# References


0. Sutton, Barto. “Reinforcement Learning: An Introduction”. MIT Press.
1. Multi Armed Bandit in “Stochastic modelling”. World Quant University.
2. Cluster Analysis in “Applied Multivariate Statistical Analysis”. Penn State University
3. Huo, Fu. “Risk-aware multi-armed bandit problem with application to portfolio selection”. Royal Society of Open Science. 2017.
4. Lu, et al. “Contextual Multi-Armed Bandits”. In AISTATS 2010.
5. Ugur Yildrim. “An Overview of Contextual Bandits”.
6. “Hierarchical Clustering”. In Scikit-learn Documentation.


# Multi-Armed Bandit (MAB) algorithm for daily stock picking automation challenge

The goal of this study is to develop an automated stock picking system that leverages the Multi-Armed Bandit (MAB) algorithm to optimize daily trading decisions where the agent chooses one stock that maximizes the expected return for a given holding horizon. The policy of the agent chooses the action among the symbols $a_t \in A = \{\text{symbols}\}$, hold for $HOLD$ business days, and optimise its action value function based on the observed return from the historical data based simulation $r_t := p_{t+H}/p_t - 1$.

In this study, we compare the performance of several MAB algorithms in terms of the stock picking performance as the cumulative daily return: the stock market in our simulation is from the Lehman brothers crisis to the recovery period (*: 2007/Q3 - 2010/Q4), of 30 US stocks half of each from financial and non-financial sectors.

(*) Although the original suggestion was to study upto 2 months of daily data, optimising 15-30 dimensions of function just for 60 data points does not really make much sense hence we enlarged the date range.

# Methodology and Investigation

## Style of stock return: clustering analysis of financial and non-financial sector symbols

The aim in this section is to perform stock symbol clustering by closeness of the stock return among symbols. The measure of the closeness we employed is Pearson's correlation to observe the granular landscape of the stock returns among the selected 30 symbols within financial and non-financial sectors.

By defining the distance metric as the residual of correlation throughout the date range of study by $d(x_i, x_j) := 1 - Corr[x_i, x_j]_{t=0...T}$, we can apply clustering techniques to group similar stocks together. Note, the distance between the two subclusters need new metric as there is no direct correlation to measure their closeness between those virtual entities. We adopted the Ward's method [2] that searches the grouping output for the minimum variance within each resultant cluster.

Below is the result of the hierarchical clustering where similar two closest symbols merge together as a subcluster, so does the subcluster to a bigger subcluster, and so forth til all symbols group as a single cluster:

Symbols are clearly clustered by the sectors. This is consistent with our expectations, as stocks within the same sector often exhibit similar return patterns especially around the period of the Lehman brothers crisis and recovery on our study where the return of the financial companies were largely affected by the monetary policy governed by governments, resulting in synchronised price movements. Another observation is that the clustering result reflects some of prominent factors related to the company analysis. For instance, large-banks/investment-banks are grouped JPM-USB-WFC/GS-MS, institutional banks are closer to each other than consumer capitals - see GS&MS versus GE versus BAC.

Below is the correlation matrix by with the sorted index calculated by the hierarchical clustering done above. Granules of high correlation are clearly visible in the heatmap around the diagonal on several level of clustering

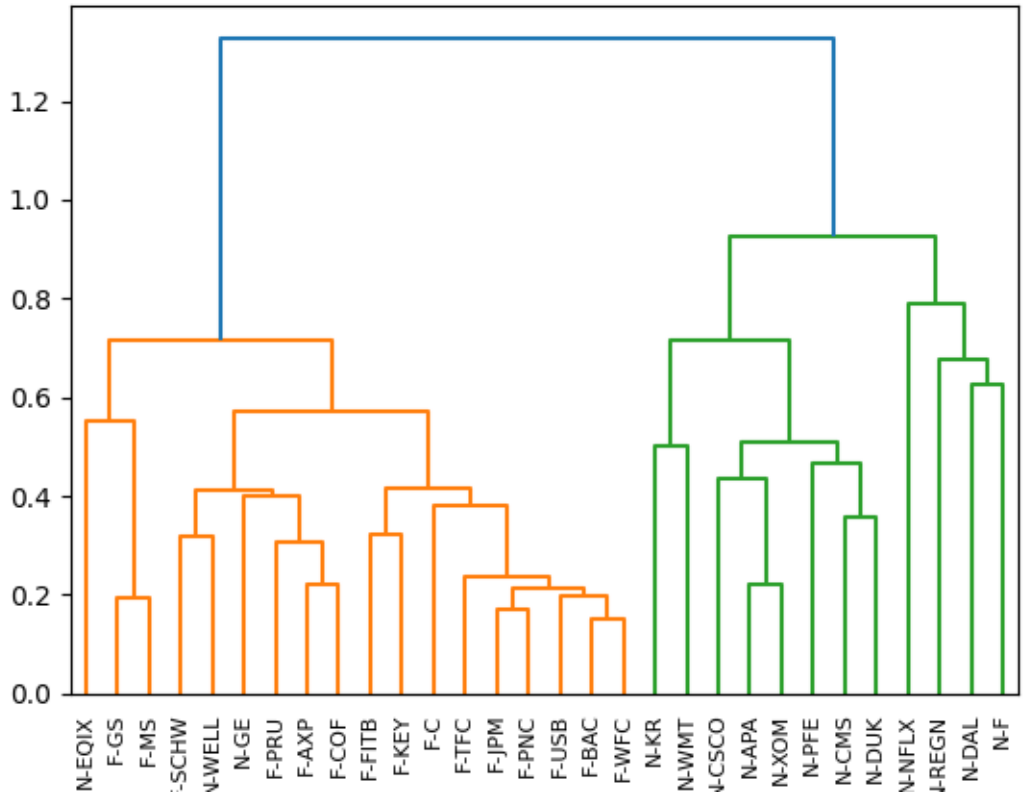


Figure 1: Hierarchical clustering of US 30 stocks by daily return

size such as financial versus non-finacial, institutional banks (GS, MS) versus commercial banks (JPM, PNC, USB, BAC, WFC), energy companies (APA to DUK) versus consumer industry (NFLX to to F), and et cetera.

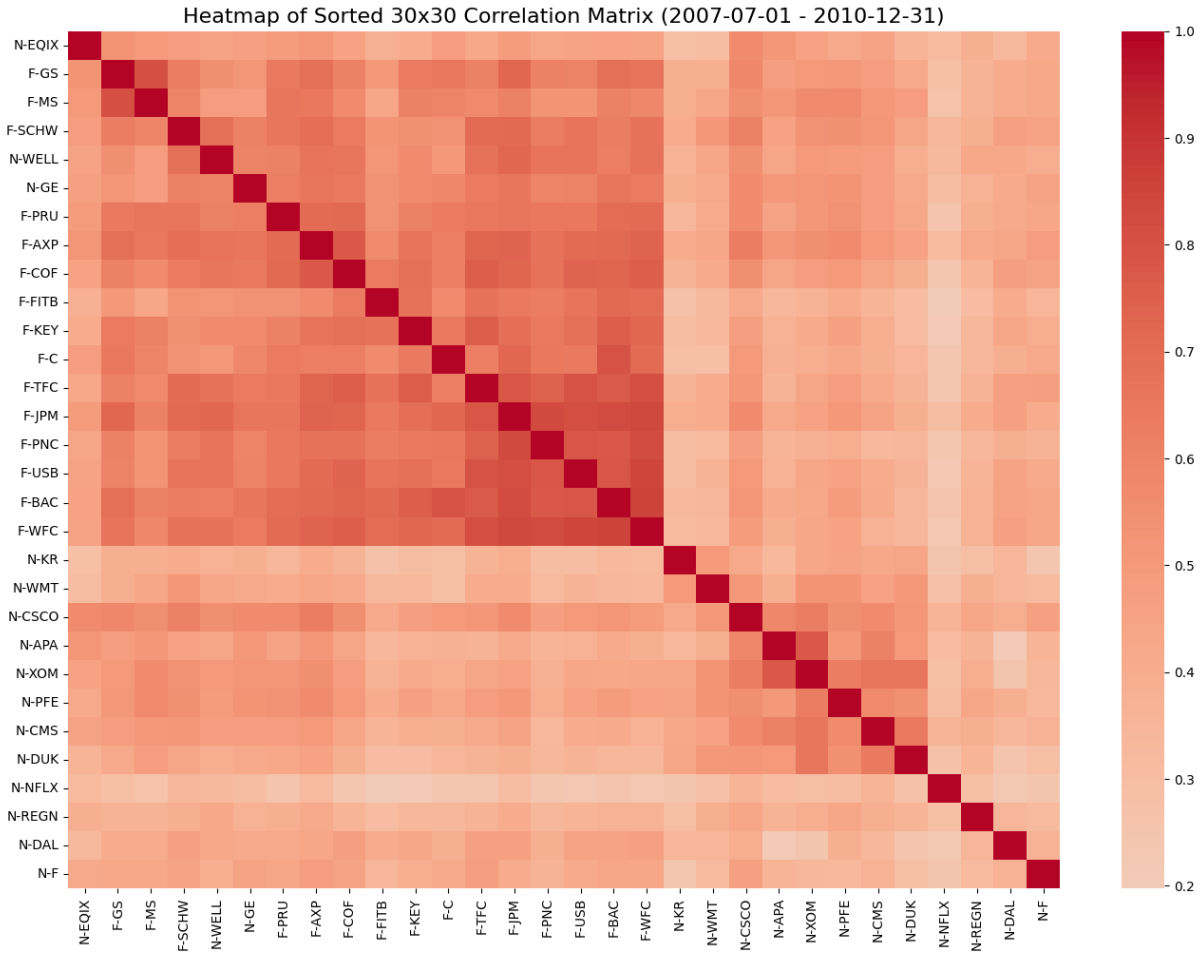


Figure 2: Pearson Correlation heatmap sorted by hierarchical clustering

## Multi-Armed Bandit (MAB) with $\epsilon$-greedy exploration

The $\epsilon$-greedy exploration directly injects random actions by the odds given by $\epsilon$.

The algorithm works as below:

- For each episode:
  - Initialise Q-value per action Q(a) as 0 for all actions
  - For each timestep $t$:
    - Choose the action $a_t$ by following:
      - If random() $< \epsilon$, select a random action $a_t \sim \text{Uniform}(A)$
      - Else, select the action amongst the highest estimated value $a_t = \text{argmax}_{a \in A}[Q(a)]$
    - Observe the reward $r_t$ for given $a_t$ as forward-looking return $p_{t+H}/p_t - 1$
    - Update the Q-value at t accordingly: $Q(a_t) \leftarrow \alpha \cdot r_t + (1 - \alpha) \cdot Q(a_t)$

Below is the verification plot for the equivalence of completely exploring MAB agent against the equally weighted portfolio strategy in terms of average return performance over many episodes. As shown in the plot, the asymptotic

performance of the MAB agent for many episodes is almost equal to that of the portfolio of randomly picked stocks at each reevaluation, of which its expected return is equivalent to the equally weighted portfolio:

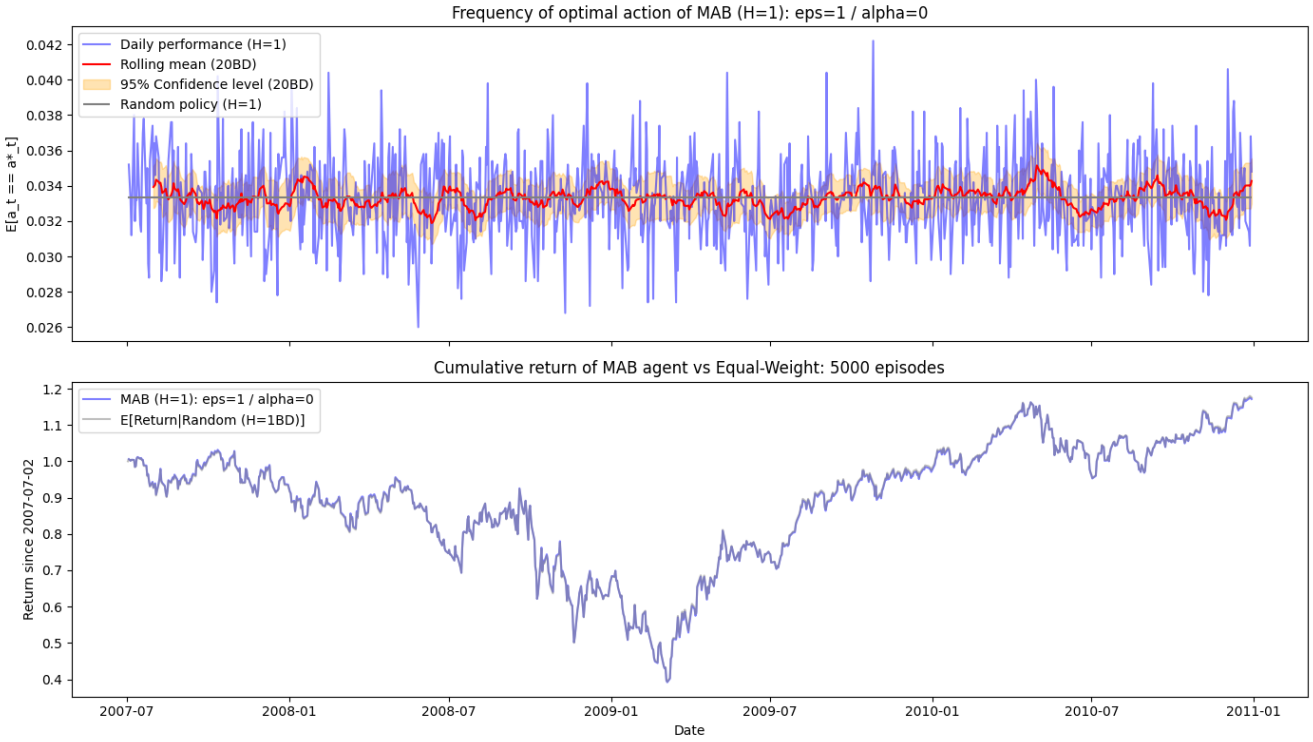


Figure 3: EPS=100% vs EquallyWeights over 5000 Episodes

Below is the result of 20% $\epsilon$-greedy exploration compared to the random policy agent over 100 episodes, with $\alpha = 0.14$ providing online-learning property to let any rewards past 20 business days contributes only 5% of the action value function:

The result shows that the performance may not necessarily higher or on par with the random policy, as the MAB agent's optimal action choice ratio does not show meaningful superiority over the pure random. This statement perhaps completely counters against what is given in the lecture ($\epsilon$-greedy exploration is better than random exploration with reasonable $\alpha$ in non-stationary dynamics).

Another experiment with the $\alpha$ to raised to 0.90 for more rapid decay of past rewards memory shows a worsen performance of the model:

Reducing the $\epsilon$ to 1% shows a significant drop in the ratio of optimal action choices, although the overall performance is just similarily worse than the random policy to that with high $\alpha$ agent:

## Multi-Armed Bandit with Upper-Confidence-Bound (UCB) value optimism-based exploration

The UCB provides optimistic action value for all actions where those with less visit during the episode is boosted higher to facilitate the exporation.

The algorithm used for this study is to utilise the concept of UCB as an uncertainty bonus to facilitate exploration for less visited actions via below algorithm [0]:

- For each episode:

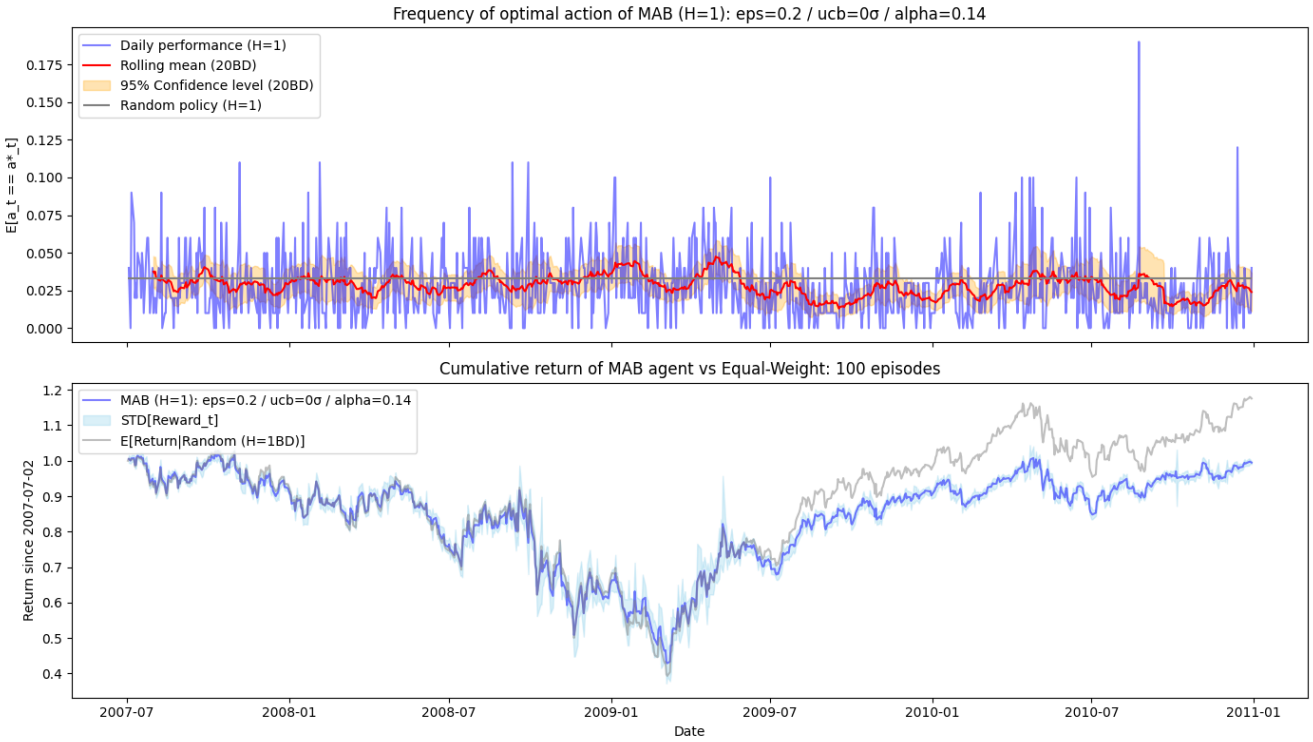


Figure 4: EPS=20%

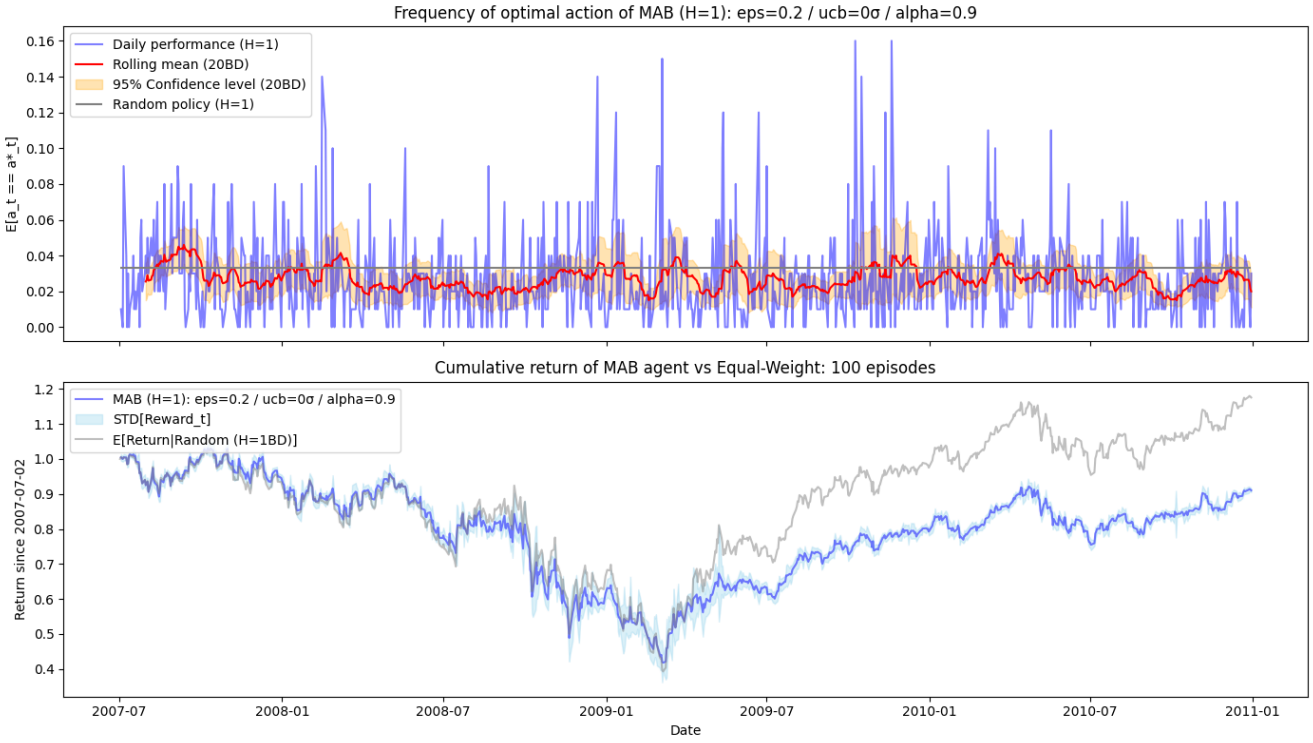


Figure 5: EPS=20% with Alpha=0.9

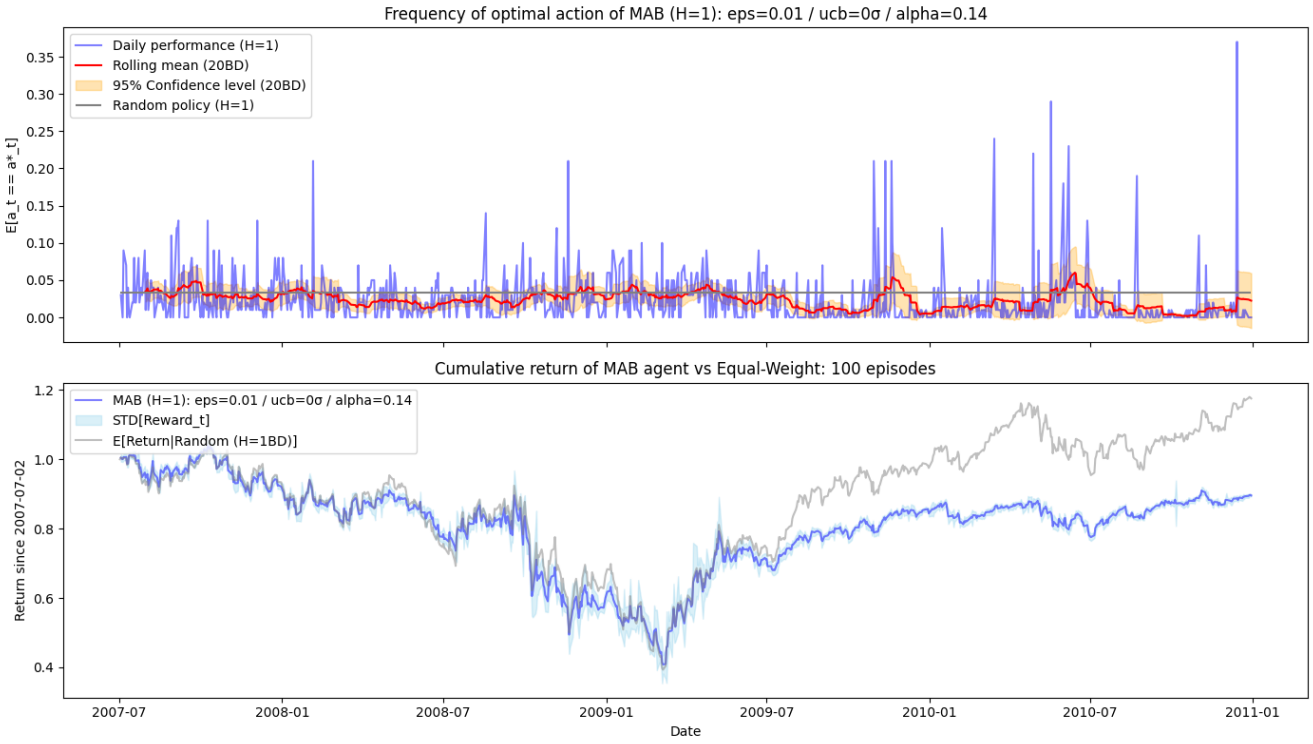


Figure 6: EPS=1%

- Initialise Q-value per action Q(a) as 0 for all actions
- For each timestep $t$:
  - Choose the action $a_t$ by following:
    - Calculate the Action value score $S_t(a) = Q(a) + c \cdot \sqrt{\frac{\ln(t)}{N(a)}}$ for $\forall a \in A$
    - Select the action from the softmax sampler where $P(a_t = i) = \frac{\exp(S_t(i)/T)}{\sum_{j \in |A|} \exp(S_t(j)/T)}$.
  - Observe the reward $r_t$ for given $a_t$ as forward-looking return $p_{t+H}/p_t - 1$
  - Update the Q-value at t accordingly: $Q(a_t) \leftarrow \alpha \cdot r_t + (1 - \alpha) \cdot Q(a_t)$

A reasonable level of $c$ is required to provide the uncertainty bonus at the fair level of Q-value, such that the agent can navigate both the exploration and exploitation. In this work, $c$ is constructed based on the standard-deviation of the rewards observed from the historical return data with a multiplier as a hyperparameter:

$$c = \sqrt{\text{Var}[r_{i \in [0 \ldots T]}]}/|A| > 0$$

, the division by the dimension of the action space $|A|$ helps to normalize the exploration bonus to prevent excessive bonuses given to the less frequently selected actions due to the high dimensionality of the action space. For instance, when action space is $|A| = 30$ dimension and the standard deviation of the reward is 1, then one visit to the action in every 30 timesteps will provide the action one standard deviation of the reward as a bonus - lesser the visitation occurs the bigger the score diffuses by square-root of time.

Another aspect of the exploration rate control is the temperature hyperparamter $T > 0$. Higher the temperature given, more uniform the $P(a)$ distributes over the action space as the exponents become more even in absolute level which encourages more occurence of randomised choice of action.

This algorithm is a different interpretation of the UCB optimism from what was instructed in the WQU lecture [1],

as the uncertainty bonus is *not* given to the Q-values but added to a delegate score that only plays a role with the soft-max sampler. Hence the Q-values are not biased upward but retains its asymptotic convergent property over many iterations.

Below is the result of UCB exploration based agent with $T = 1$. The result shows the clear improvement in performance over the $\epsilon$-greedy policy and it is on par with the random policy. Level of randomness rules the performance where the level of exploration aligns both strategies in the same line. In order to validate this marginal improvement, further experiments are needed to compare the long-term performance of the UCB agent against other strategies.

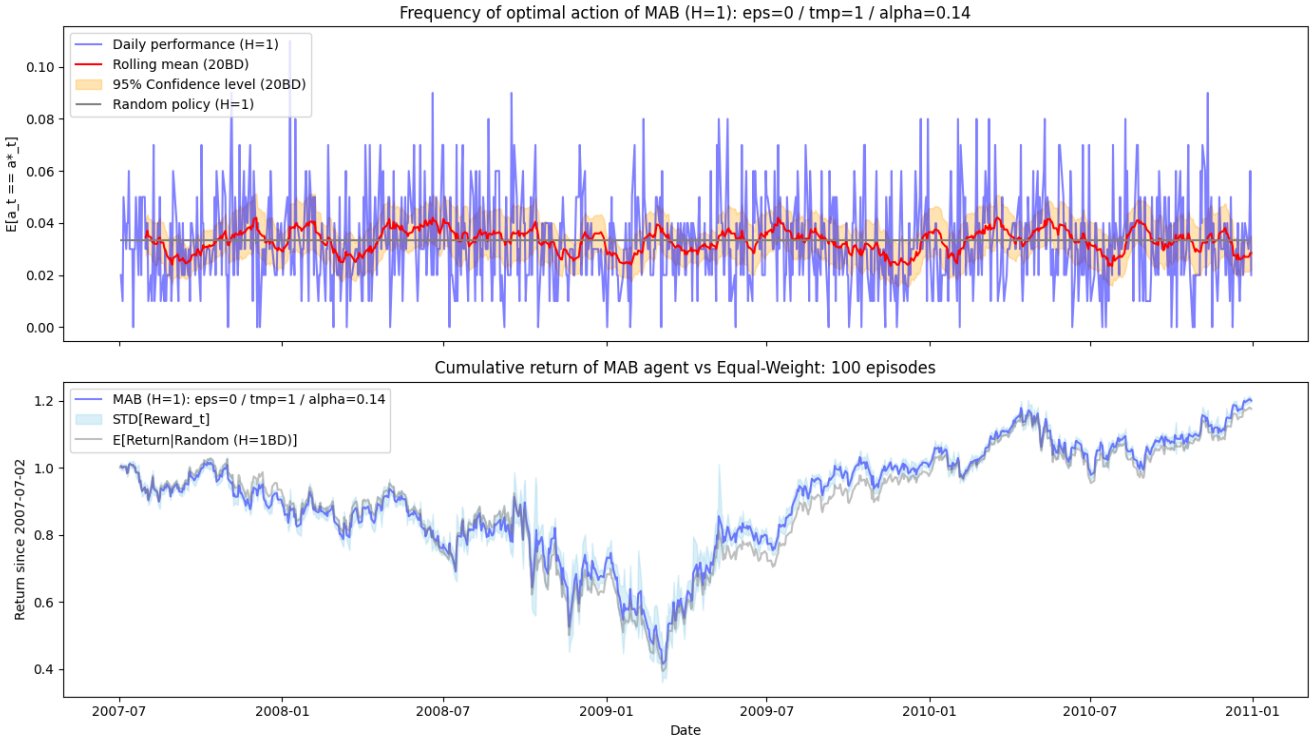


Figure 7: UCB with T=1

Rather more likely, we suspect this performance superiority may be purely due to the fact that the exploration rate of the above UCB agent is much higher than that of the $\epsilon$-greedy agent we studied above so that the performance aligns mostly with the random policy. In order to validate this, one should conduct a series of ablation studies by systematically reducing the exploration rate of the UCB agent and observing the impact on its performance. To prove this postulation, we run the same simulation with lower temperatures to discourage exploration. As shown in below, the simulation results show that the lower exploration ratio indeed tends to deteriorate the expected performance of the agent:

At $T = 0.01$ the odds of choosing non-greedy action was 15%, which is almost the same as $\epsilon$-greedy policy with $\epsilon = 0.85$:

At $T = 0.001$ the odds of choosing non-greedy action was 77%, which is almost the same as $\epsilon$-greedy policy with $\epsilon = 0.2$:

The result resembles with that of $\epsilon$-greedy policy MAB agent as that higher exploration rate facilitates the higher performance but only upto what random policy achieves. For reference, the first experiment with $T = 1$ had an exploration rate of 97%, which means the strategy is almost the same as the pure exploration policy.

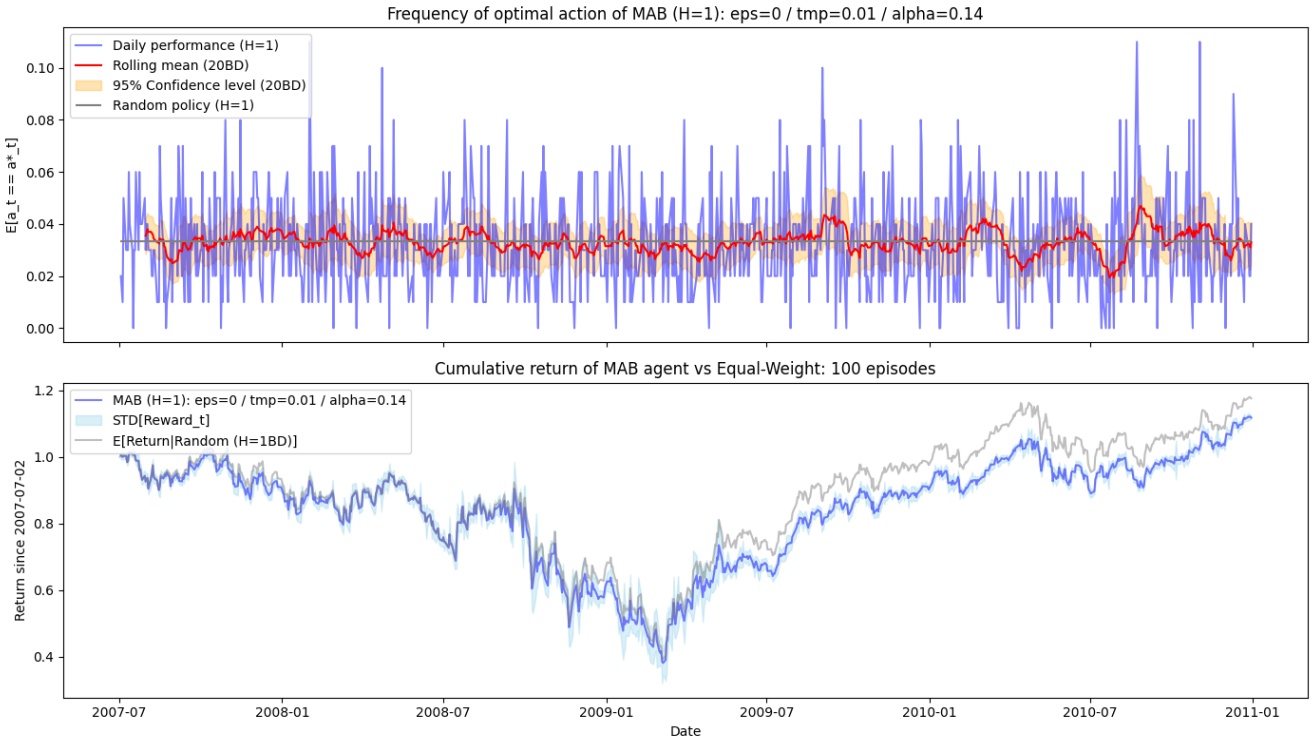


Figure 8: UCB with T=0.01

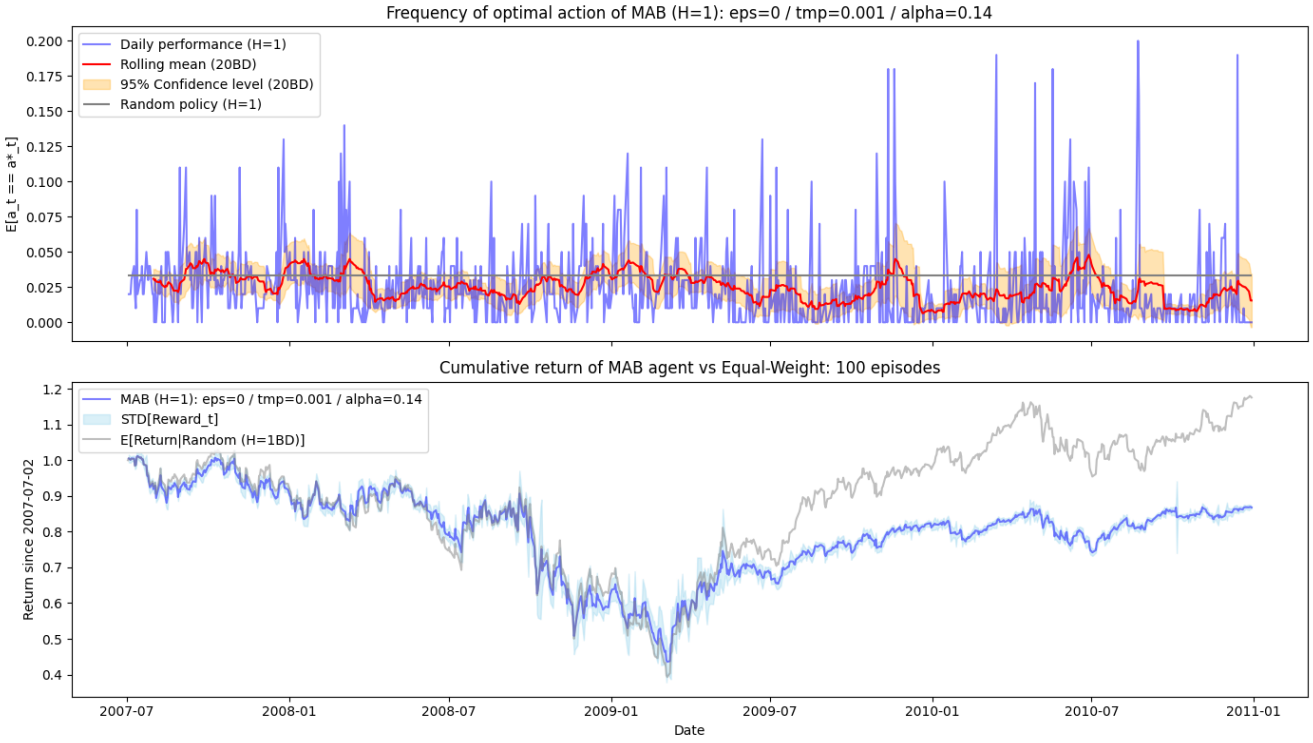


Figure 9: UCB with T=0.001

## Summary of stock-picking challenge simulation with context-free Multi-Armed Bandit agent

The performance of the $\epsilon$-greedy MAB agent is highly randomised so that good learning in this setup provides the optimal action by maximum on par with the random agent. However, the true trading performance typically is recorded worse with lower level of exploration $\epsilon$ and higher learning rate $\alpha$. This conclusion differs from the optimistic result shown in the WQU demonstration [1]. Nonetheless this quite aligns with what's expected: market is does not allow more than what the strategy knows about the future, and any agent without solid contextual memory on the past movements and market information can never beat against the random.

The same pattern is observed with the UCB-optimism stochastic bandit agent, where the high level of temperature $T$ boosts the randomness (=explorativeness) of the agent resulting in higher cumulative porfolio return converging to the random policy's performance expectation.

Observation of this pattern from both MAB strategies suggests that the optimal agent at the *current* setup evolves the agent to be on par with the random (or equally-weighting) agent by best. The limitation of the *current* agent is that nothing about the market state (or at least in form of market context) is given to the agent such that it has no oracle to align its action values with expected market dynamics. Therefore, the natural venue to advance with the quality of behavior is to incorporate the market context as the policy state to evaluate more accurate Q-value as a practice called contextual Bandit algorithm [4]. Indeed, the better algorithm must be the (reduced) Monte-Carlo methods such that the action-state value function is formulated with the market context and its deicision is evaluated and feeds back into the agent such that the market trend becomes a learnable trait for the agent.

## Context-free Multi-Armed Bandit agent performance on recent market

In this section, we apply the same context-free MAB algorithms for the same stock-picking challenge but on the same stocks but over a more recent time period - year 2020. This allows us to evaluate trend dependency of the agents since the style of market changed since the Lehman crisis.

As shown below, the market seems to be much more correlated in 2020 compared to the landscape in the Lehman crisis' by sector. Nonetheless, the patterns of granularity largely resemble with what was observed in the the Lehman crisis' such as commercial banks, consumer sector. One big change must be the split of the energy sector into two: CMS-DUK versus XOM versus APA where the upstream energy developers have developed more independent movement compared to the companies staying on traditional resources and infrastructure, and downstream/full-chain players (such as XOM) indeed align more with oil-based chain as a whole (such as Ford and General Electric).

### Search of performing context-free agent by a variety of hyperparameters

UCB with T = 1, 0.1, 0.01, 0.001 result in 4%, 4.7%, 14%, 65% exploitation of the optimal action value that correspond to $\epsilon$ = 99%, 95%, 86%, 35%:

With the year 2020 data, the UCB agent shows that the more explorativeness does not always lead to better performance in limited amount of simulation as high $T$ does not lead to better performance. Nonetheless, the perforance ceiling to the equal-weight porfolio still persists.

On the other hand, $EPS$ = 20%-greedy algorithm performs on par with the equal-weight porfolio for this date range:

Raising $\alpha$ from 0.14 to 0.56 and 0.9 results in deterioration in performance, although the relationship with the cumulative portfolio return is not linear:

One interesting observation is that the $\alpha = 0.005$ provides the superior performance to the equal-weight portfolio which rebuts the performance ceiling argument made in the last section:

The stock basket in year 2020 may have experienced long-standing preference on certain sectors or stocks, leading to the observed performance improvements with the low learning rate - or low *forget* rate of past experiences on

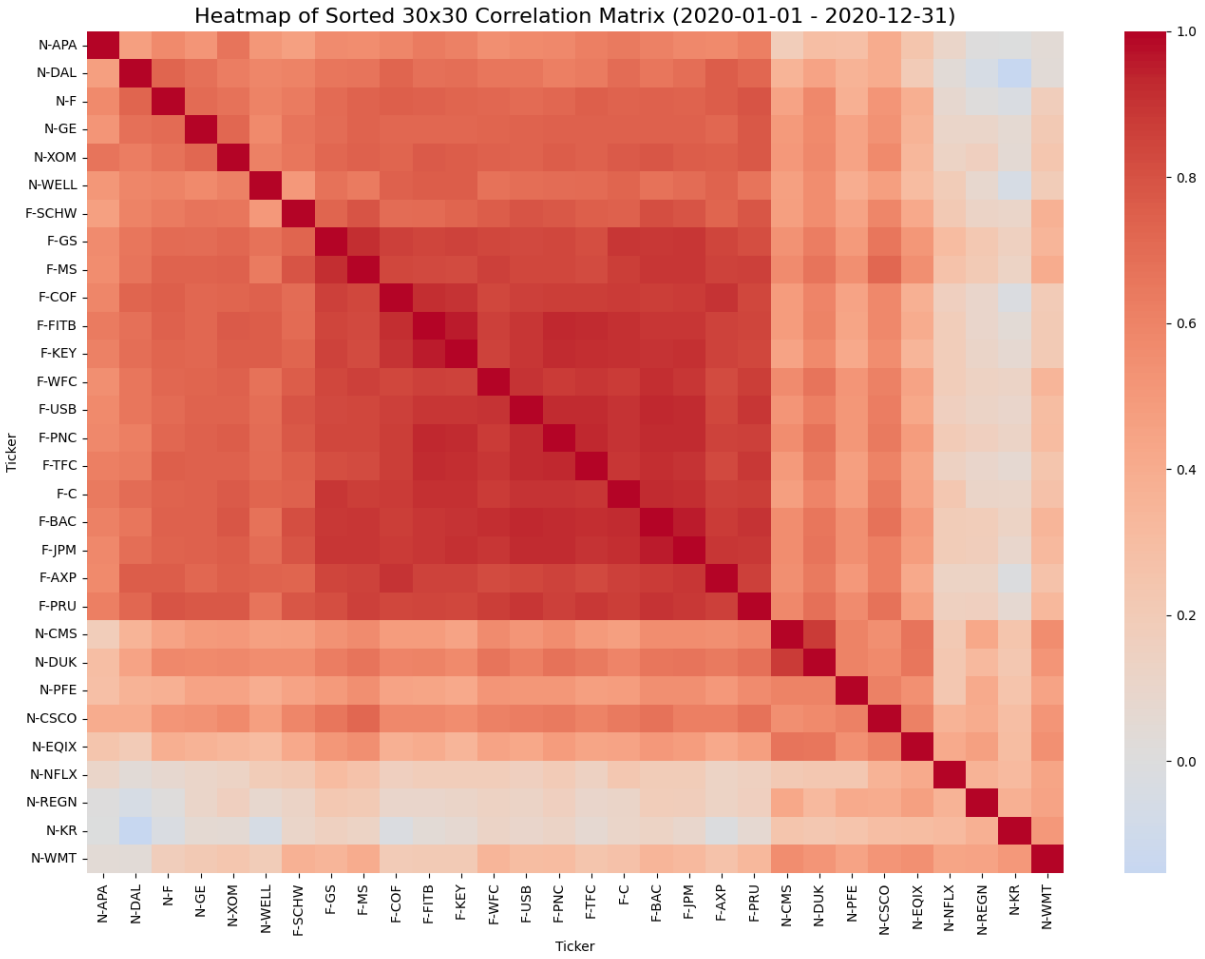


Figure 10: Sorted Pearson Correlation heatmap on Y2020

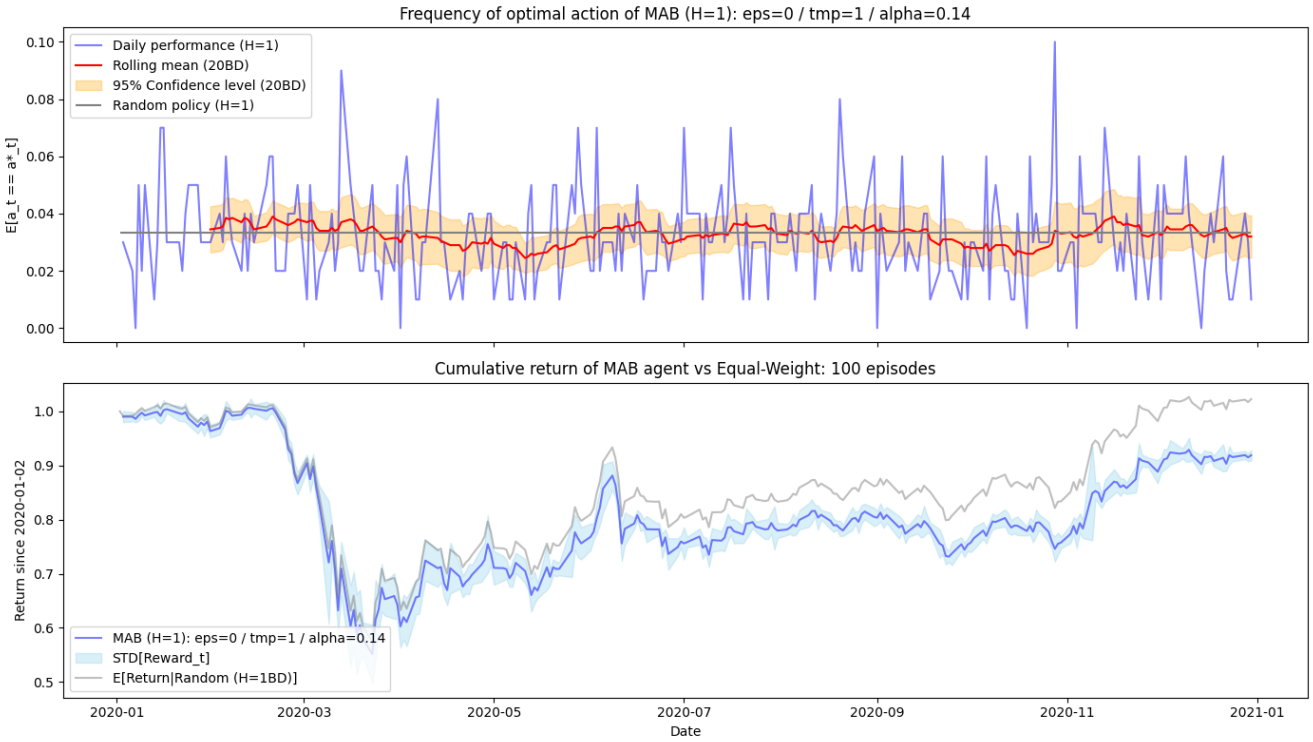


Figure 11: UCB with T=1

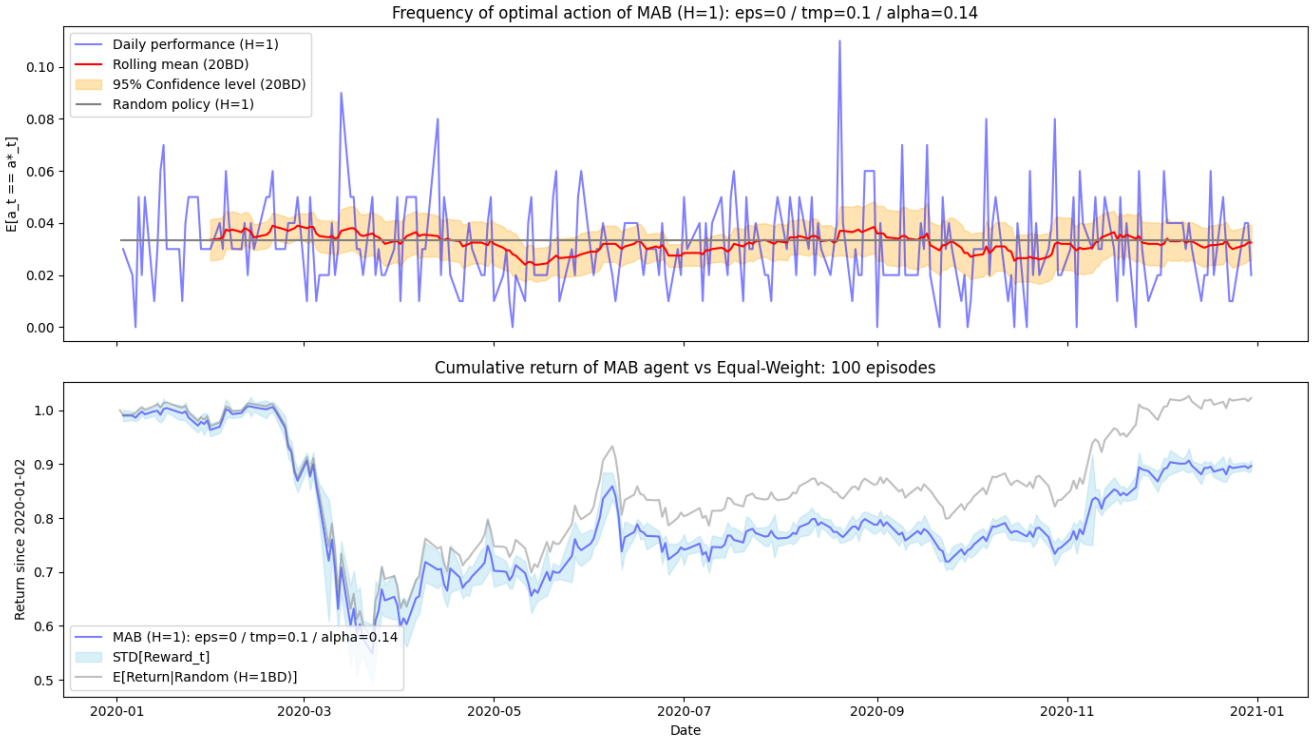


Figure 12: UCB with T=0.1

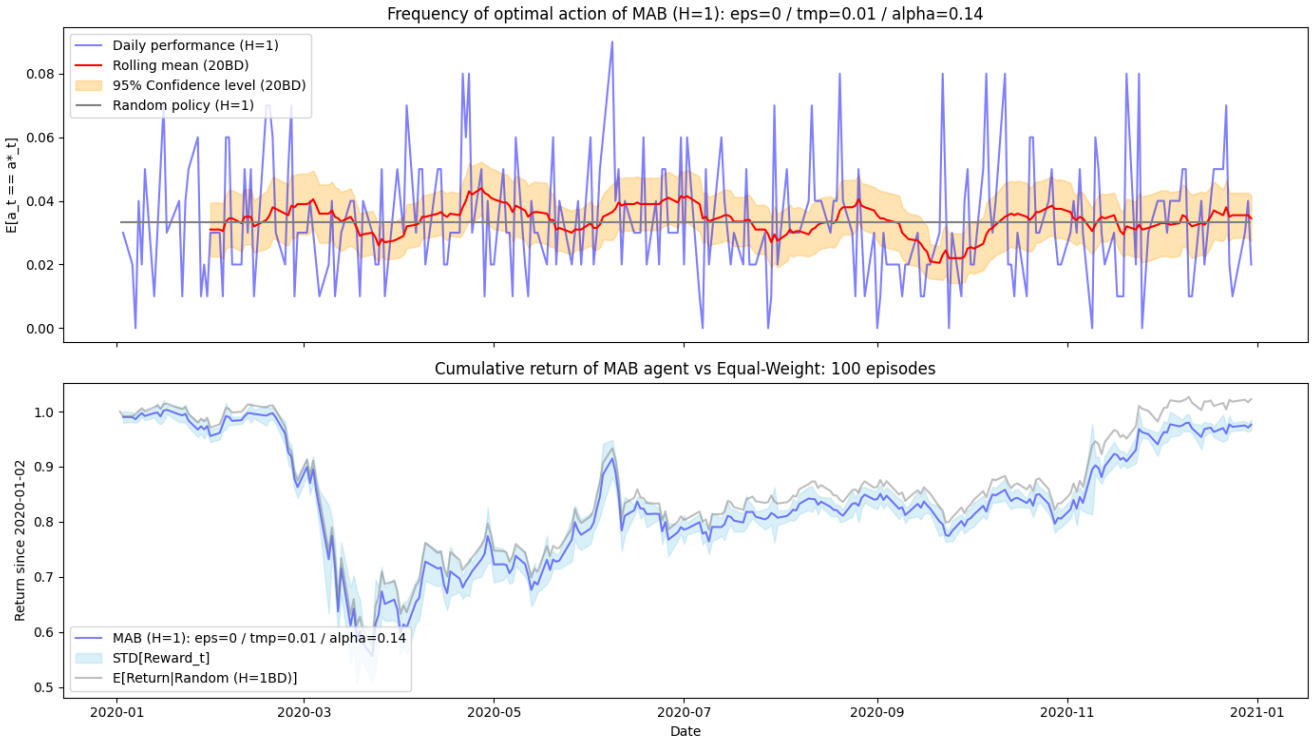


Figure 13: UCB with T=0.01

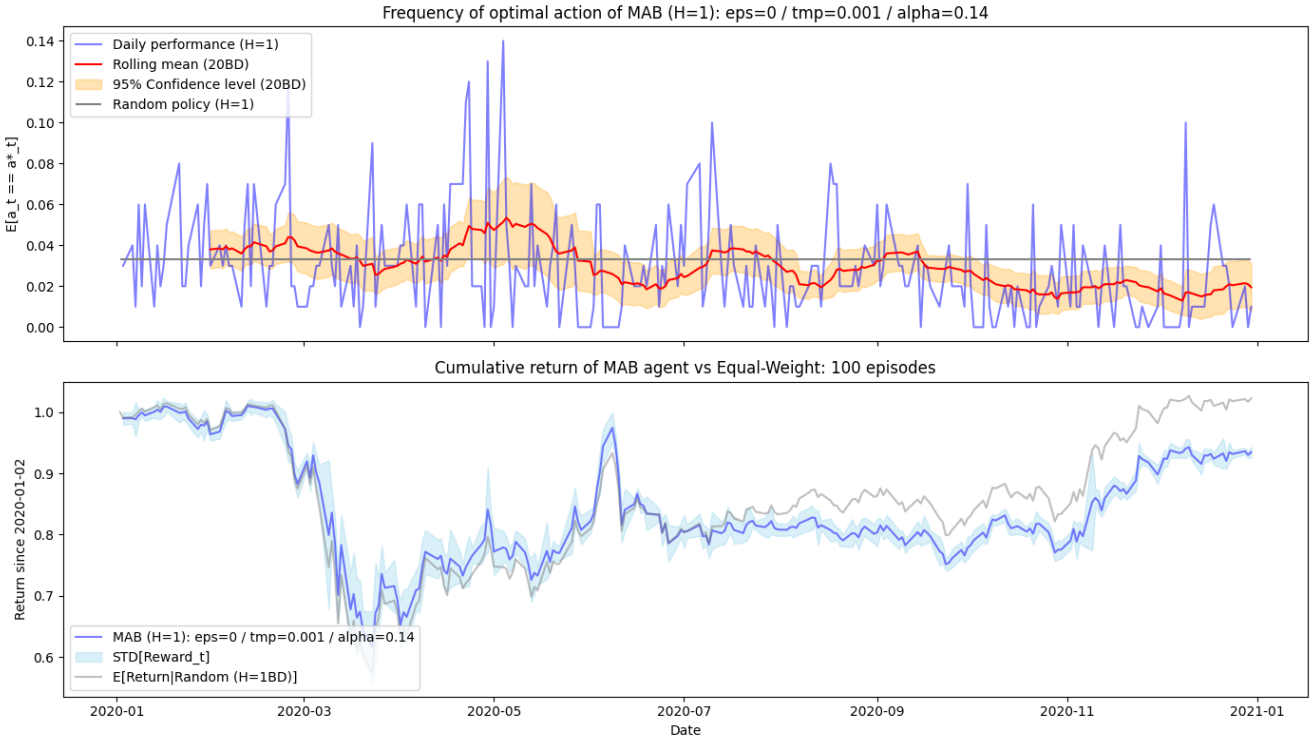


Figure 14: UCB with T=0.001

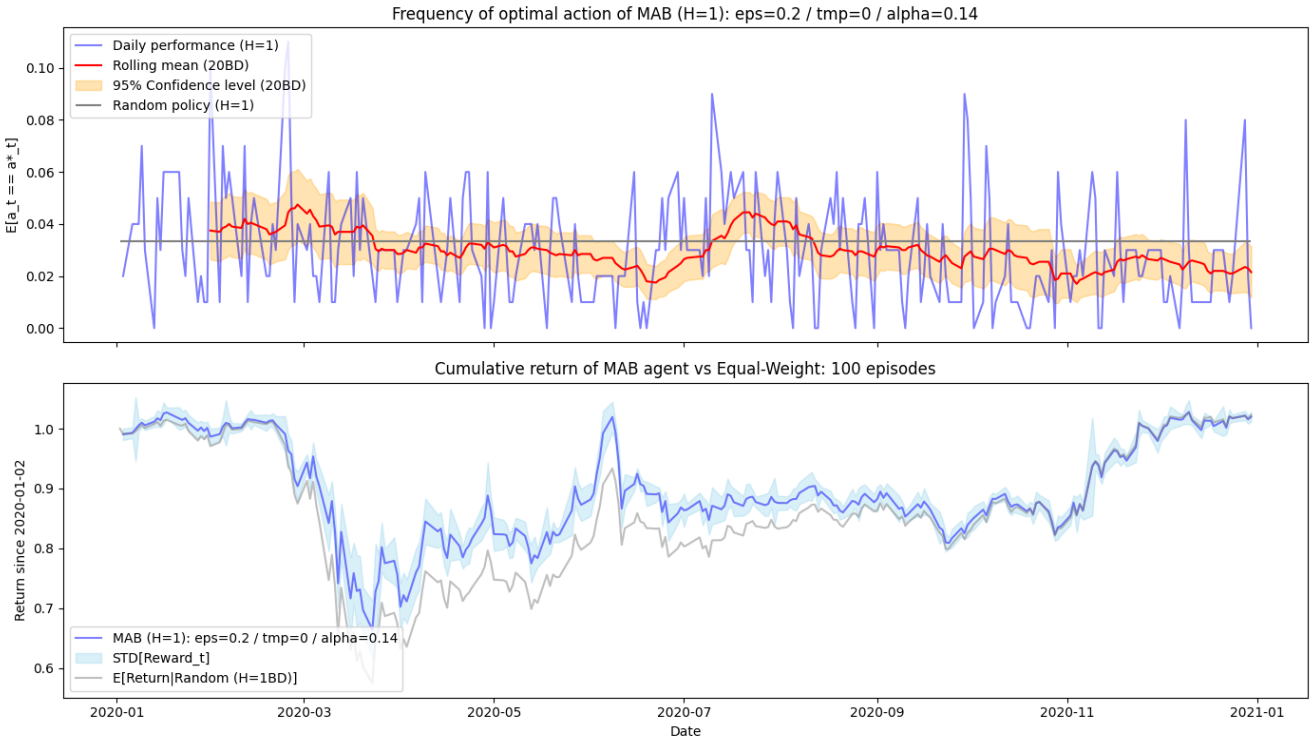


Figure 15: EPS=20% with ALPHA=0.14

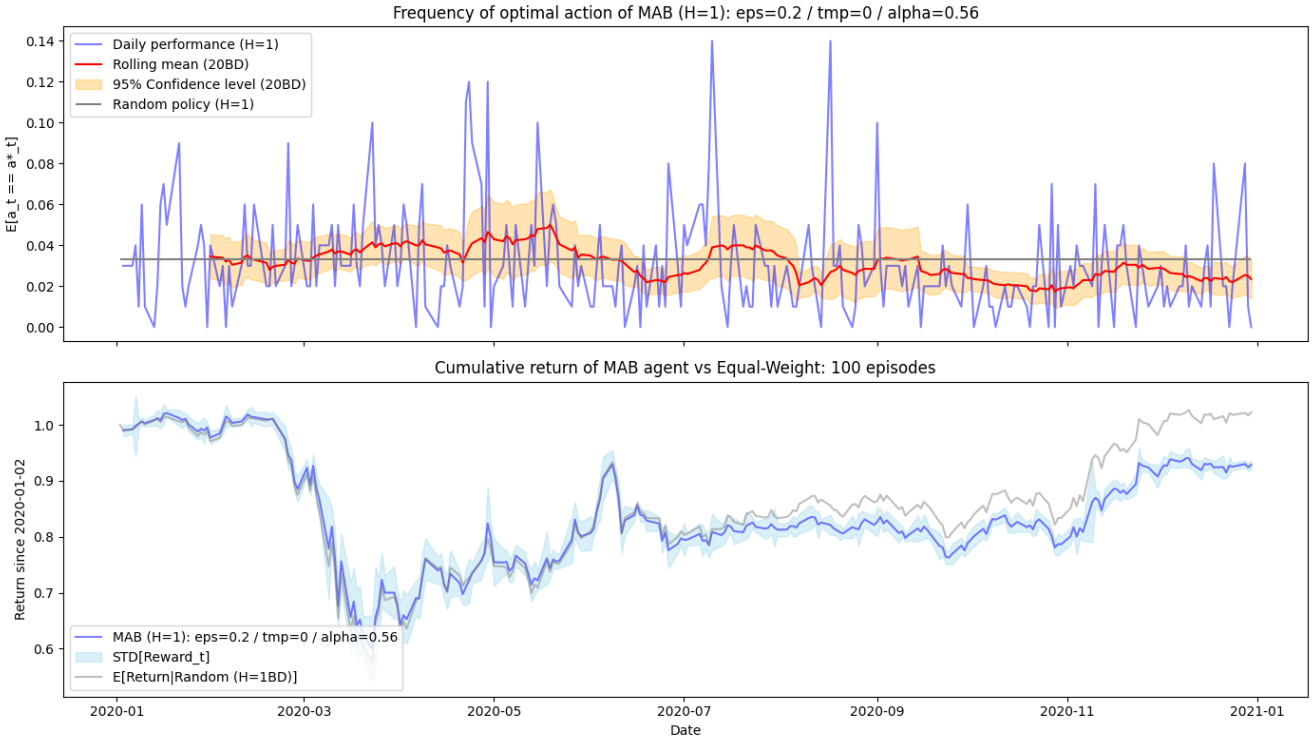


Figure 16: EPS=20% with ALPHA=0.56

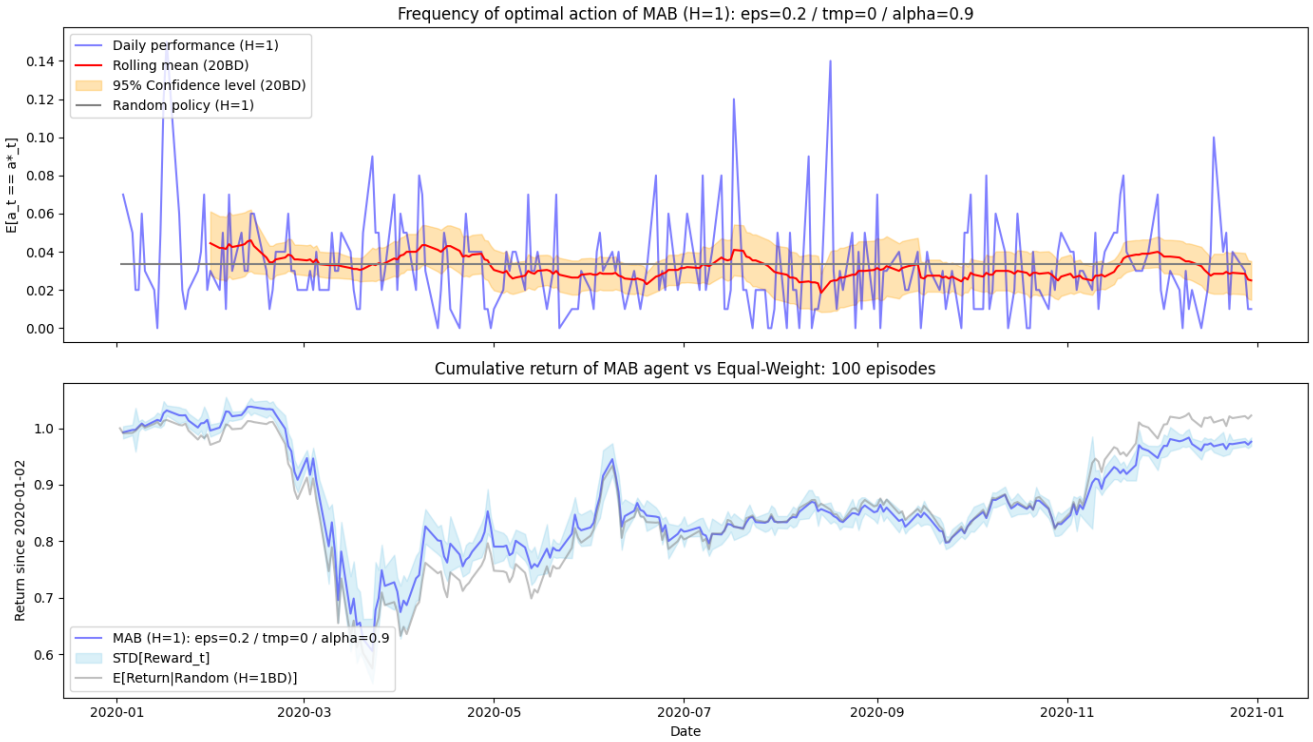


Figure 17: EPS=20% with ALPHA=0.9

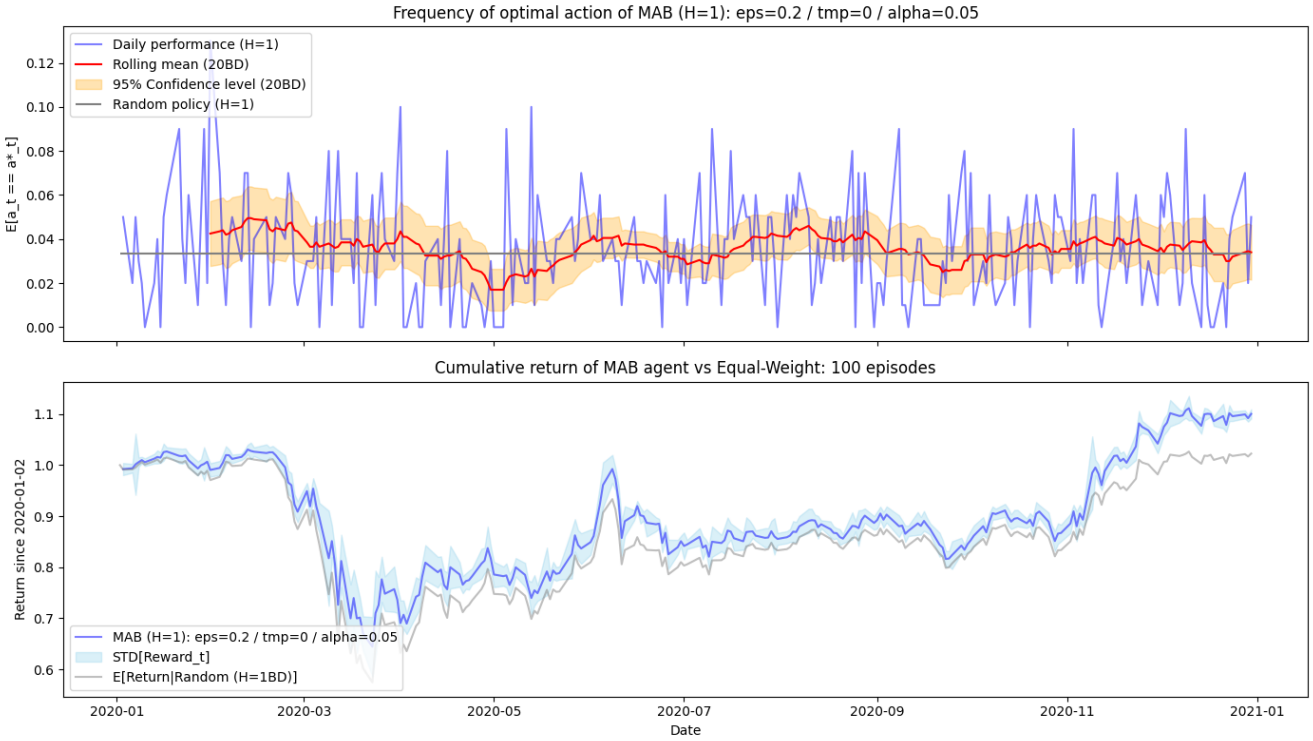


Figure 18: EPS=20% with ALPHA=0.05

action values. Although the number of episodes is not sufficient enough, the results suggest that the optimal hyperparameter is not linearly *(nor convexly) searchable and varies by the time due to the dynamic characteristics of the market.

### Comparison with Randomly Selected Best Agent

Here we discuss how the market foresight leads the portfolio performance with perfect oracle. On top of the equal-weight portfolio, we replace its stock-pick decision with the best-performing symbol of the day for randomly chosen dates. Below is the simulation of our best context-free MAB agent ($\epsilon$=20% with $\alpha$=0.14), equal-weight portfolio, and the equal-weight portfolio with best pick for 3.3% on the date range:

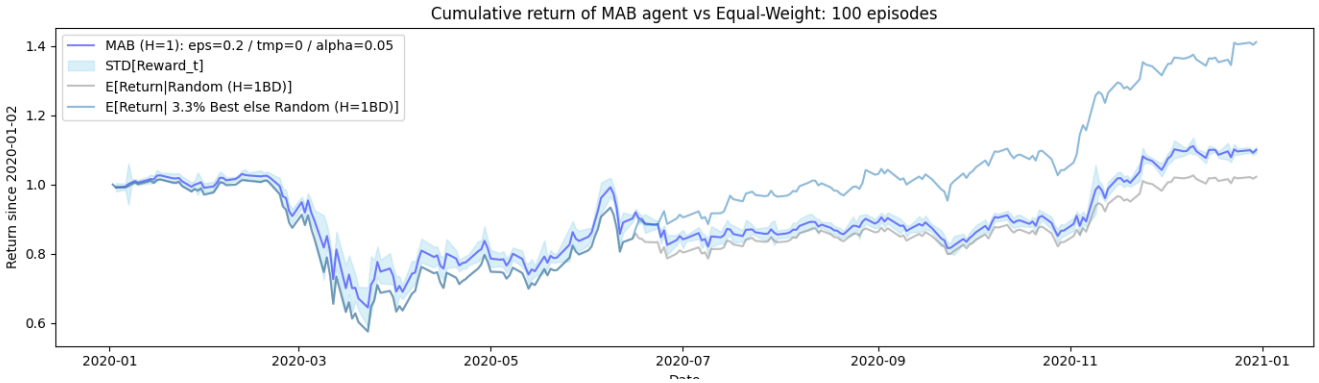


Figure 19: EPS=20% with ALPHA=0.05 MAB versus Randomly Best Equally-weighted policy

By raising a few percent of the stock-pick accuracy from the random policy - 3.3% to 6.6% -, the overall portfolio performance improves significantly, demonstrating the importance of market foresight in stock selection. This re-affirms the potential benefits of introducing market contextual state into the action value function that has meaningful forecast accuracy that can hit once every 16 tryouts. Machine-learning based state-action value functions for the contextual bandit agent will be the natural advance in this regards, but it is beyond the realm of this study.

## Relevant work

Huo et al. [3] propose a risk-aware multi-armed bandit approach that elaborate the above studies MAB strategy by below aspects. Below describes their algorithm mark 1.

First, they undertook asset filtering using a technique called “minimum spanning tree” based on correlation amongst the stock groups. This is done in an anticipation that it will provide better diversification and reduced systemic risk during market crises.

Second, their MAB agent negotiates over the one stock-to-pick action space by balancing the expected return at each decision timestep with the UCB optimism. In addition, the agent builds a global portfolio vector (amongst the selected stocks) minimises the conditional value-at-risk (CVaR). Both the one-hot-vector from the UCB policy and the global portfolio vector constitute the true action vector to act on the market, and the agent is rewarded with the realized return of the portfolio. The algorithm is claimed to be convex which means the optimal solution is always computationally reachable.

The primary criticism on this paper must be that the experimentation was based on simulated price paths made by multi-variate Geometric Brownian Motion (GBM) instead of the real market data for training the agent. The drift is known for the GBM hence is completely learnable without knowing the market context as their GBM does not even change their drift factor over time. This greatly favours with the MAB, but the assumption is never realistic and we saw the limitations of the MAB algorithms in previous sections.

What may be useful is the idea of decomposing the stock baskets into bigger granules. As quoted in the paper, the conventional approach is to conduct factor analysis such as Principal component analysis to extract factors to represent orthogonal baskets of assets to trade which effectively reduces the control dimension as well as risk dimension to monitor.

Whilst the decomposition method stated in this paper looks to be a special type of spectral analysis, the paper does not clarify how its decomposition maps the basket of 15 or 30 stock symbols into reduced space and how the action space is designed for the trading agent accordingly Specifically, we have conducted the hierarchical clustering that resembles with minimum spanning tree algorithm does in its core. However, parsing the stocks into layers of clusters do not sufficiently provide the true mapping of the stock space into the action space since the mapping requires the adequate mapping design such as eigen-space for the PCA to constitute principal components. If some stocks are excluded from the action space, then one should provide the criteria of doing so from the spectral analysis result.