Reinforcement Learning for Optimal Portfolio Management in Dynamic Financial Environments

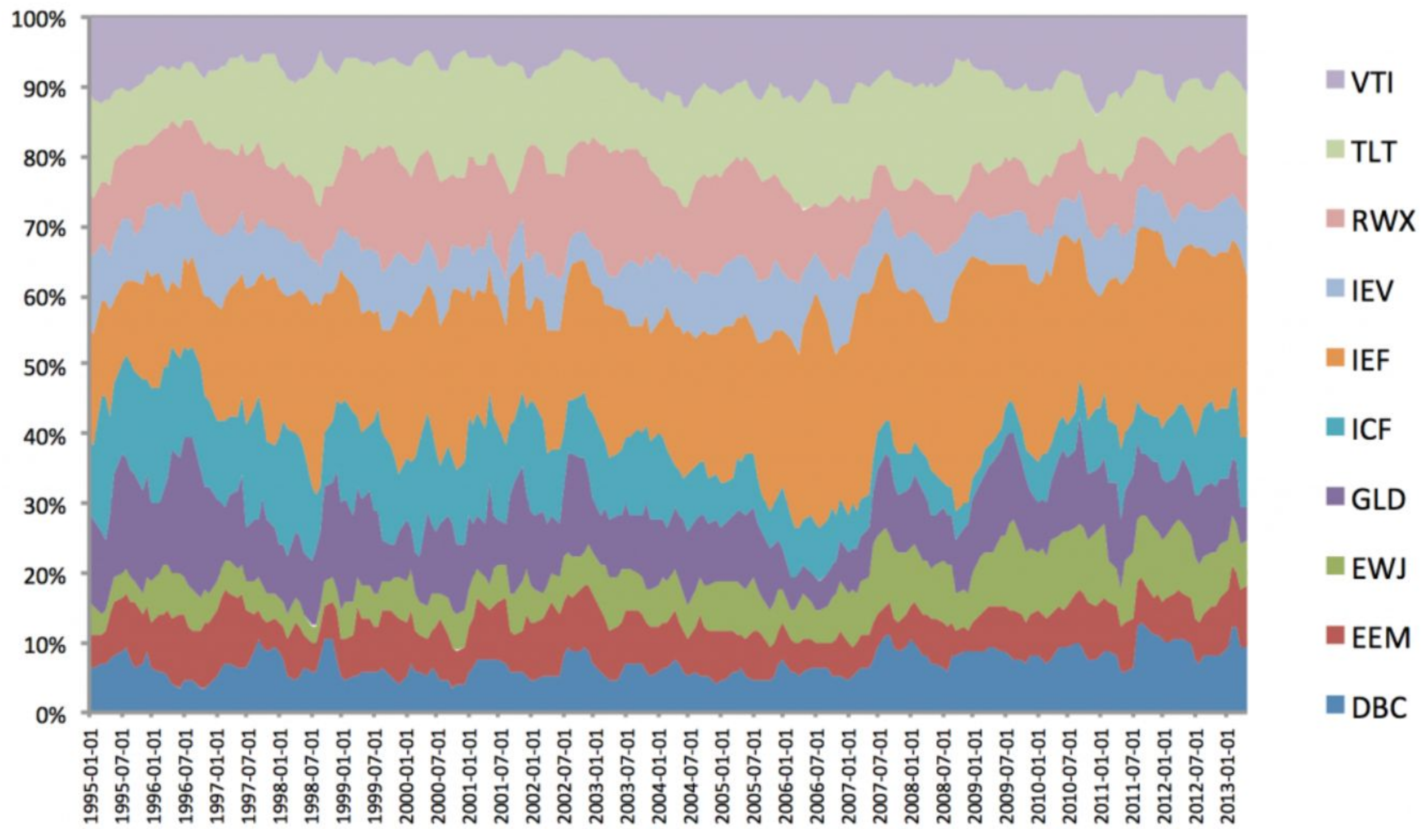
Mentor - Dr. Chandrashekar Lakshmi Narayanan
Jerry John Thomas 111901055
Ishwar Govind 111901024

Contents

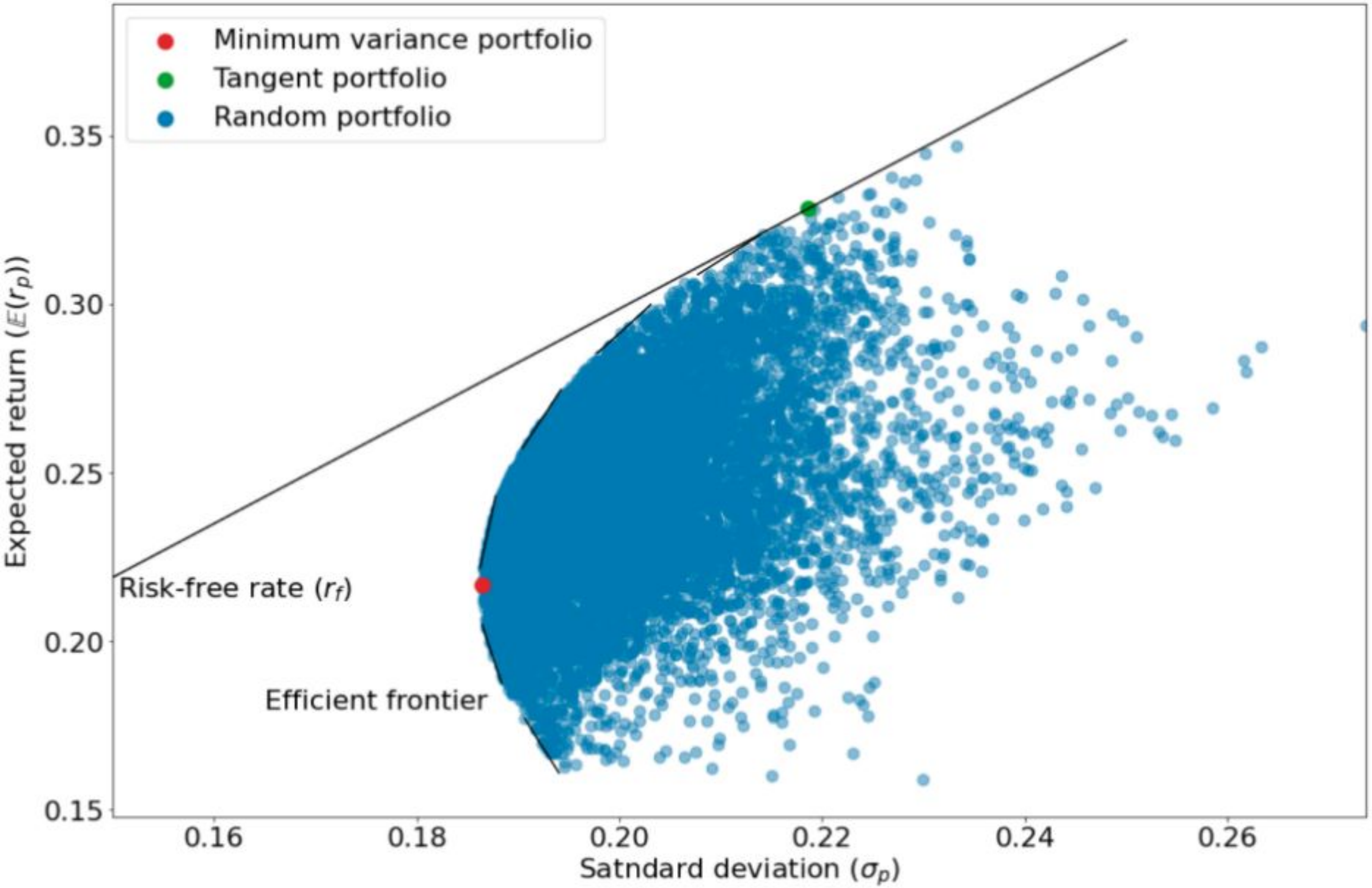
Introduction

- Portfolio optimization seeks to find the most efficient combination of investments by re-distributing stocks/shares/bonds that will provide the highest return at the lowest risk

- We look at different ways to achieve this optimization



https://investresolve.com/dynamic-asset-allocation-for-practitioners-part-4-naive-risk-parity/


Minimum variance portfolio
Tangent portfolio
Random portfolio
0.35
0.30
0.25
0.20
0.15
Expected return ($\mathbb{E}(r_p)$)
Risk-free rate (r_f)
Efficient frontier
0.16
0.18
0.20
0.22
0.24
0.26
Satndard deviation (σ_p)

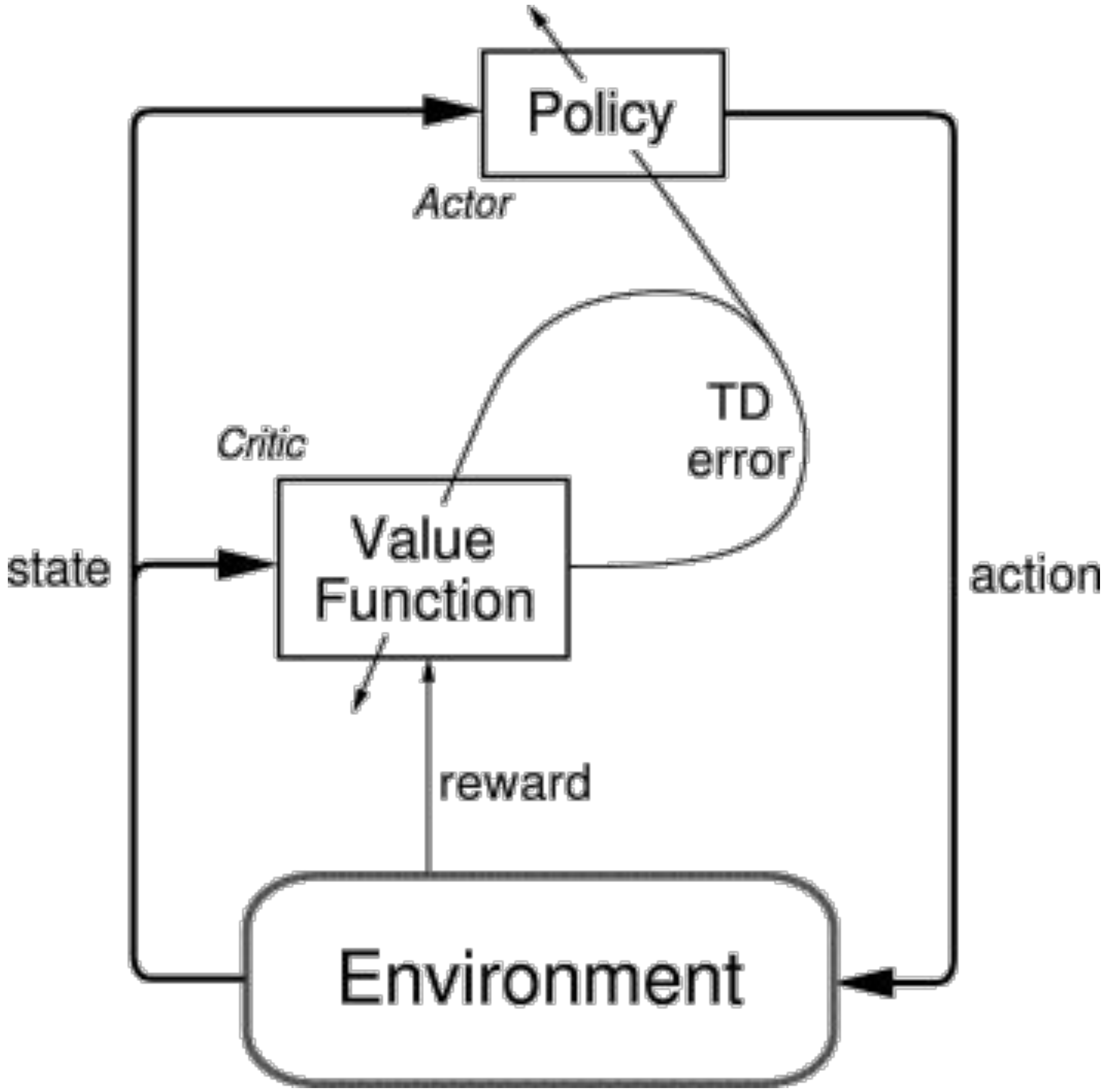
Why Reinforcement Learning?

- Most Supervised Learning model are used for predicting stock price movements.
- The actions of buying and selling cannot be done by supervised learning models.
- RL provides algorithms to which can train agents to perform actions by using “intelligence”.
- Model Free Deep Reinforcement Learning was shown to be successful in previous attempts at algorithmic trading.

Sharpe Ratio

- Measure of risk-adjusted return
- Used to compare the performance of Portfolios
- Higher Sharpe ratio indicates better risk-adjusted returns
- It is important to note that the Sharpe Ratio should not be used in isolation, as it does not take into account other factors such as liquidity, fees, or taxes.

$$S = \frac{R_p - R_f}{\sigma_p}$$



$$\underbrace{A(s,a)}_{} = \underbrace{Q(s,a)}_{\text{q value for action a in state s}} - \underbrace{V(s)}_{\text{average value of that state}}$$

$$A(s,a) = \boxed{Q(s,a)} - V(s)$$

$$r + \gamma V(s')$$

$$A(s,a) = \underbrace{r + \gamma V(s') - V(s)}_{\text{TD Error}}$$

Environment

- State Space
  - [Current Balance , [Stock Prices] , [Shares Owned]]
  - Dimension - 2n+1, Range [0,∞)
- Action Space
  - [Stock Action]
  - Dimension :- n , Range [-1 , 1]
- Reward
  - $$R_t = Assets_t - Assets_{t-1}$$

$$Assets = S_0 + \sum_{i=1}^{n} S_i * S_{n+i}$$

Proximal Policy Optimization

- PPO is a policy gradient reinforcement learning method.
- It is relatively easy to use and is efficient at solving complex problems
- It focuses more on sample efficiency and stability

Updating Policy

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \quad g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right)$$

Updating Value Function

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2$$

Schulman, John et al. "Proximal Policy Optimization Algorithms." *ArXiv* abs/1707.06347 (2017): n. pag.

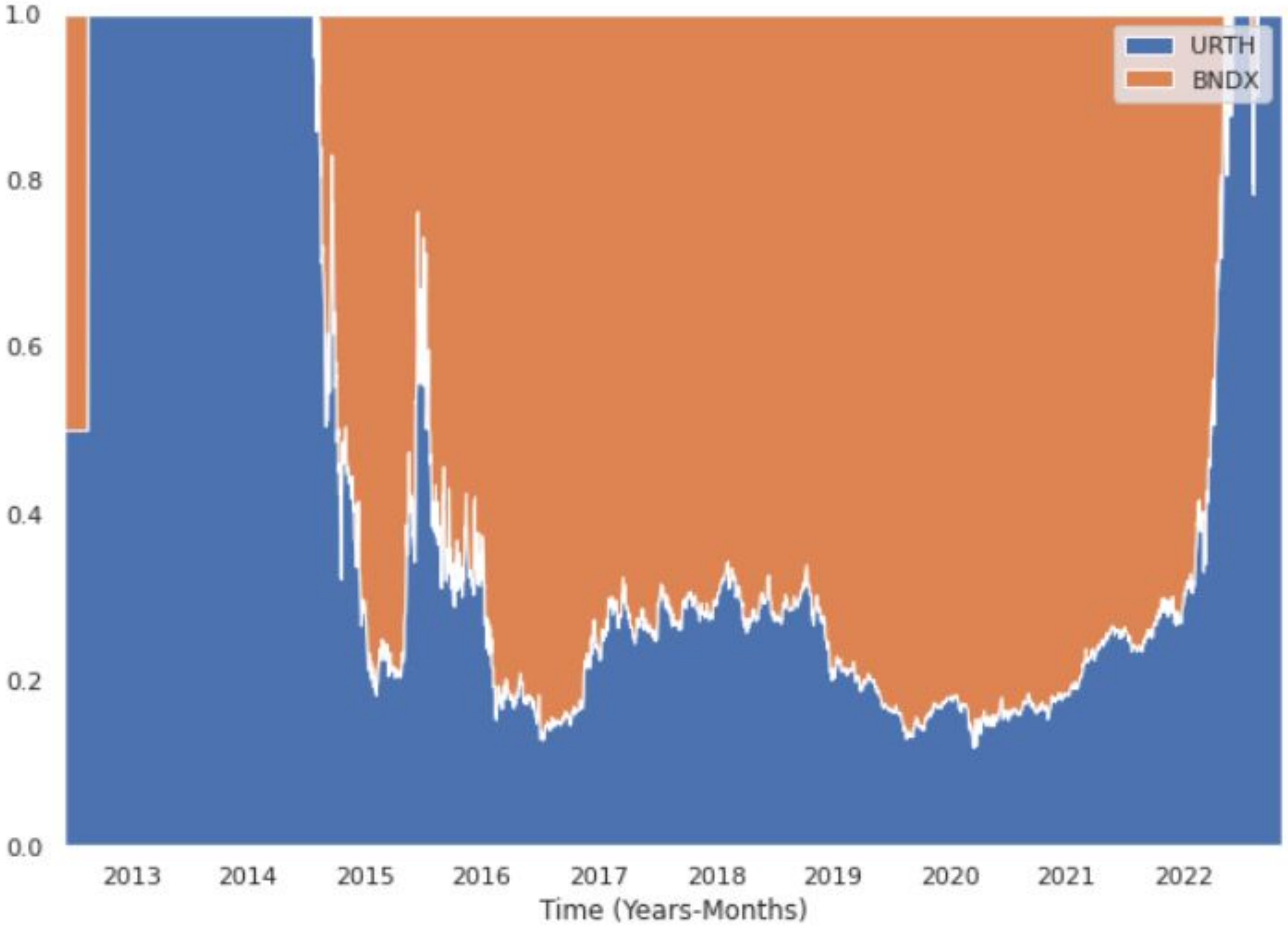
Exp 1 : Maximizing Sharpe

Stocks

- iShares MSCI World ETF (URTH)
- Vanguard Total International Bond Index Fund ETF (BNDX) *(risk free)*

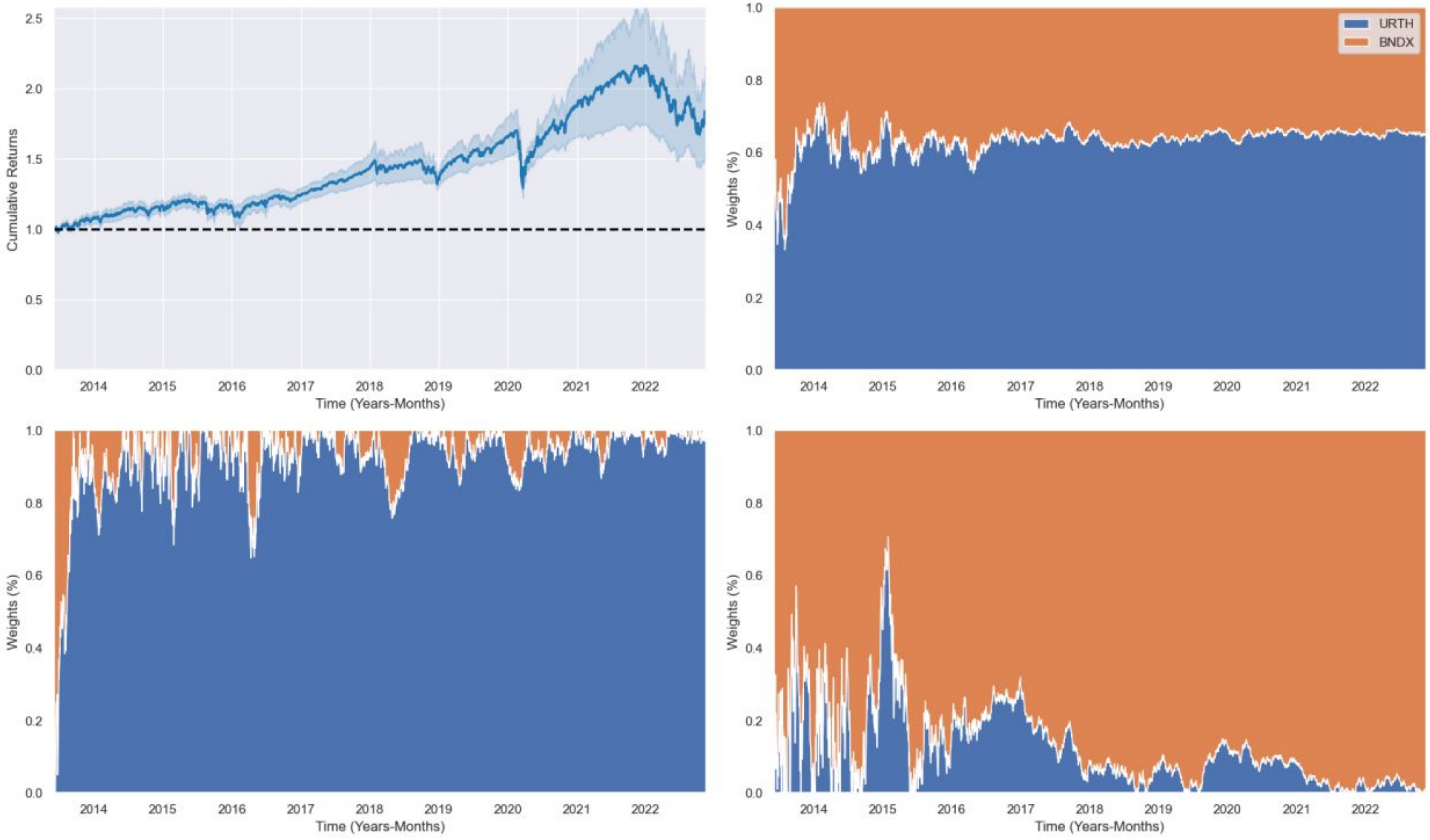
Convex Optimization

Cumulative Returns



Portfolio Distribution

Sharpe as PPO reward



Exp 2: Risk Sensitive Actor-Critic

- What is Risk ?
  - Potential for financial loss or uncertainty associated with investment decisions.
  - Level of price volatility associated with a particular asset

- What is Risk Sensitive?
  - Neither risk averse nor risk seeking but adapting the behavior based on circumstances.

Exp 2: Risk Sensitive Actor-Critic

We have created a continuous action space Actor Critic algo inspired from Borkar's paper

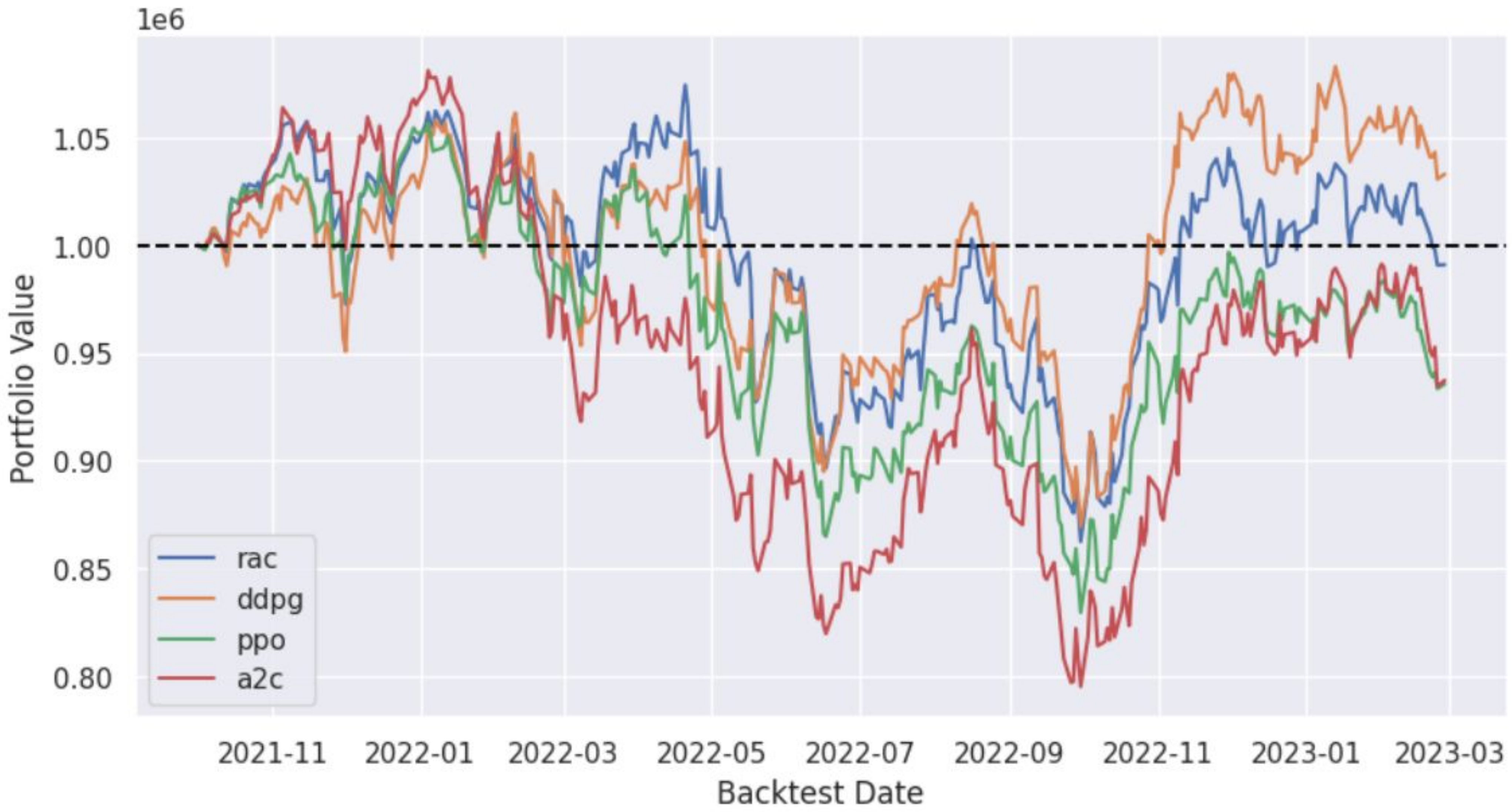
$$Q_{loss} = (Q_{target} - Q(s_t, a_t))^2$$

$$Q_{target} = \frac{Q(s_{t+1}, a_{t+1})}{Q(s_{ref}, a_{ref})} * e^{-R} - \alpha * log(\pi(s', a'))$$

$$\nabla_\theta J = \nabla_\theta \frac{1}{|B|} \sum_{s \in B} (Q_\phi(s, a) - \alpha \log \pi_\theta(s', a'))$$

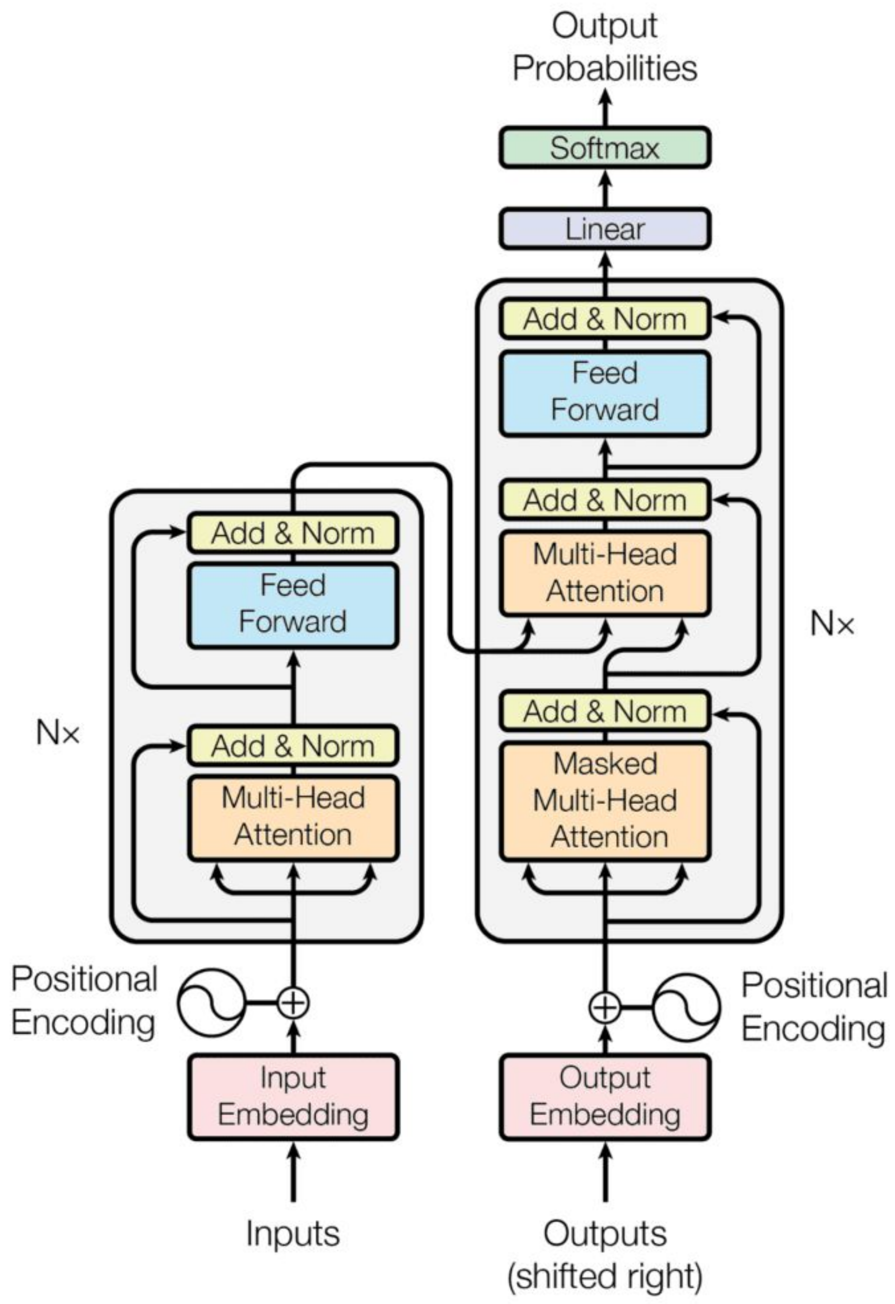
Borkar, V.s. (2001). A sensitivity formula for risk-sensitive cost and the actor–critic algorithm. Systems & Control Letters. 44. 339-346. 10.1016/S0167-6911(01)00152-9.

Cumulative Returns



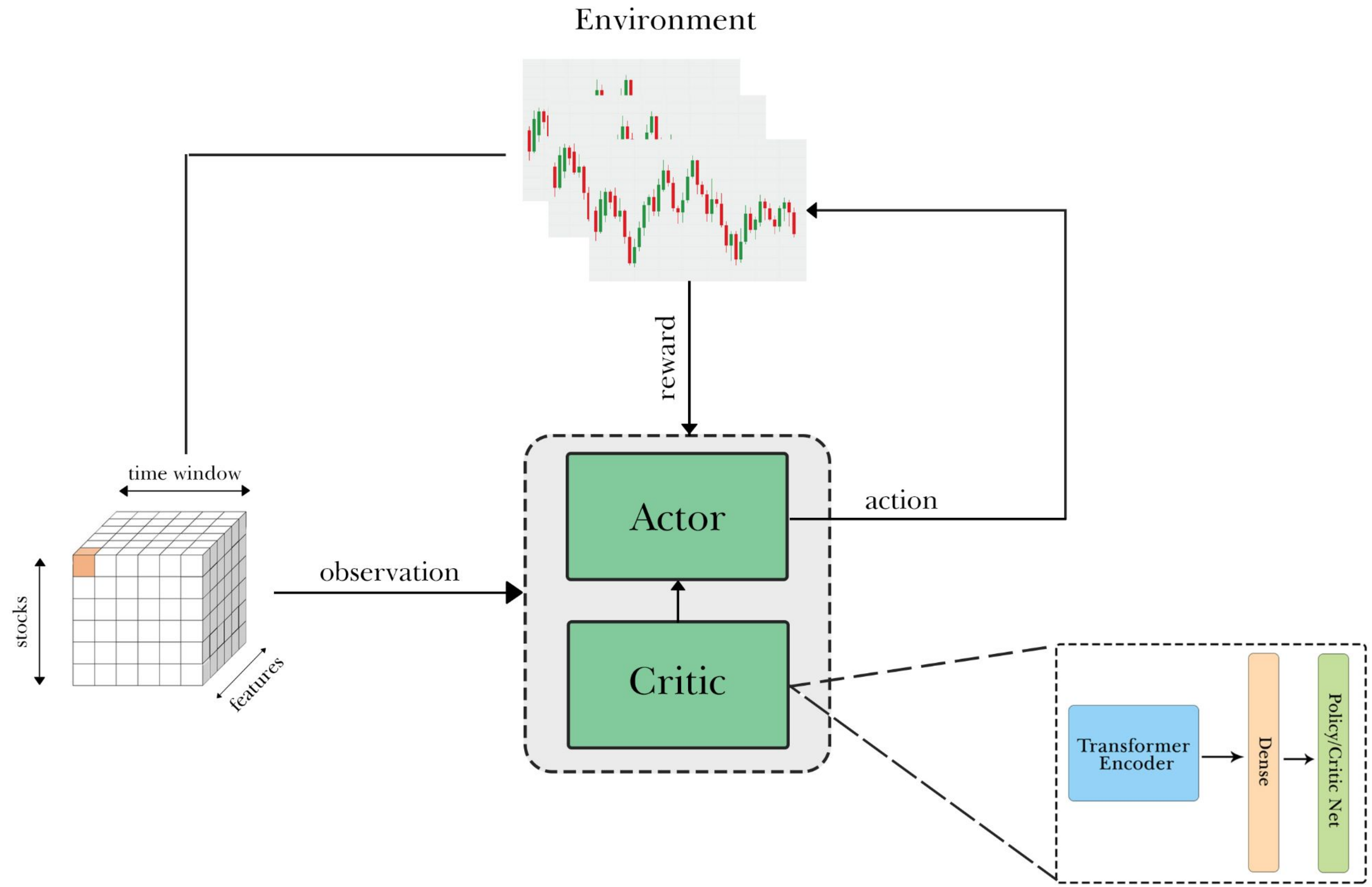
Transformer

- The Transformer architecture consists of an encoder-decoder framework.
- Transformer Encoder
  - Core component of the Transformer architecture.
  - Processes input sequence and generates encoded representations
  - Captures global context and positional information.
- Transformer models have shown state of the art performance in a number of time series forecasting problems



Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

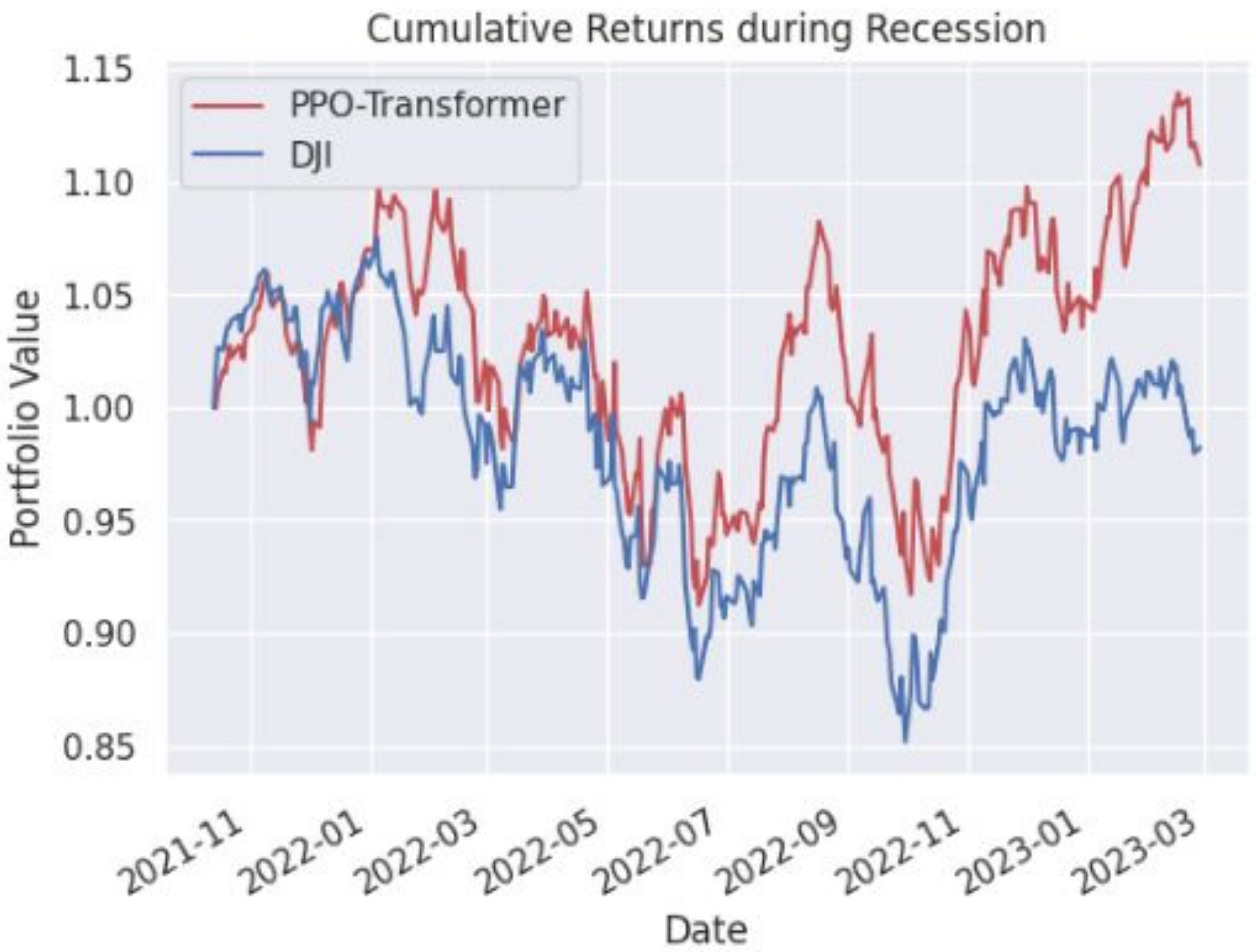
Exp 3: Transformer-based PPO



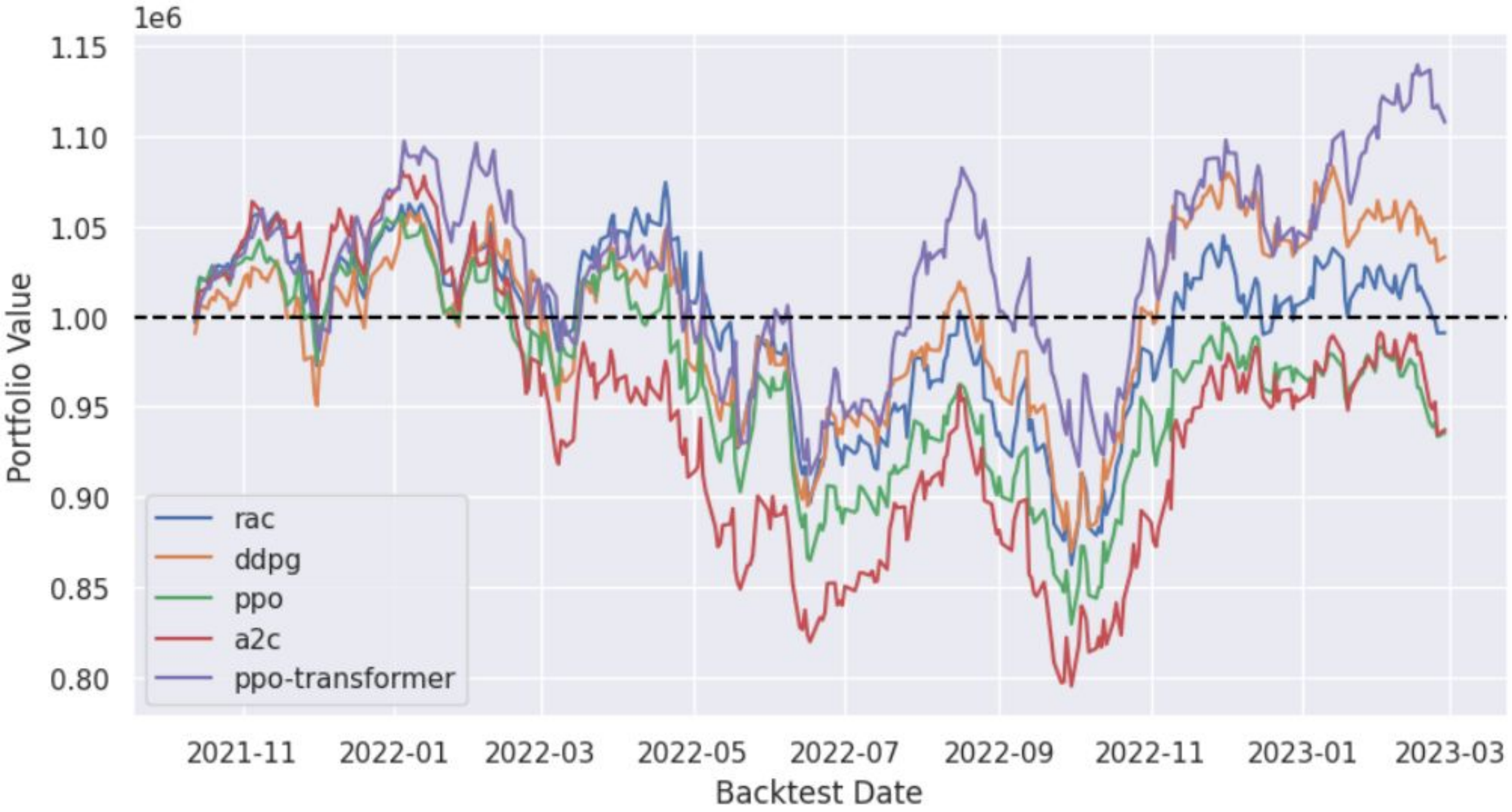
Updated Environment

- State Space
  - Dimension - (Time window, No. of stocks,stock features + previous weights)
  - In our case (7,28,14)
  - Range [0,∞)
- Action Space
  - [Stock Action]
  - Dimension :- n , Range [-1 , 1]

Comparing Transformer-PPO with Dow Jones Index



Comparison of different RL algorithms



	Risk Sensitive AC	DDPG	Transformer PPO
Annual return	-0.644%	2.346%	7.758%
Cumulative returns	-0.901%	3.302%	10.771%
Annual volatility	17.335%	17.381%	18.953%
Sharpe ratio	0.05	0.22	0.49
Max drawdown	-19.726%	-18.096%	-16.851%
Daily value at risk	-2.181%	-2.175%	-2.351%

Future Work and Scope

- Understanding the agent’s actions to get insights into better trading strategies.
- Biasing the agent’s reward
- Explore imitation Learning and transfer learning in the financial market
- Include social media sentiment in the environment
- Create an Explainable AI framework
- Exploring the use of graph neural network encoding

References

- https://github.com/JerryJohnThomas/PortfolioOptimisation
- https://github.com/ishwargov/PortfolioOptimization
- Finrl stock env https://github.com/AI4Finance-Foundation/FinRL/blob/master/finrl/meta/env stock trading/env stocktrading.py
- Schulman, John, et al. "Proximal policy optimization algorithms. OpenAI
- Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- V. Borkar, “A sensitivity formula for risk-sensitive cost and the actor–critic algorithm,” Systems Control Letters
- Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, “Deep direct reinforcement learning for financial signal representation and trading,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 3, pp. 653–664, 2017.
- U. Upadhyay, N. Shah, S. Ravikanti, and M. Medhe, “Transformer based reinforcement learning for games
- P. L.A. and M. Ghavamzadeh, “Actor-critic algorithms for risk-sensitive mdps,” Advances in Neural Information Processing Systems, 02 2013

Thank You