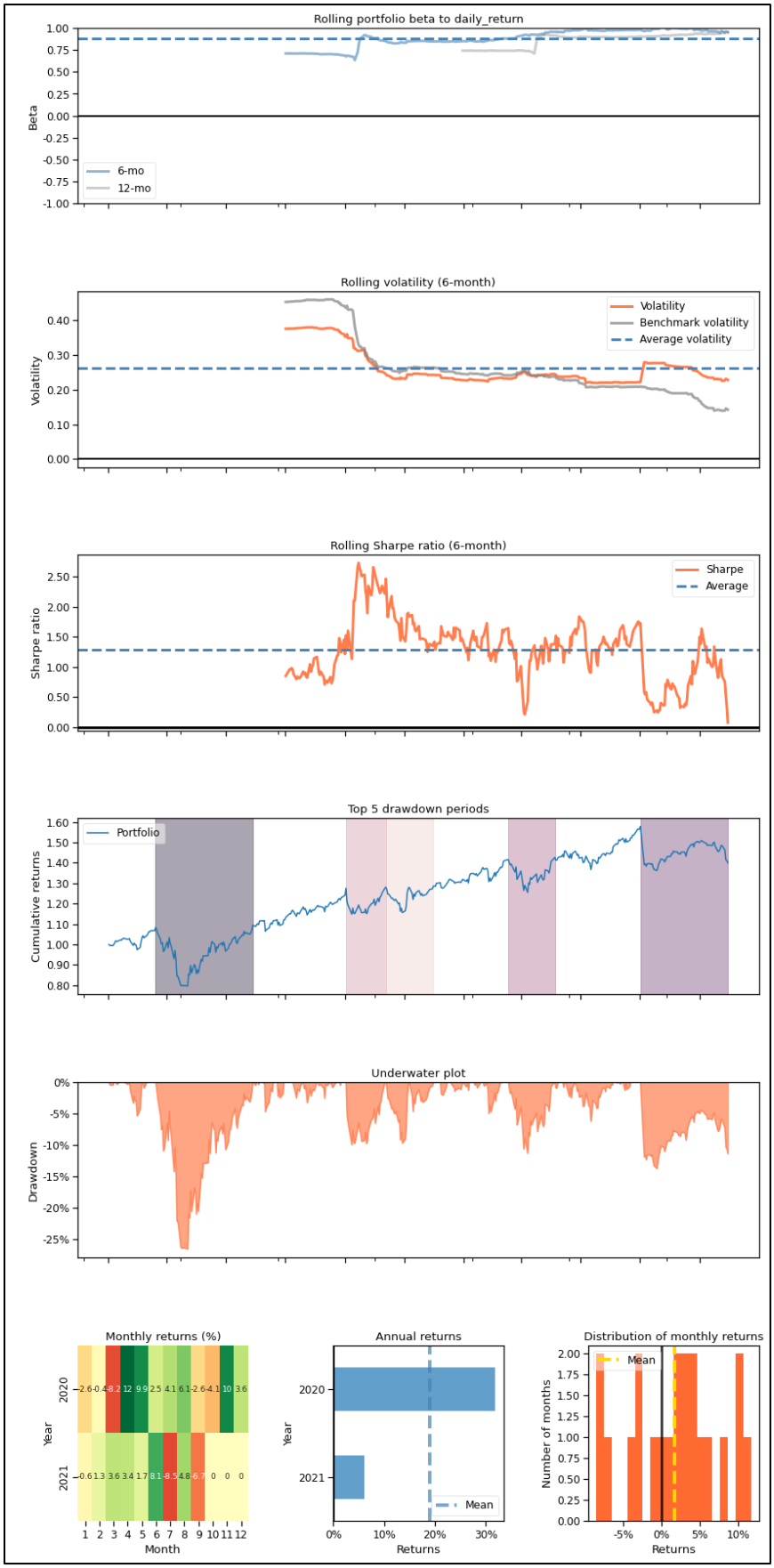

Fig A2 TD3 Backtest Plots

3. PPO

Backtest	
Annual return	16.90%
Cumulative returns	31.42%
Annual volatility	25.52%
Sharpe ratio	0.74
Calmar ratio	0.65
Stability	0.85
Max drawdown	-26.03%
Omega ratio	1.14
Sortino ratio	1.04
Skew	NaN
Kurtosis	NaN
Tail ratio	0.92
Daily value at risk	-3.14%
Alpha	-0.05
Beta	0.71

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	26.03	2020-02-19	2020-03-23	2020-06-08	79
1	10.22	2021-02-12	2021-03-08	2021-04-01	35
2	9.69	2020-09-02	2020-10-30	2020-11-06	48
3	8.06	2021-04-27	2021-05-12	2021-06-14	35
4	5.99	2021-08-30	2021-09-30	NaT	NaN

Table A3 PPO Backtest Stats

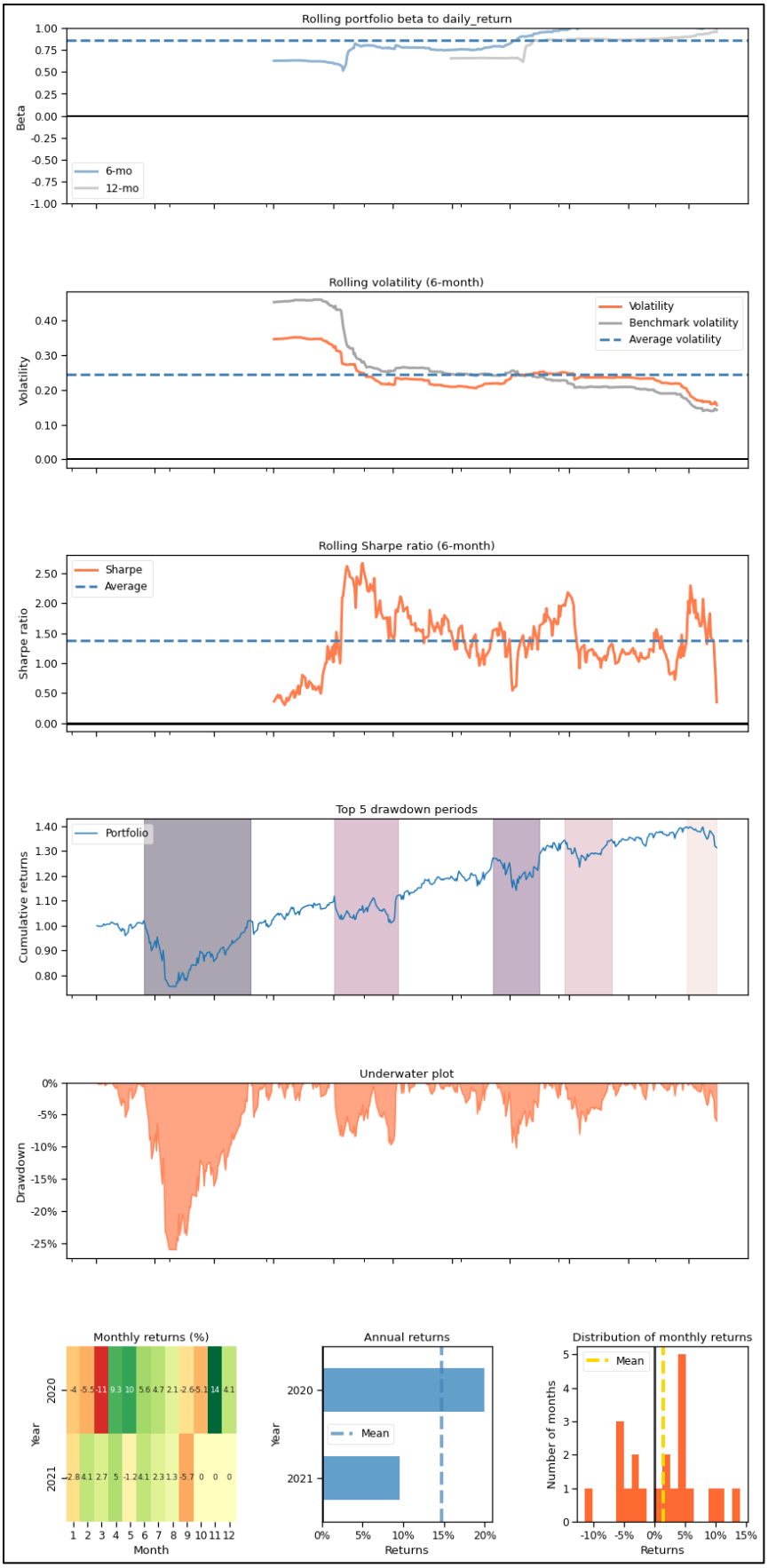

Fig A3 PPO Backtest Plots

4. A2C

Backtest	
Annual return	43.13%
Cumulative returns	87.29%
Annual volatility	31.61%
Sharpe ratio	1.3
Calmar ratio	1.63
Stability	0.9
Max drawdown	-26.43%
Omega ratio	1.25
Sortino ratio	1.91
Skew	NaN
Kurtosis	NaN
Tail ratio	1
Daily value at risk	-3.82%
Alpha	0.14
Beta	0.84

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	26.43	19/02/2020	23/03/2020	5/06/2020	78
1	17.16	12/02/2021	8/03/2021	17/06/2021	90
2	11.07	2/09/2020	23/09/2020	9/10/2020	28
3	9.75	11/01/2021	29/01/2021	12/02/2021	25
4	7.51	6/11/2020	10/11/2020	27/11/2020	16

Table A4 A2C Backtest Stats

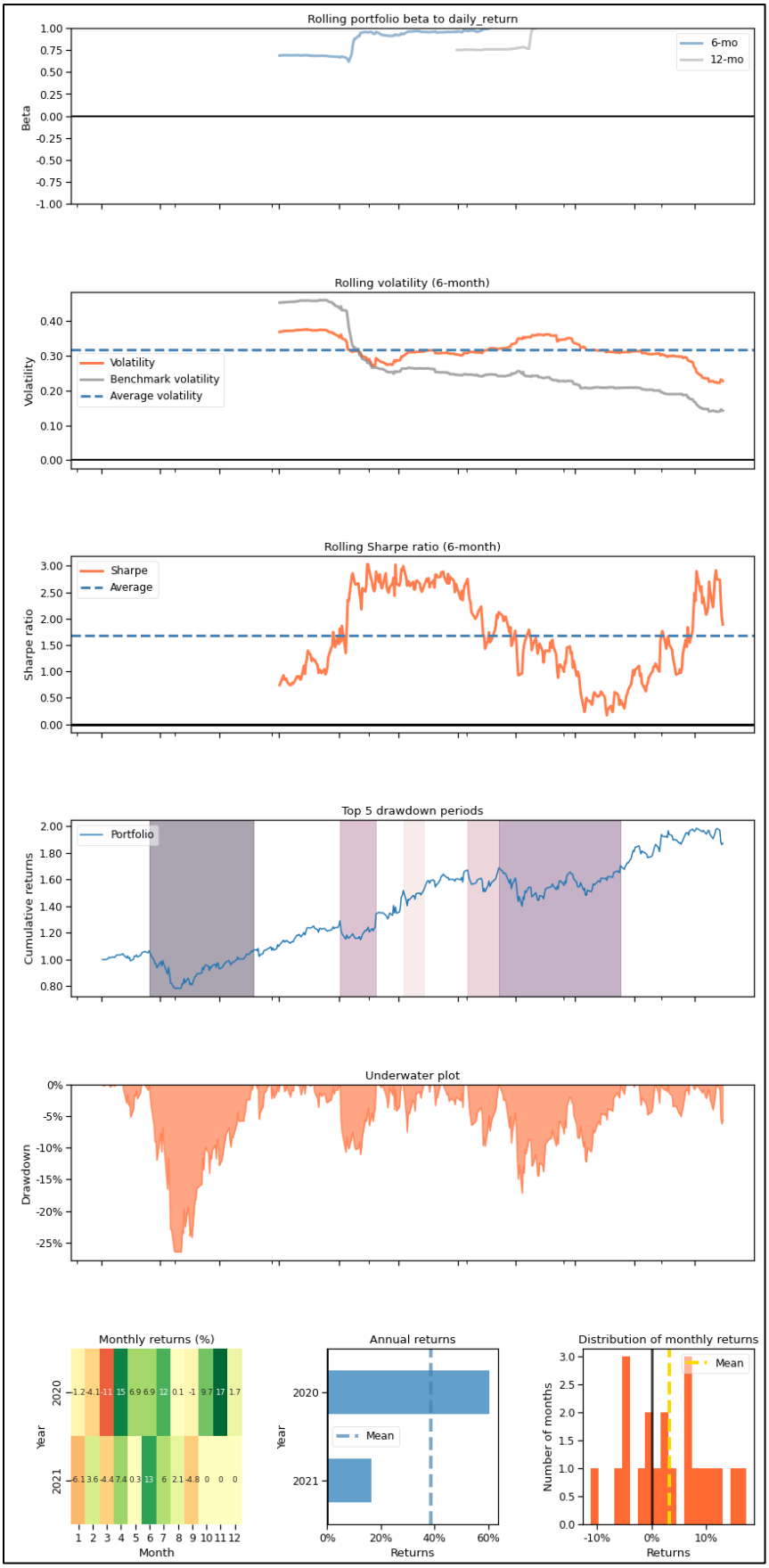

Fig A4 A2C Backtest Plots

5. SAC

Backtest	
Annual return	15.08%
Cumulative returns	27.87%
Annual volatility	25.62%
Sharpe ratio	0.68
Calmar ratio	0.53
Stability	0.83
Max drawdown	-28.52%
Omega ratio	1.13
Sortino ratio	0.94
Skew	NaN
Kurtosis	NaN
Tail ratio	0.94
Daily value at risk	-3.16%
Alpha	-0.06
Beta	0.7

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	28.52	19/02/2020	17/03/2020	2/09/2020	141
1	9.49	2/09/2020	23/09/2020	12/10/2020	29
2	7.59	3/08/2021	29/09/2021	NaT	NaN
3	7.48	5/04/2021	12/05/2021	10/06/2021	49
4	6.26	24/02/2021	4/03/2021	15/03/2021	14

Table A5 SAC Backtest Stats

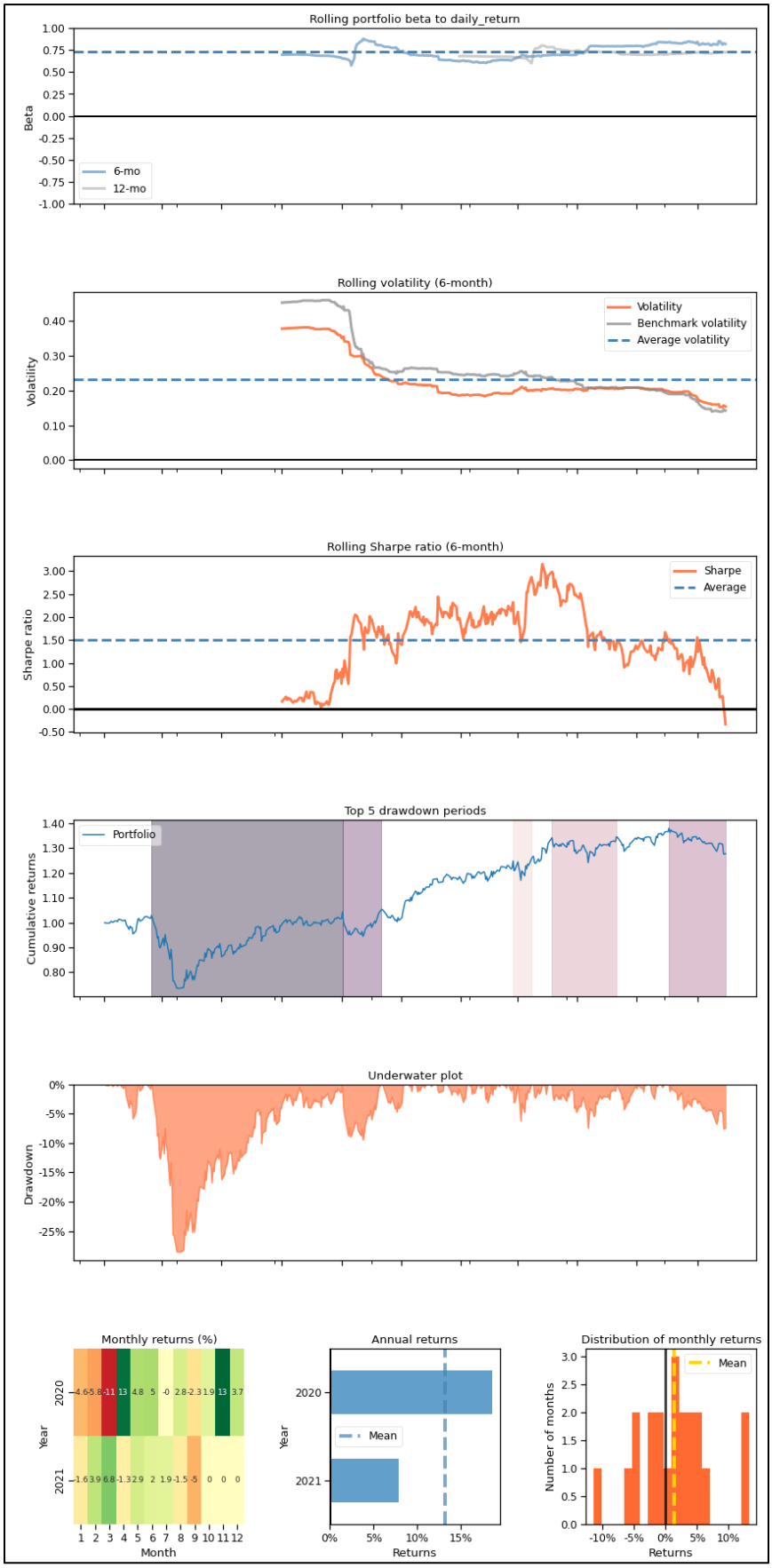

Fig A5 SAC Backtest Plots

6. Ensemble

Backtest	
Annual return	27.04%
Cumulative returns	52.02%
Annual volatility	31.70%
Sharpe ratio	0.91
Calmar ratio	0.85
Stability	0.87
Max drawdown	-31.86%
Omega ratio	1.19
Sortino ratio	1.40
Skew	NaN
Kurtosis	NaN
Tail ratio	0.99
Daily value at risk	-3.87%
Alpha	0.05
Beta	0.71

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	31.86	2020-01-22	2020-04-03	2020-07-20	129
1	18.06	2021-02-16	2021-03-08	2021-04-01	33
2	13.79	2021-07-12	2021-09-30	NaT	NaN
3	9.79	2020-09-02	2020-09-24	2020-10-27	40
4	7.37	2021-04-16	2021-05-12	2021-06-04	36

Table A6 Ensemble Backtest Stats

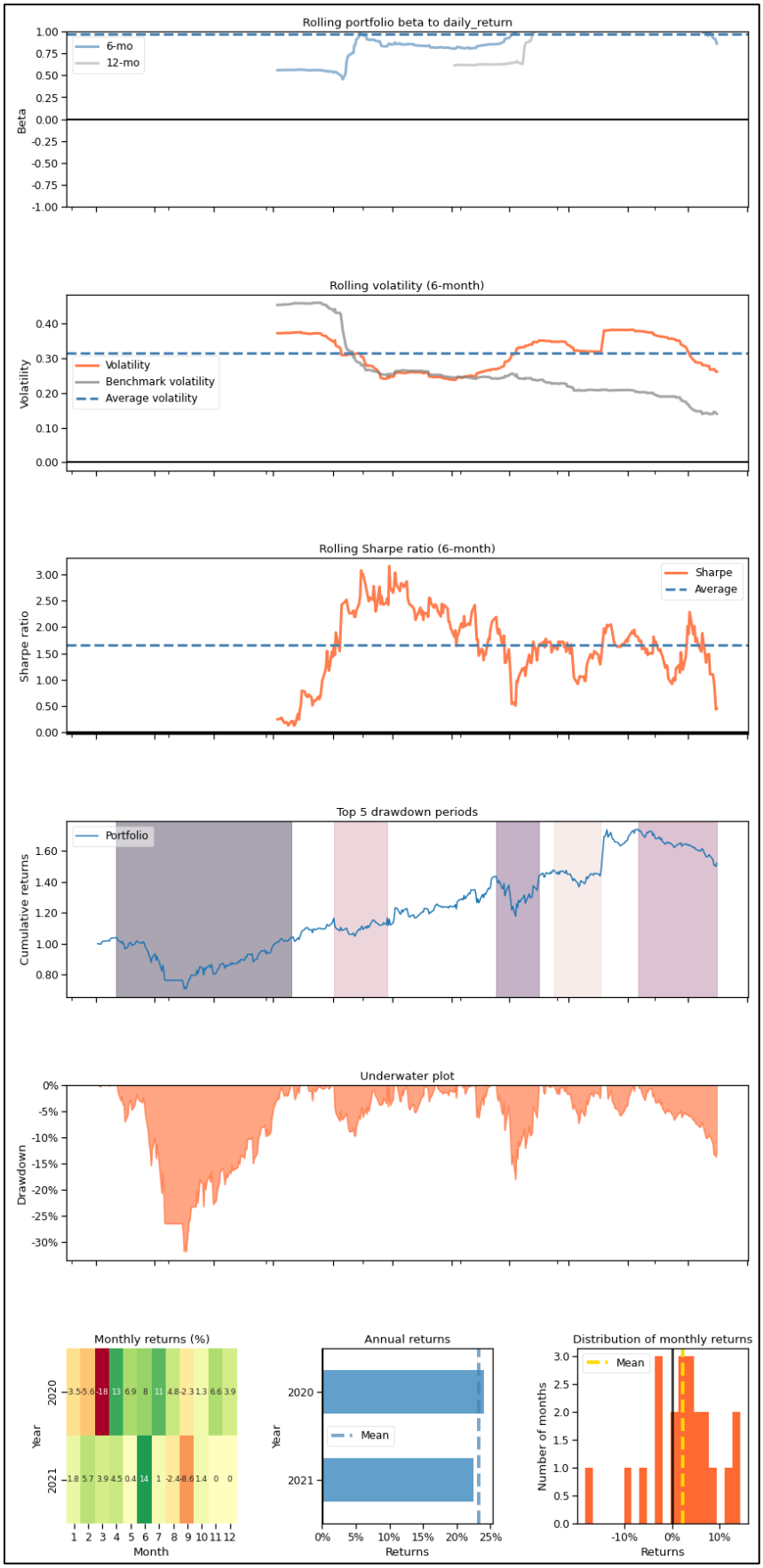

Fig A6 Ensemble Backtest Plots

7. Min Variance

Backtest	
Annual return	14.27%
Cumulative returns	26.29%
Annual volatility	22.39%
Sharpe ratio	0.71
Calmar ratio	0.54
Stability	0.82
Max drawdown	-26.56%
Omega ratio	1.15
Sortino ratio	1
Skew	NaN
Kurtosis	NaN
Tail ratio	0.93
Daily value at risk	-2.76%
Alpha	-0.06
Beta	0.66

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	26.56	19/02/2020	23/03/2020	6/07/2020	99
1	9.74	2/09/2020	30/10/2020	16/12/2020	76
2	7.65	9/02/2021	4/03/2021	15/04/2021	48
3	4.92	2/09/2021	30/09/2021	NaT	NaN
4	3.68	20/07/2020	28/07/2020	18/08/2020	22

Table A7 Min Variance Backtest Stats

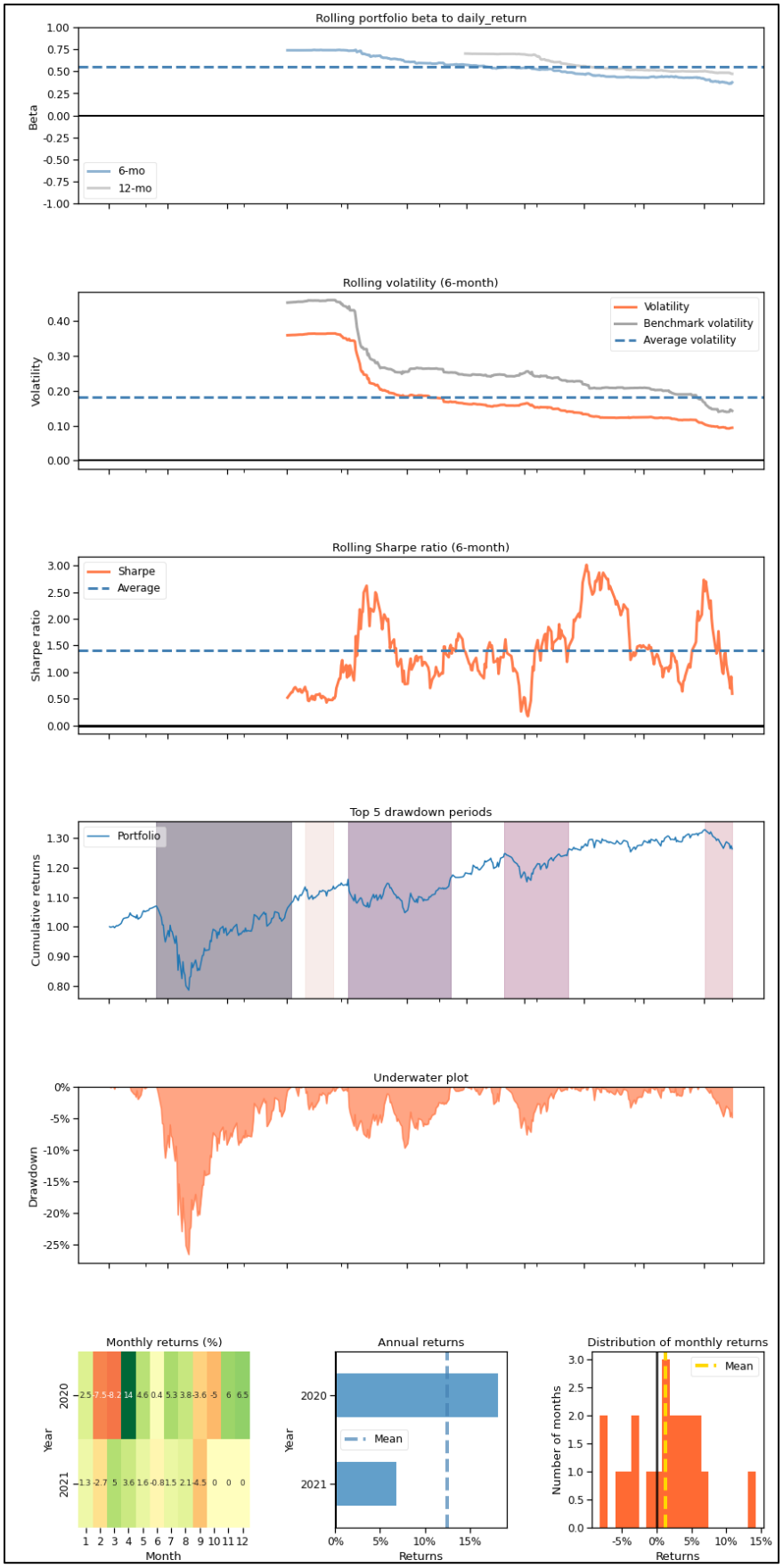

Fig A7 Min Variance Backtest Plots

Appendix B (Cash Penalty Environment)

1. DDPG

Backtest	
Annual return	14.59%
Cumulative returns	26.84%
Annual volatility	12.84%
Sharpe ratio	1.13
Calmar ratio	2.05
Stability	0.93
Max drawdown	-7.10%
Omega ratio	1.22
Sortino ratio	1.62
Skew	NaN
Kurtosis	NaN
Tail ratio	1.04
Daily value at risk	-1.56%
Alpha	0.18
Beta	-0.06

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	7.10	2021-02-16	2021-03-05	2021-04-01	33
1	6.39	2021-04-19	2021-05-13	2021-06-11	40
2	5.66	2021-09-07	2021-09-30	NaT	NaN
3	5.27	2020-10-13	2020-11-02	2020-11-09	20
4	5.17	2020-09-03	2020-09-24	2020-10-13	29

Table B1 DDPG Backtest Stats

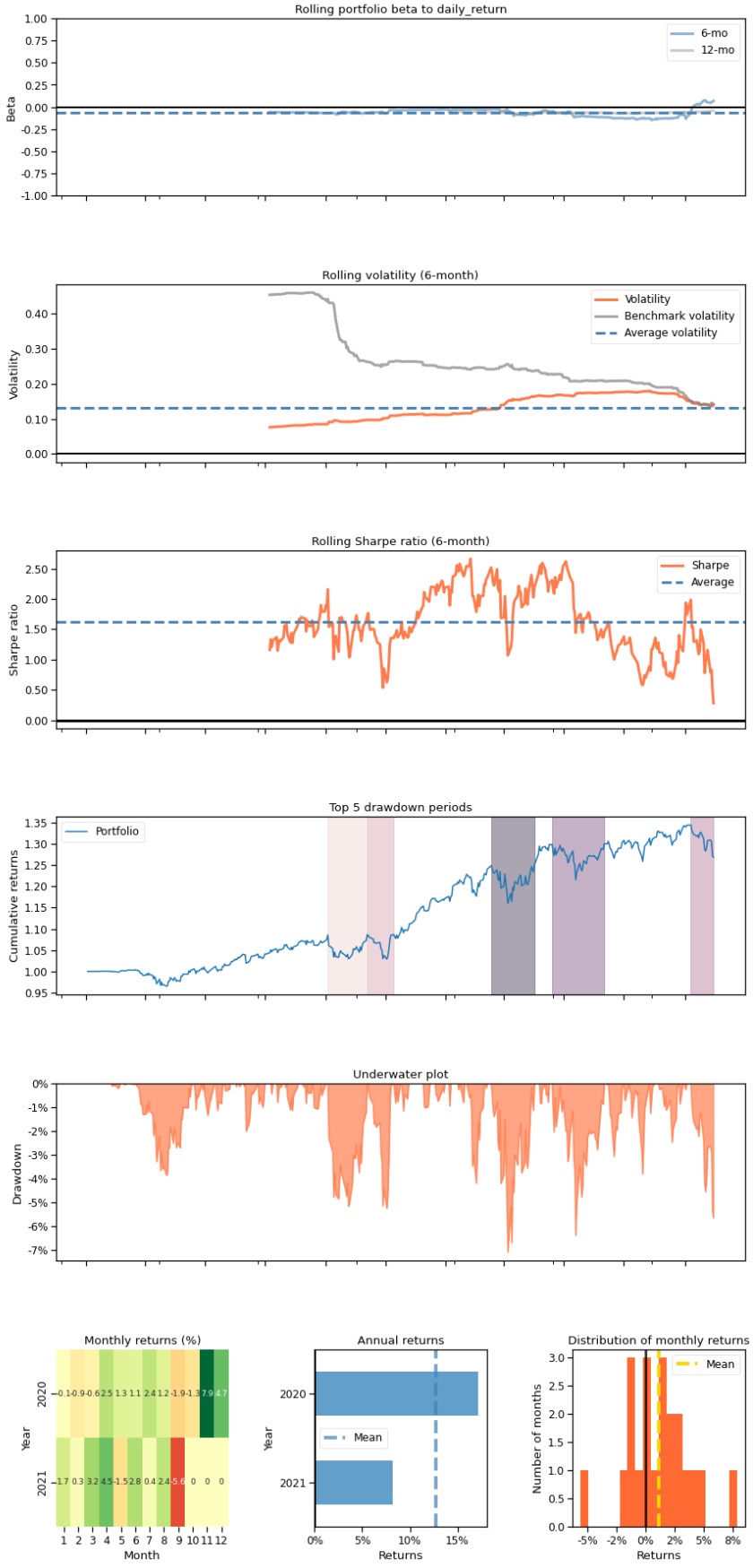

Fig B1 DDPG Backtest Plots

2. TD3

Backtest	
Annual return	17.49%
Cumulative returns	32.51%
Annual volatility	14.56%
Sharpe ratio	1.18
Calmar ratio	2.01
Stability	0.95
Max drawdown	-8.69%
Omega ratio	1.23
Sortino ratio	1.71
Skew	NaN
Kurtosis	NaN
Tail ratio	1.08
Daily value at risk	-1.76%
Alpha	0.22
Beta	-0.07

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	8.69	2021-02-16	2021-03-09	2021-04-05	35
1	6.86	2021-04-19	2021-05-13	2021-06-29	52
2	5.85	2020-09-03	2020-09-24	2020-10-13	29
3	5.56	2020-10-14	2020-11-02	2020-11-09	19
4	4.91	2021-09-07	2021-09-30	NaT	NaN

Table B2 TD3 Backtest Stats

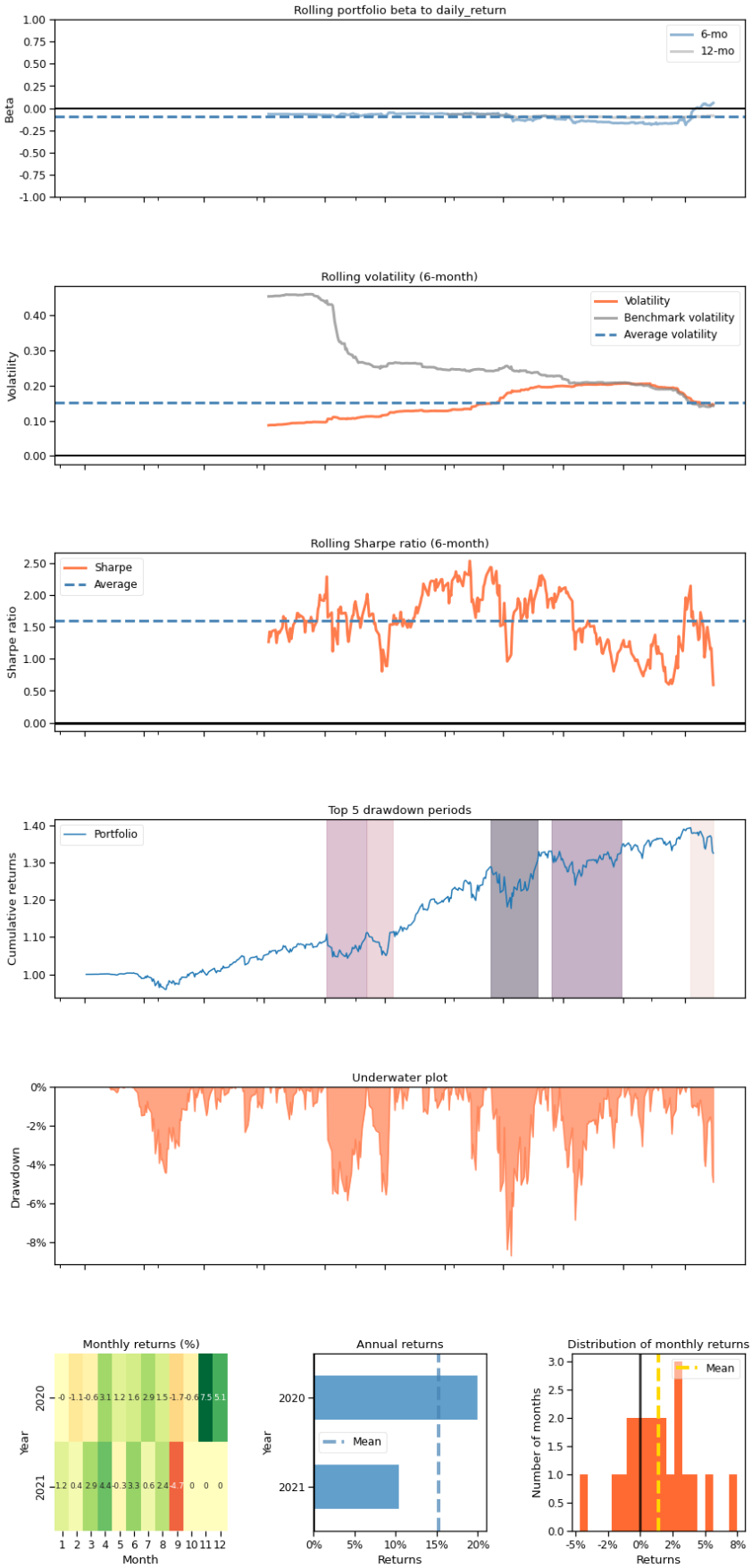

Fig B2 TD3 Backtest Plots

3. PPO

Backtest	
Annual return	1.77%
Cumulative returns	3.11%
Annual volatility	1.64%
Sharpe ratio	1.08
Calmar ratio	1.82
Stability	0.95
Max drawdown	-0.97%
Omega ratio	1.20
Sortino ratio	1.52
Skew	NaN
Kurtosis	NaN
Tail ratio	1.05
Daily value at risk	-0.2%
Alpha	0.02
Beta	-0.01

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	0.97	2021-02-16	2021-03-09	2021-04-05	35
1	0.82	2020-02-20	2020-03-23	2020-05-12	59
2	0.79	2021-04-27	2021-05-13	2021-06-11	34
3	0.72	2020-09-03	2020-09-24	2020-11-06	47
4	0.66	2021-08-31	2021-09-30	NaT	NaN

Table B3 PPO Backtest Stats

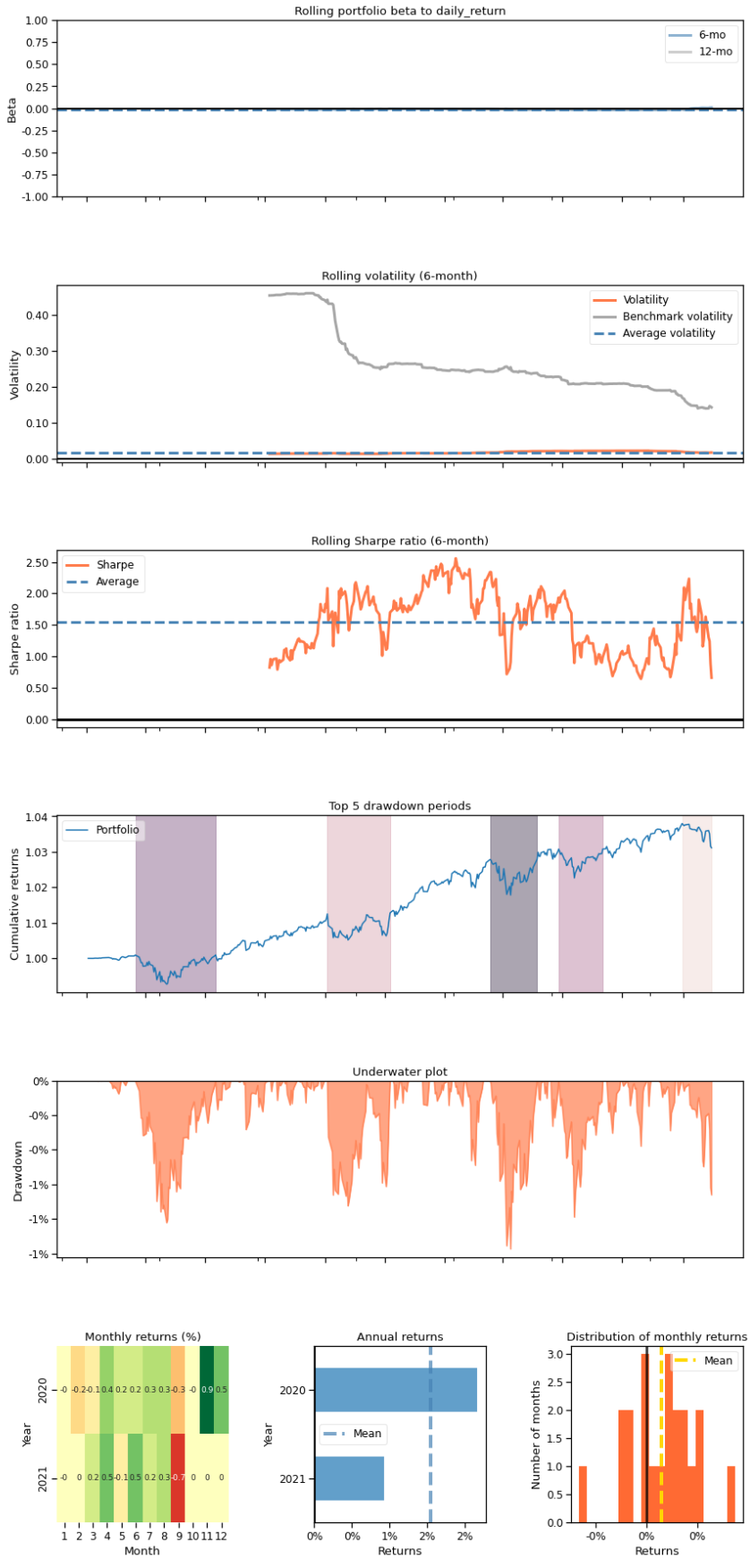

Fig B3 PPO Backtest Plots

4. A2C

Backtest	
Annual return	1.567%
Cumulative returns	2.752%
Annual volatility	1.654%
Sharpe ratio	0.95
Calmar ratio	1.56
Stability	0.93
Max drawdown	-1.008%
Omega ratio	1.17
Sortino ratio	1.34
Skew	NaN
Kurtosis	NaN
Tail ratio	1.11
Daily value at risk	-0.202%
Alpha	0.02
Beta	-0.01

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	1.01	2020-02-20	2020-03-24	2020-06-01	73
1	0.97	2021-02-16	2021-03-05	2021-04-06	36
2	0.83	2021-09-03	2021-09-30	NaT	NaN
3	0.76	2020-09-03	2020-09-24	2020-11-16	53
4	0.71	2021-04-27	2021-05-13	2021-06-11	34

Table B4 A2C Backtest Stats

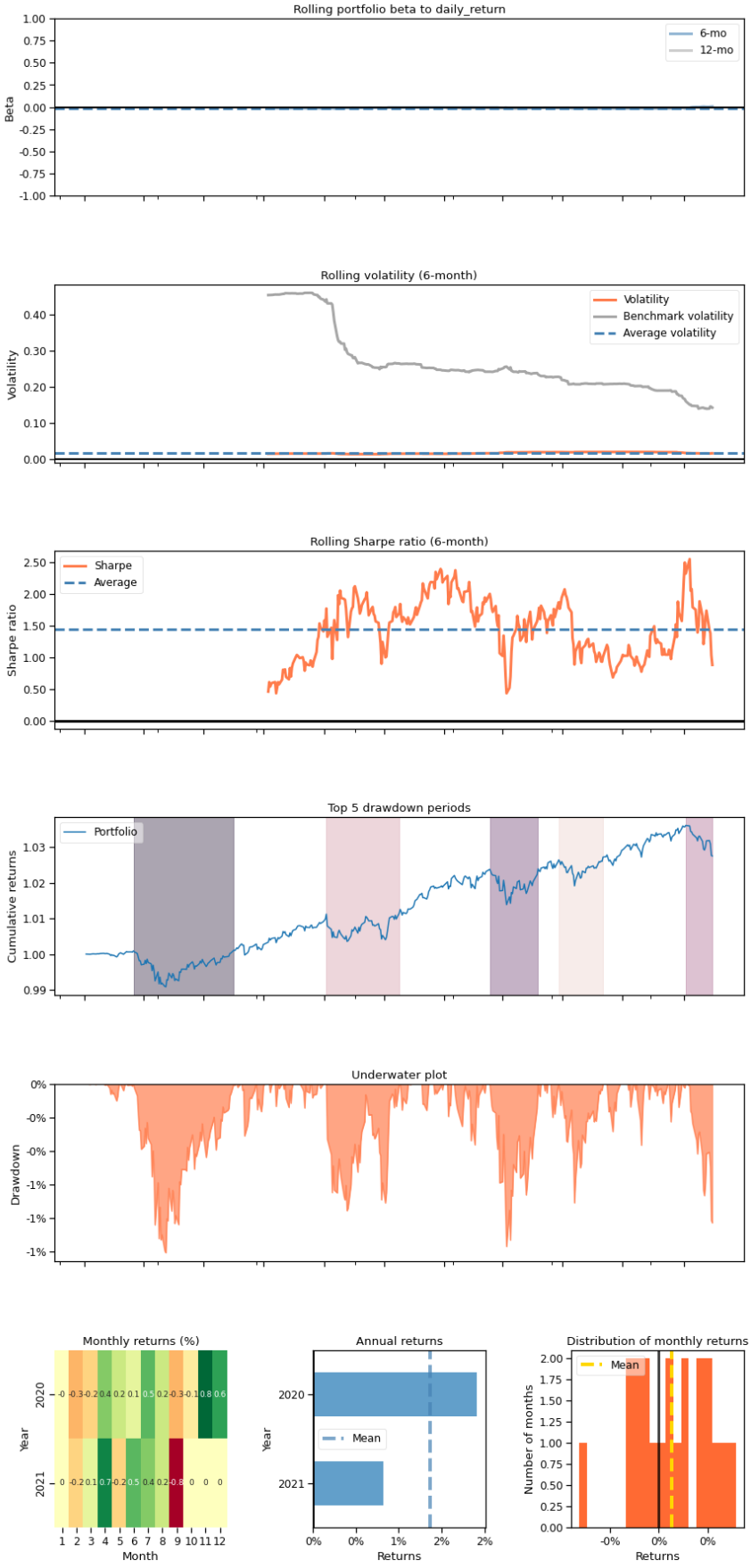

Fig B4 A2C Backtest Plots

5. SAC

Backtest	
Annual return	18.45%
Cumulative returns	34.41%
Annual volatility	13.64%
Sharpe ratio	1.31
Calmar ratio	2.54
Stability	0.96
Max drawdown	-7.28%
Omega ratio	1.26
Sortino ratio	1.89
Skew	NaN
Kurtosis	NaN
Tail ratio	1.04
Daily value at risk	-1.64%
Alpha	0.23
Beta	-0.07

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	7.28	2021-02-16	2021-03-05	2021-03-29	30
1	7.27	2020-09-03	2020-09-24	2020-11-06	47
2	5.05	2020-02-20	2020-03-24	2020-04-20	43
3	5.01	2021-04-27	2021-05-13	2021-06-07	30
4	4.32	2021-01-13	2021-02-01	2021-02-09	20

Table B5 SAC Backtest Stats

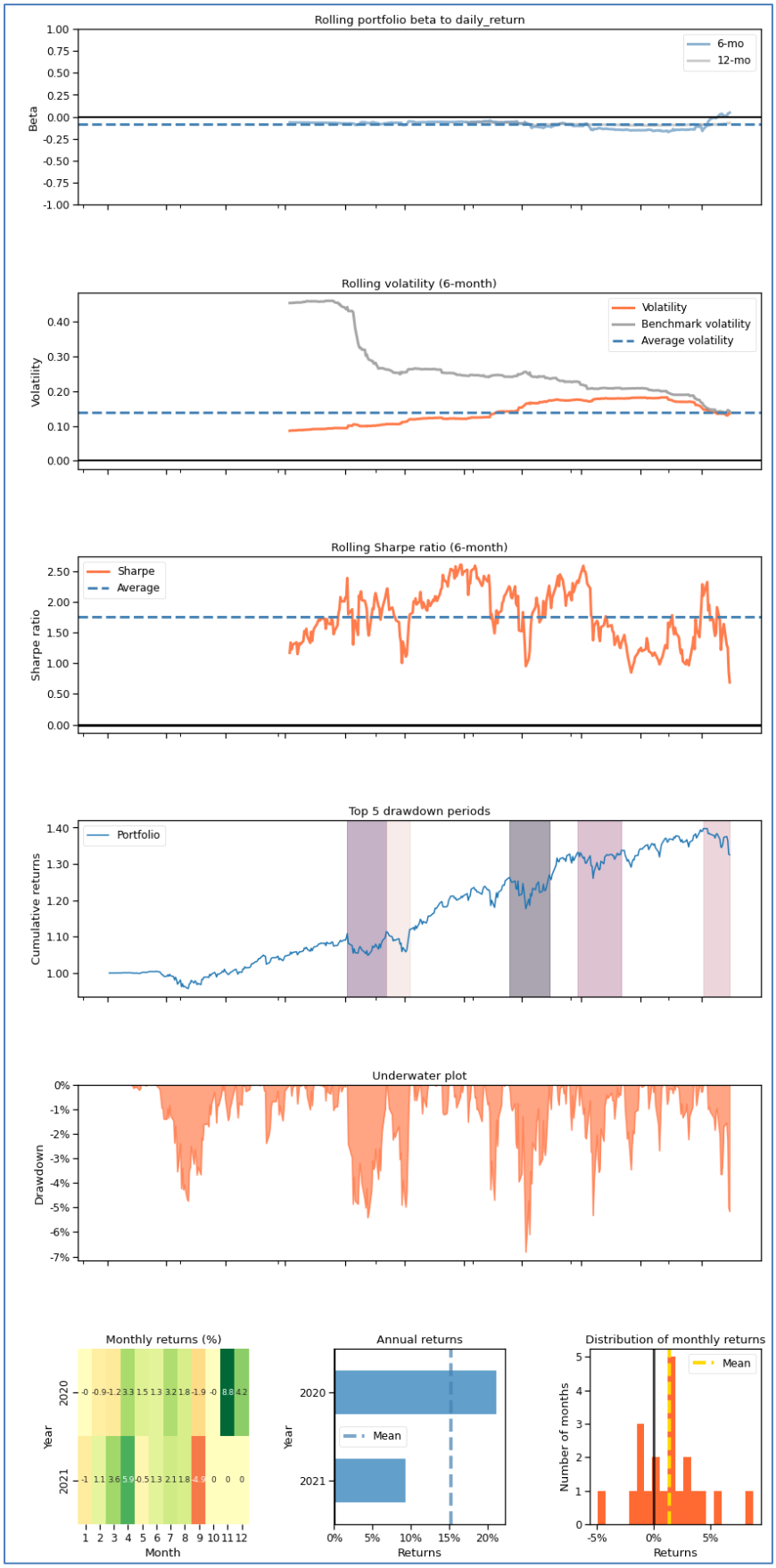

Fig B5 SAC Backtest Plots

Appendix C (Stop Loss Environment)

1. DDPG

Backtest	
Annual return	12.52%
Cumulative returns	22.87%
Annual volatility	12.67%
Sharpe ratio	1.00
Calmar ratio	1.77
Stability	0.94
Max drawdown	-7.06%
Omega ratio	1.19
Sortino ratio	1.40
Skew	NaN
Kurtosis	NaN
Tail ratio	0.98
Daily value at risk	-1.54%
Alpha	0.16
Beta	-0.05

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	7.07	2021-02-16	2021-03-05	2021-04-05	35
1	6.71	2021-07-06	2021-07-20	2021-09-03	44
2	5.32	2021-09-07	2021-09-30	NaT	NaN
3	5.11	2020-09-03	2020-09-24	2020-10-13	29
4	5.02	2021-04-27	2021-05-13	2021-06-07	30

Table C1 DDPG Backtest Stats

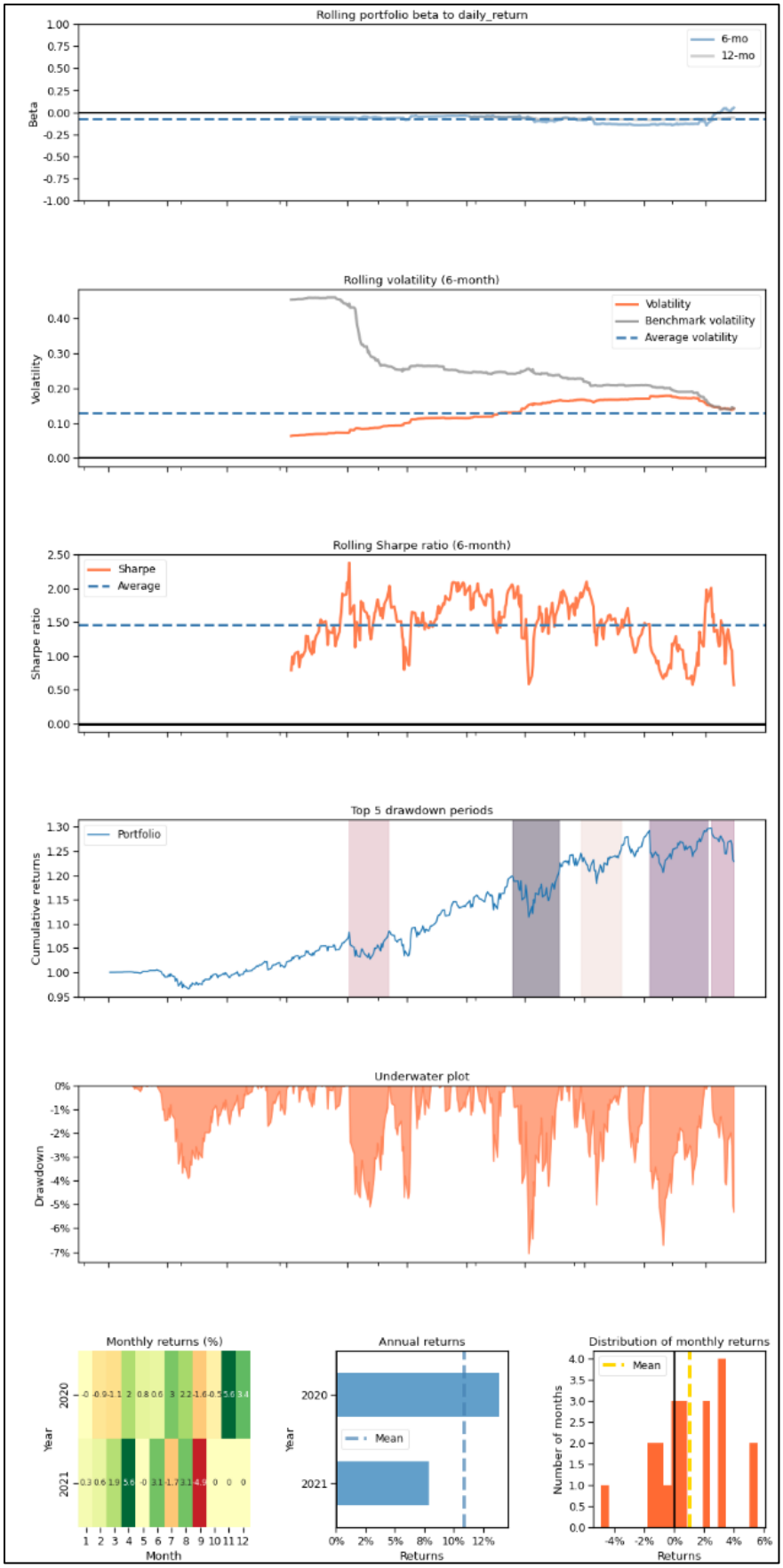

Fig C1 DDPG Backtest Plots

2. TD3

Backtest	
Annual return	14.01%
Cumulative returns	25.74%
Annual volatility	12.14%
Sharpe ratio	1.14
Calmar ratio	1.90
Stability	0.94
Max drawdown	-7.39%
Omega ratio	1.22
Sortino ratio	1.63
Skew	NaN
Kurtosis	NaN
Tail ratio	1.06
Daily value at risk	-1.47%
Alpha	0.17
Beta	-0.06

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	7.39	2021-02-16	2021-03-05	2021-04-05	35
1	6.15	2020-09-03	2020-11-02	2020-11-17	54
2	5.39	2021-04-19	2021-05-13	2021-06-07	36
3	3.97	2021-09-07	2021-09-29	NaT	NaN
4	3.81	2020-02-20	2020-03-23	2020-05-21	66

Table C2 TD3 Backtest Stats

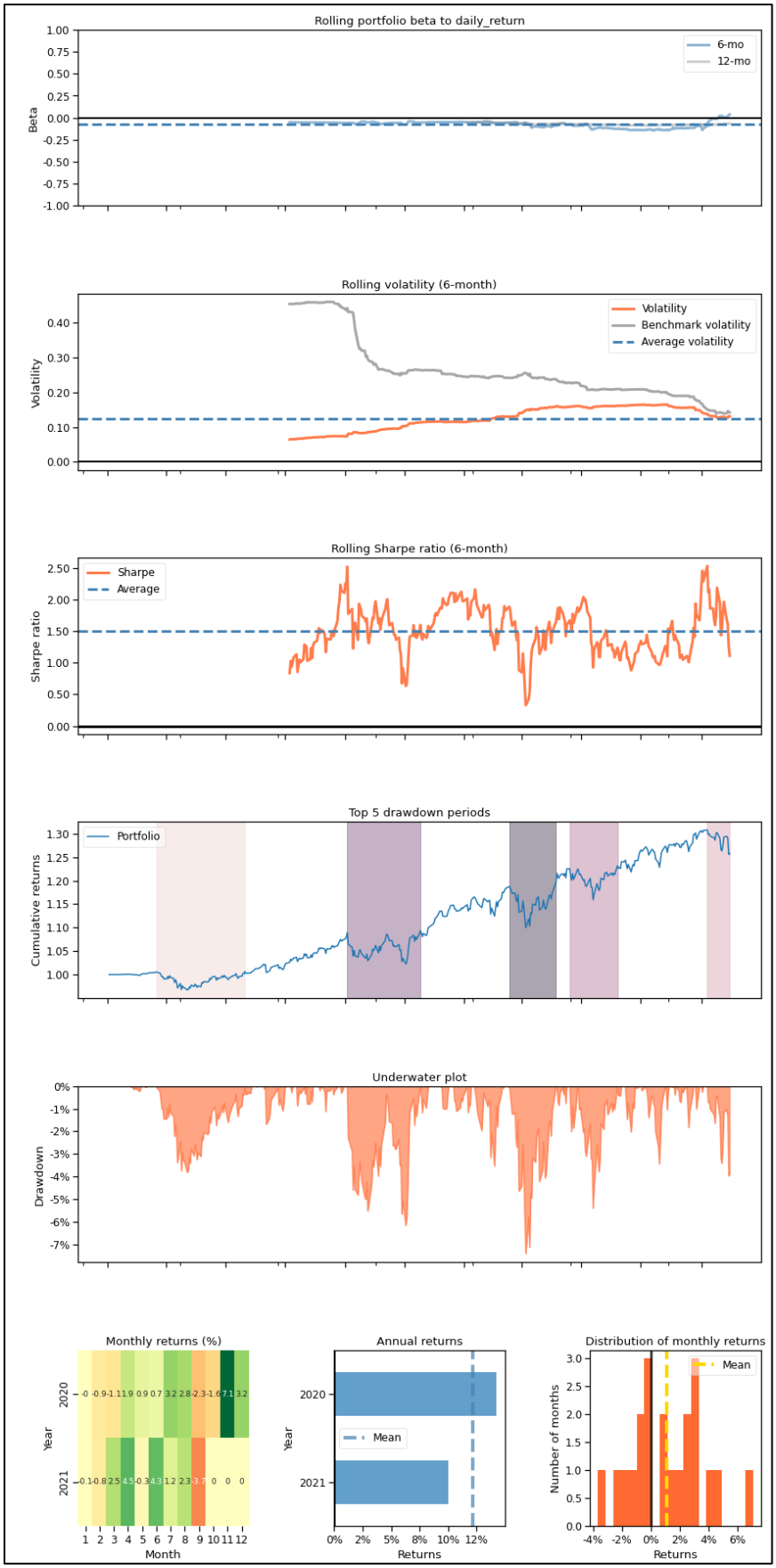

Fig C2 TD3 Backtest Plots

3. PPO

Backtest	
Annual return	1.17%
Cumulative returns	2.06%
Annual volatility	1.31%
Sharpe ratio	0.90
Calmar ratio	1.64
Stability	0.91
Max drawdown	-0.71%
Omega ratio	1.17
Sortino ratio	1.27
Skew	NaN
Kurtosis	NaN
Tail ratio	0.96
Daily value at risk	-0.16%
Alpha	0.01
Beta	-0.01

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	0.72	2020-02-13	2020-03-23	2020-06-08	83
1	0.68	2021-02-16	2021-03-05	2021-04-01	33
2	0.63	2021-04-06	2021-05-13	2021-06-07	45
3	0.53	2021-09-07	2021-09-30	NaT	NaN
4	0.52	2021-07-06	2021-07-20	2021-08-26	38

Table C3 PPO Backtest Stats

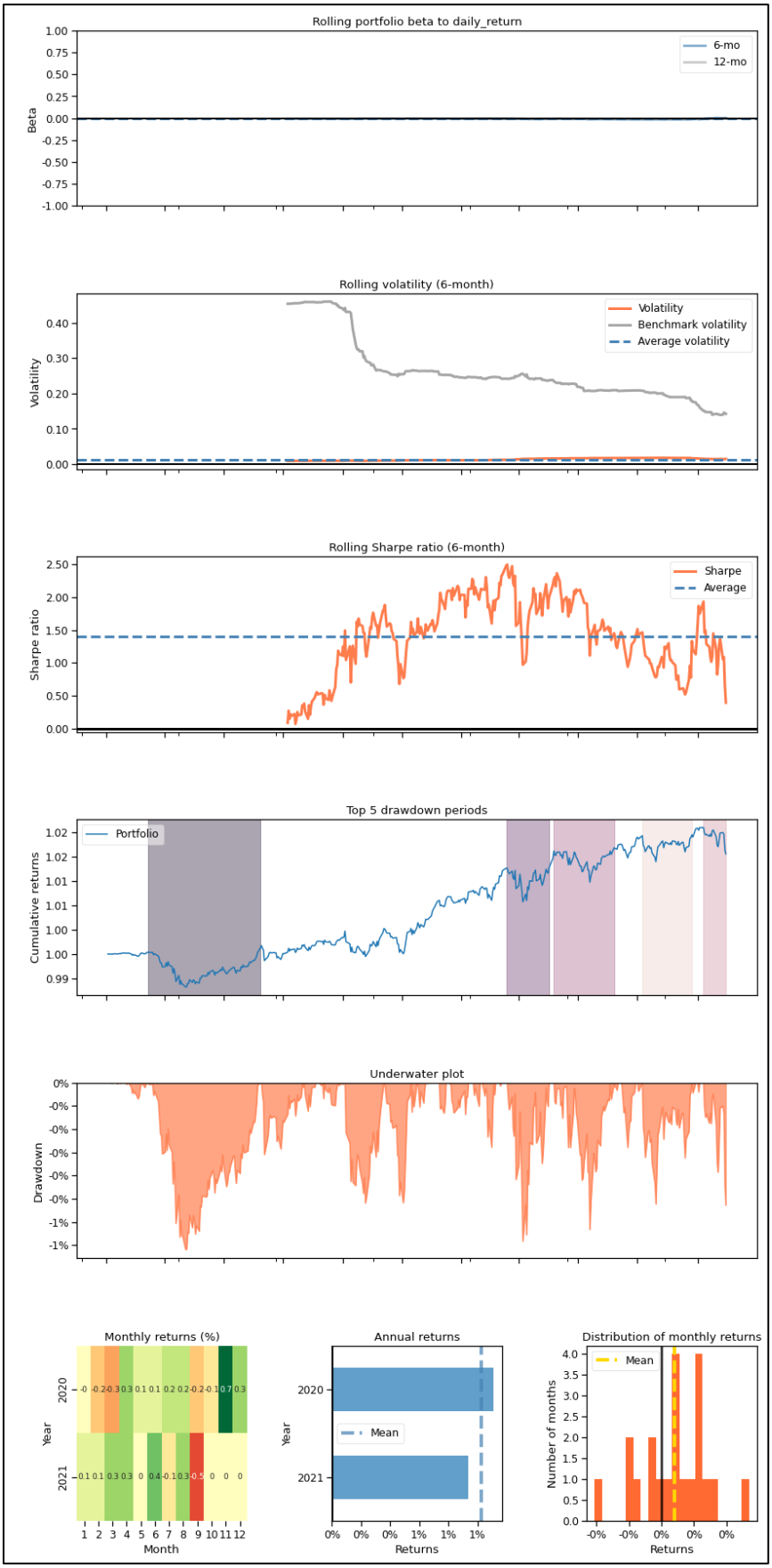

Fig C3 PPO Backtest Plots

4. A2C

Backtest	
Annual return	1.15%
Cumulative returns	2.03%
Annual volatility	1.47%
Sharpe ratio	0.79
Calmar ratio	1.43
Stability	0.91
Max drawdown	-0.80%
Omega ratio	1.15
Sortino ratio	1.10
Skew	NaN
Kurtosis	NaN
Tail ratio	0.99
Daily value at risk	-0.18%
Alpha	0.01
Beta	-0.01

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	0.81	2020-02-13	2020-03-24	2020-06-08	83
1	0.79	2021-02-16	2021-03-05	2021-04-01	33
2	0.73	2021-04-19	2021-05-13	2021-08-02	76
3	0.63	2020-09-03	2020-09-24	2020-10-13	29
4	0.60	2020-10-13	2020-11-02	2020-11-10	21

Table C4 A2C Backtest Stats

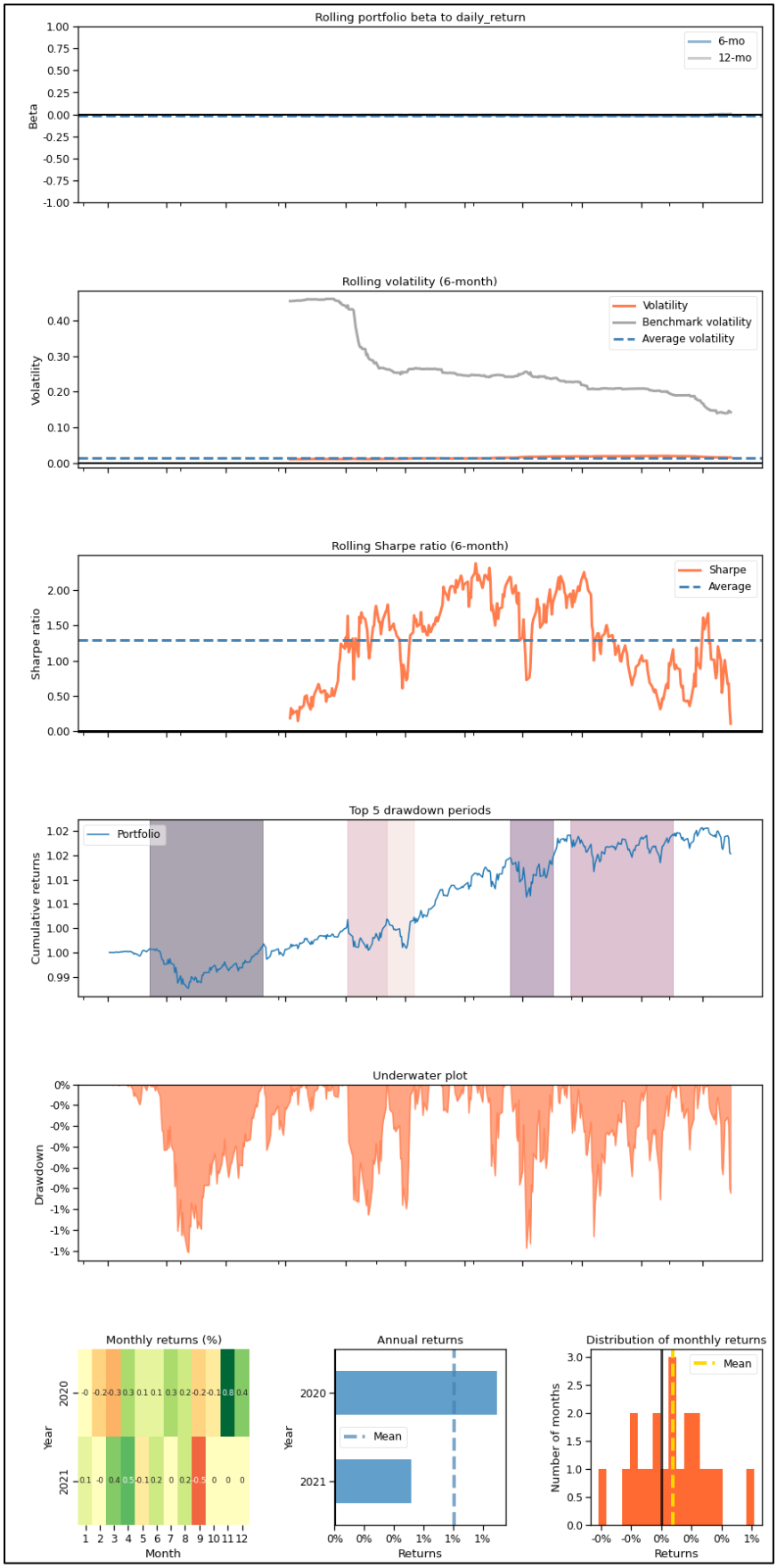

Fig C4 A2C Backtest Plots

5. SAC

Backtest	
Annual return	2.34%
Cumulative returns	4.12%
Annual volatility	3.13%
Sharpe ratio	0.76
Calmar ratio	1.28
Stability	0.91
Max drawdown	-1.82%
Omega ratio	1.14
Sortino ratio	1.05
Skew	NaN
Kurtosis	NaN
Tail ratio	1.00
Daily value at risk	-0.385%
Alpha	0.03
Beta	-0.01

Worst drawdown periods	Net drawdown in %	Peak date	Valley date	Recovery date	Duration
0	1.82	2021-02-16	2021-03-05	2021-03-29	30
1	1.65	2021-08-31	2021-09-29	NaT	NaN
2	1.57	2021-04-27	2021-06-21	2021-07-26	65
3	1.26	2020-09-03	2020-09-24	2020-10-13	29
4	1.21	2020-10-13	2020-11-02	2020-11-10	21

Table C5 SAC Backtest Stats

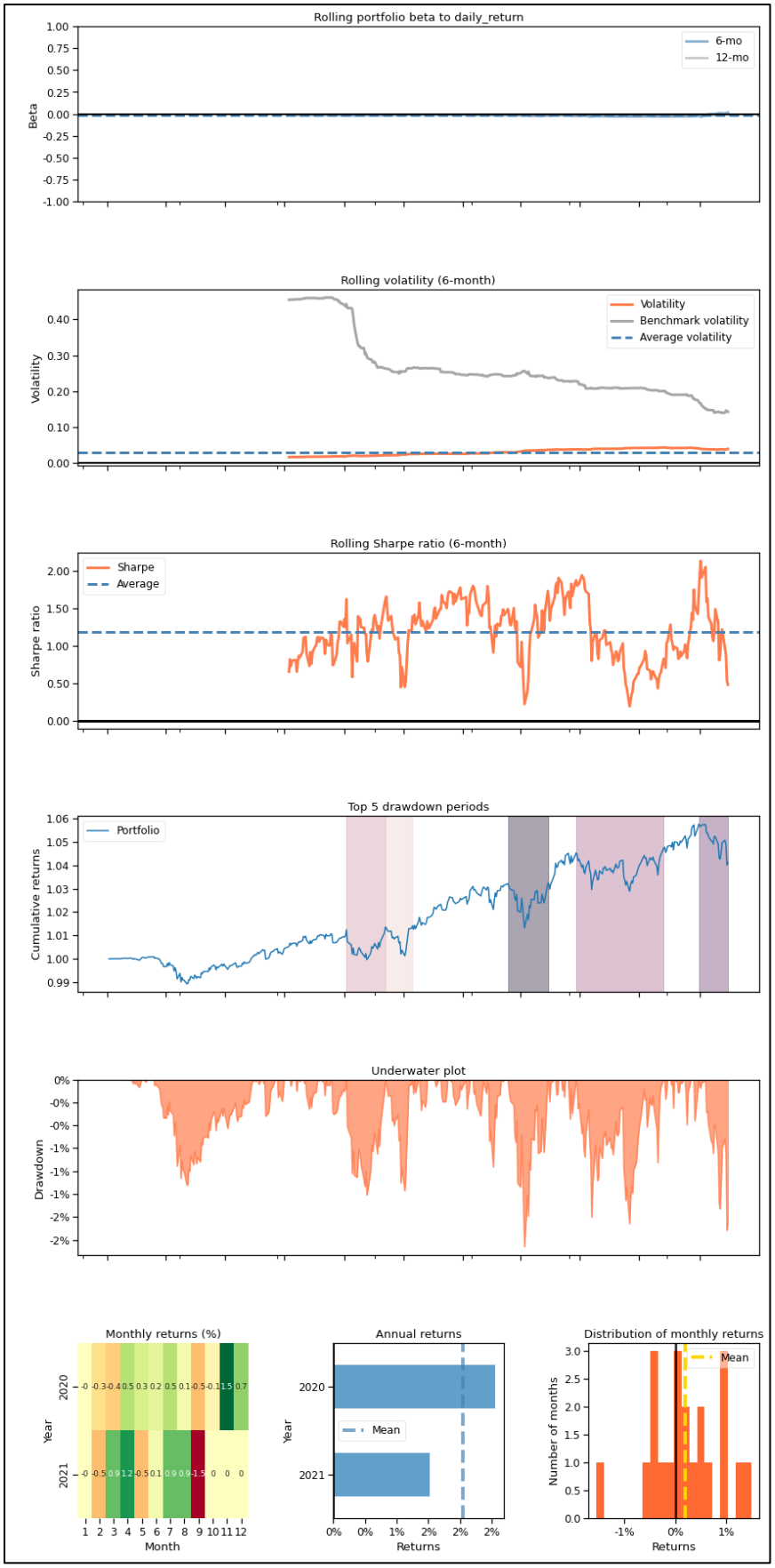

Fig C5 SAC Backtest Plots