Risk-Aware Reinforcement Learning for Multi-Period Portfolio Selection

David Winkel[✉][0000−0001−8829−0863], Niklas Strauß[0000−0002−8083−7323], Matthias Schubert[0000−0002−6566−6343], and Thomas Seidl[0000−0002−4861−1412]

LMU Munich, Germany
{winkel,strauss,schubert,seidl}@dbs.ifi.lmu.de

Abstract. The task of portfolio management is the selection of portfolio allocations for every single time step during an investment period while adjusting the risk-return profile of the portfolio to the investor's individual level of risk preference. In practice, it can be hard for an investor to quantify his individual risk preference. As an alternative, approximating the risk-return Pareto front allows for the comparison of different optimized portfolio allocations and hence for the selection of the most suitable risk level. Furthermore, an approximation of the Pareto front allows the analysis of the overall risk sensitivity of various investment policies. In this paper, we propose a deep reinforcement learning (RL) based approach, in which a single meta agent generates optimized portfolio allocation policies for any level of risk preference in a given interval. Our method is more efficient than previous approaches, as it only requires training of a single agent for the full approximate risk-return Pareto front. Additionally, it is more stable in training and only requires per time step *market* risk estimations *independent* of the policy. Such risk control per time step is a common regulatory requirement for e.g., insurance companies. We benchmark our meta agent against other state-of-the-art risk-aware RL methods using a realistic environment based on real-world Nasdaq-100 data. Our evaluation shows that the proposed meta agent outperforms various benchmark approaches by generating strategies with better risk-return profiles.

Keywords: Portfolio Selection · Multi-Objective Optimization · Deep RL

1 Introduction

The modern financial system offers investors the possibility to store wealth over long time horizons. Typically, wealth is accumulated in times of productivity and is then consumed in times of need. This can for example allow a private investor to retire or allow an institutional investor, such as an insurance company, to distribute funds to its clients at a later point in time. Thereby arises the fundamental question of how to manage the stored wealth while it is not needed for consumption. The task of portfolio management addresses this question and deals with the most suitable selection of assets out of a basket of available assets.

Besides the obvious goal of maximizing the expected economic return, often, the investor's capacity to bear risk, i.e., the uncertainty in his economic returns, has to additionally be taken into account. This bearing capacity of risk for an investor is summarized in his individual risk preference level. The individual risk preference level can depend on various factors, such as the investor's investment horizon, his return expectations as well as his individual risk appetite.

While there are various works on short-term *trading* such as [3,30], we focus on the long-term task of *portfolio selection* which brings multiple practical challenges for investors. According to requirements for many institutional investors from regulatory frameworks, such as Solvency II[1] , the risk in returns needs to be considered on a per time step basis. Professional investors are furthermore generally evaluated by their customers on their periodical performance, including the periodical risk taken on. The aforementioned individual risk preference of an investor can be difficult to quantify. In practice, the identification of a risk preference parameter is therefore often done by comparing alternative risk-return optimized allocation policies to one another and by then selecting the allocation policy fitting best. However, the identification of various optimized allocation policies on the Pareto front is computationally expensive, especially in multi-period settings which allow the investor to dynamically adjust his asset allocation during the trajectory. A typical example for the high computational demand is the extensive use of Monte Carlo simulations in the field of Asset Liability Management (ALM) applications, which can be seen as a specific type of the portfolio selection task, as discussed e.g., in [1].

In this paper, we frame the task of portfolio selection as a Markov decision process (MDP) which we set up to allow the modelling of a complex multi-period stochastic financial environment. To solve the MDP, we propose a risk-aware RL approach, which is able to control the risk in returns for each time step over the entire investment horizon. We choose to estimate the risk independently from the agent's current policy, making it only dependent on a market risk estimator as well as on the agent's current action. Contrary to policy dependent estimators in RL, such as critics, which can suffer from a moving target problem, our proposed risk estimator does not suffer from this issue, thereby allowing for sample efficiency and accelerated convergence. We propose a *meta agent* which uses the risk preference level as an inference parameter rather than as a hyperparameter. This allows for the agent to be trained over an interval of risk preference levels. In contrast, previous approaches have relied on training different agents for each level of risk preference, which has the drawback of requiring separate computationally expensive trainings, separate model networks and separate sets of hyperparameters. The usage of the risk preference level as an inference parameter further allows for approximating the Pareto front in a computationally efficient manner. One single trained meta agent is able to generate optimized asset allocation policies for any risk preference level within the specified interval during inference time. The implementation of our agent is based on PPO by [26], using a Dirichlet action distribution. In our experiments,

[1] `https://www.eiopa.europa.eu/browse/solvency-2_en`

our PPO based approach is able to outperform three alternative approaches: firstly, a simple Equal Weight Buy and Hold strategy; secondly, a DDPG based risk-aware RL approach by [1] and thirdly, a TD3 based risk-aware RL approach by [32].

We benchmark all approaches in two different settings: on previously not known data from the training environment and on a full year of unseen real world Nasdaq-100 data in a backtesting setting.

The main contributions of this paper are:

- A computationally efficient way to approximate the risk-return Pareto front for a continuous interval of risk preference levels by training only a single meta agent
- A method that allows estimating the risk of returns independently from the agent's current policy
- A PPO based approach with a Dirichlet action distribution suitable for the task of multi-period portfolio selection

2 Related Work

The related work to our approach can be categorized into four main areas: risk measures in risk-aware RL, portfolio optimization, RL applications to financial tasks, approximation of the Pareto front.

The related work on **risk measures in risk-aware RL** considers several different risk measures. Early works use the *standard deviation* as a measure of risk such as [29] who proposed a risk-adjusted objective function by subtracting the standard deviation from the cumulated discounted rewards. However, this formulation violates the temporal persistence property necessary to guarantee the convergence to an optimum for policy iteration algorithms. Alternative approaches such as [10] use the *conditional value-at-risk (CVaR)* as a risk measure, thereby addressing the risk of small probability events with high impact. Recent approaches have recognized the importance of measuring dispersion not solely in cumulated returns, i.e., over the entire trajectory, but of also addressing the variability in rewards per time step which can be highly relevant, e.g., for economic tasks such as trading or portfolio construction. A risk measure addressing this issue is the *reward volatility* defined by [6] which captures the variability of rewards between steps. [32] too proposed a framework optimizing the *variance of a per-step reward*. Another risk measure aiming to capture the variability per time step was published by [1] where it is defined as the *variance in rewards per time step observed in the current trajectory*. In contrast to the approaches mentioned above, we exploit the fact that in our setting, the risk of a step can be computed based solely on the current action and the market risk which is estimated independently from the policy. This in turn allows for the estimation of the risk in a very sample efficient way.

The foundations for **portfolio optimization** in financial literature were laid by the work of [16] who formulated the modern portfolio theory which is the basis of many works such as the one by [7]. They too used a mean-variance (MV)

optimization approach in order to find the optimal weightings of investments in a portfolio offering the best risk-return trade-off. Thereby, the risk is measured as the variance in economic returns for a single time step. A more recent approach by [11] introduces a regime-switching factor model which – while still in the MV setup – allows for a single period optimization under different market regimes. Such different market regimes correspond to different states of the market, e.g., optimistic and pessimistic market sentiments. Other works such as the one by [8] introduce a framework extending the MV single-period optimization to a MV multi-period optimization.

The area of **RL applications to financial tasks** has become more popular in recent years, as RL methods can naturally handle multi-period problems as well as different states, such as different market regimes, in the context of a MDP and are thus well suited to tackle the requirements of financial tasks. Many of the correspondingly published works, such as [3,30] focus on trading which is characterized by a rather short term view. Other authors use RL methods to find long term strategies to solve a portfolio selection task. [25] apply a policy iteration algorithm to the portfolio selection problem in combination with a risk-adjusted objective function. In order to model the actions of an investor in a MV setup, [4] use a policy gradient method and propose the usage of the Dirichlet distribution. [1] propose the usage of the DDPG algorithm to optimize the risk-reward trade-off faced in a portfolio selection task for a life insurance company. In contrast to the approaches mentioned above, our approach allows a single meta agent to be trained over a continuous interval of risk preference levels, instead of training different agents for each level of risk preference individually.

The **approximation of a Pareto front** in the context of a multi-objective optimization (MOO) is discussed by authors such as [18]. In contrast to our approach, they focus on a supervised MOO problem instead of a RL one. Other authors such as [22] propose the approximation of the Pareto front in a RL MOO setting. However, in their setting, they deal with a multi-objective Markov decision process (MOMDP) with multiple reward functions, while we formulate the task as an MDP using a scalarized objective function by linearly combining the economic return objective and the economic risk objective. Thus, our method computes all Pareto optimal solutions on the convex hull but neglects those being Pareto optimal for non-linear scalarization functions [24]. Though this restriction systematically reduces the number of found Pareto optimal policies, we argue that the approximation generated by our method yields a sufficiently large and intuitive set of user options.

3 Background

A discrete-time MDP is described by a five tuple $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$, consisting of the state space $\mathcal{S}$, the action space $\mathcal{A}$, a reward R which will be treated as a random variable as well as the transition probability function $P(s'|s,a) \in [0,1]$ for $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$ and a discount factor γ discounting future rewards.

The random variables for the next state S' and for the reward R are determined jointly and depend only on the preceding state s and action a. Their joint probability distribution is described by

$$p(s', r|s, a) = Pr(S' = s', R = r|S = s, A = a).$$

In the case that R is a continuous reward random variable, we obtain

$$\hat{R}(s, a) \coloneqq \mathbb{E}[R|S = s, A = a] = \int_r \int_{s'} r \; p(s', r|s, a) ds' dr.$$

A trajectory $\tau = (s_0, a_0, r_1, s_1, a_1, ...)$ is a sequence of states and actions. Let

$$P(\tau|\pi) = \mu_0(s_0) \prod_{t=0}^{T-1} p(s_{t+1}, r_{t+1}|s_t, a_t)\pi(a_t|s_t)$$

represent the probability of observing the trajectory τ given policy π. The term $\mu_0(s_0)$ describes the probability of observing s_0 as the initial state, i.e., $s_0 \sim \mu(\cdot)$.

We define the return as the observed discounted cumulative rewards for the trajectory τ, i.e.,

$$G(\tau) \coloneqq \sum_{t=0}^{T-1} \gamma^t r_{t+1}$$

where r_{t+1} are the observed rewards from time step t, given s_t, a_t and s_{t+1}.

The objective function is then defined as the expected return for a given policy π and thus

$$J(\pi) \coloneqq \underset{\tau \sim P(\tau|\pi)}{\mathbb{E}}(G) = \int_\tau P(\tau|\pi)G(\tau)d\tau.$$

4 Risk-aware Portfolio Optimization

We consider an agent (i.e., investor) with a fixed investment horizon T who wants to allocate his wealth into different assets in order to maximize the trade-off between the expected return and the individual preference for risk for the periods $t = 0, \ldots, T$. The investable asset universe contains N assets. The **discount factor** is set to $\gamma = 1$.

We define the **state space** of the MDP as $\mathcal{S} = \mathcal{T} \times \mathcal{W} \times \mathcal{V} \times \mathcal{U}$. Here, the space $\mathcal{T} \subseteq \mathbb{R}$ is populated by the parameter λ which is used to represent the agent's individual risk preference level. In contrast to other approaches [32,1], we thus use λ as an inference parameter, rather than as a hyperparameter. This parameter is crucial in enabling the agent to learn an interval of different risk preference levels by being randomly sampled at the beginning of each trajectory during training and then remaining constant until the end of the trajectory. $\mathcal{W} \subseteq \mathbb{R}_0^+$ represents the current absolute wealth level of the agent while

the standard-simplex $\mathcal{V} = \left\{ v \in \mathbb{R}^N : \sum_{i=0}^{N-1} v_i = 1, v_i \geq 0 \text{ for } i = 0, \ldots, N-1 \right\}$ represents the current relative portfolio allocation. $\mathcal{U} \subseteq \mathbb{R}^N$ represents all the observed single asset returns from the previous time step.

The **action space** $\mathcal{A}$ is also defined as a standard-simplex to represent the weighting vector chosen by the agent as action $a_t = [a_{t,0}, \ldots, a_{t,N-1}] \in \mathcal{A}$ at time step t. The choice of the action space $\mathcal{A}$ as a standard-simplex represents the need of the agent to allocate all available funds into its portfolio within each period, i.e., $a_t^\intercal \mathbb{1} = 1$, whereby short-selling of assets is not permitted, i.e., $a_i \geq 0 \; \forall i$.

The random vector $\Theta = [\Theta_0, \ldots, \Theta_{N-1}] \in \mathcal{U}$ models the economic return of each asset individually for each time step. The portfolio return is a random variable with an expected value denoted as

$$\mathbb{E}\left[\Theta_{PF}\right] = \mathbb{E}\left[a^\intercal \Theta\right] = a^\intercal \mathbb{E}\left[\Theta\right].$$

Changes in the portfolio weightings a_t in period t by the agent cause transaction costs, defined by

$$tc_t = \left(|a_t - v_t|\right)^\intercal c$$

where the vector $c = [c_0, .., c_{N-1}]$ contains the asset-specific transaction costs caused by a trade of the respective asset. Note that the transaction costs are non-stochastic and fully determined by action a_t.

We then define the observed economic reward r as a combination of the transaction costs tc and a realization ϑ_{PF} of the random variable of the portfolio's economic return Θ_{PF}, i.e.,

$$r = \vartheta_{PF} - tc. \tag{1}$$

To include the element of risk awareness in the **reward** of the MDP, we shape the reward to include the economic reward as well as a risk measure weighted by a penalty term:

$$r'(s, a) \coloneqq r(s, a) - \lambda f_{risk, \Theta_{PF}}(s, a).$$

The term λ is the risk penalty factor which reflects the agent's individual preference to take on risk. Note that the risk in the reward, i.e., $f_{risk, \Theta_{PF}}(s, a)$, is measured per time step, cannot be observed directly and therefore has to be estimated.

Subsequently, the risk-aware return is defined as:

$$G'(\tau) \coloneqq \sum_{t=0}^{T-1} \gamma^t \Big(r_{t+1} - \lambda \hat{f}_{risk, \Theta_{PF}}(s_t, a_t) \Big)$$

where $\hat{f}_{risk, \Theta_{PF}}(s_t, a_t)$ is an estimated function to measure the risk in r_{t+1} and only depends on the state-action pair of time step t. With our approach, $\hat{f}_{risk, \Theta_{PF}}$ can therefore be estimated over different trajectories regardless of the agent's current policy.

4.1 Risk measure

Based on the financial setting, we use the standard deviation as a risk measure. This risk measure is widely accepted in finance, as e.g., discussed by [13]. Thus, our approach requires estimating the risk per time step, i.e., the standard deviation in returns associated with each state-action pair. In our setting and in line with other authors such as [8], the returns of financial assets are assumed to be independent between time steps. The only source of stochasticity in the estimator for the portfolio's risk is the market risk of the individual assets, while the action is a deterministic component of the estimator function.

The variance of the economic portfolio return is defined as:

$$Var(\Theta_{PF}) = a^\intercal \Sigma_\Theta a$$

where Σ_Θ is the covariance matrix for asset-wise economic returns Θ and a describes the weightings in the individual assets – which in our setting is the action selected by the agent. Note that the standard deviation is a risk measure free from assumptions about the underlying distribution. The $N \times N$ covariance matrix Σ_Θ can be rewritten in terms of the first and second moment of Θ:

$$\Sigma_\Theta = \mathbb{E}[\Theta\Theta^\intercal] - \mathbb{E}[\Theta]\mathbb{E}[\Theta]^\intercal \ .$$

The covariance matrix can be estimated independently from both the agent's action as well as from his current policy and solely depends on the state of the market environment from which the estimator receives the latest observable information $u \in \mathcal{U}$ which is included in $s \in \mathcal{S}$, and thus

$$\hat{f}_{Cov}(s) \coloneqq \widehat{\Sigma_\Theta} \ .$$

Including action a in our estimator function, the estimator for the risk of the portfolio return in a single time step is defined as

$$\hat{f}_{risk,\Theta_{PF}}(s,a) \coloneqq \sqrt{a^\intercal \widehat{\Sigma_\Theta} a} = \sqrt{a^\intercal \hat{f}_{Cov}(s) a} \ .$$

We use two neural networks, $\hat{M}_1$ and $\hat{M}_2$, to estimate the first and second moment of Θ. Due to our multivariate setting with N individual assets, $\hat{M}_1$ has to estimate N values. For the second moment, $\hat{M}_2$ has to estimate the unique elements present in the symmetric matrix, i.e., $(N+1) \cdot N \cdot 0.5$ elements. These moment estimators are trained simultaneously with the agent's policy.

4.2 Policy

As a policy function for our PPO based implementation, we use the Dirichlet distribution as proposed in a similar context by [4]. The Dirichlet distribution is

a multivariate probability distribution governed by the concentration parameter vector $\alpha = [\alpha_0, \ldots, \alpha_{N-1}]$ where $\alpha_i > 0$ with $i = 0, \ldots, N-1$. Its probability density function for a random vector is defined as

$$f(x; \alpha) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=0}^{N-1} x_i^{\alpha_i - 1}$$

where $B(\alpha)$ is the multivariate beta function. A sample $x = [x_0, \ldots, x_{N-1}]$ drawn from a Dirichlet distribution satisfies the properties $\sum_{i=0}^{N-1} x_i = 1$ and $x_i > 0$, and is thus a member of a standard simplex fulfilling the requirements imposed on actions in the context of portfolio selection. In the experimental part, we further examine for comparison purposes a DDPG based as well as a TD3 based implementation of our method. For both implementations, the natural way of enforcing the sampled outputs to be members of a standard simplex is by applying a softmax function in the output layer. The exploration is done by adding the explorational noise to the parameters in the hidden layers of the policy network, which is an approach described by [23].

4.3 Algorithm

The algorithm for the PPO based implementation can be found in Algorithm 1. Note that in our setting the ability of the meta agent to learn asset allocation policies for any level of risk preference on a continuous interval is enabled through (a) the formulation of a policy independent risk measure and (b) the treatment of the risk preference parameter λ as an inference parameter by inclusion in the state $s \in \mathcal{S}$. During training, at the beginning of each trajectory, the risk preference parameter λ is sampled from a continuous uniform distribution. Within each trajectory i the initially sampled λ_i remains constant.

4.4 Network Architectures

For our PPO based framework, we have four different models: an actor network $\pi(a|s, \theta)$, a critic network $v(s)$ and two moment estimating neural networks $\hat{M}_1$ and $\hat{M}_2$, responsible for the estimation of the first and second moments of the individual assets to form an estimated covariance matrix. The architecture of the actor network and the critic network share the same body network of four fully connected hidden layers of size 512, 256, 128 and 64 with ReLU activation functions. These layers are followed by an attention based GTrXL architecture by [21] allowing for also handling tasks requiring memory. The use of a GTrXL element instead of the standard transformer architecture improves the architecture's optimization properties in RL settings significantly. The GTrXL element consists of a single transformer unit with one encoder layer as well as one decoder layer with four attention heads and an embedding size of 64. The network's body is then split into two heads, in which the actor network's output layer utilizes an exponential activation function. This enforces the output to be in the value range of $\mathbb{R}^+$, to meet the requirements of the parameter input of the Dirichlet

Algorithm 1 Risk controlling PPO

Input: environment ϵ

1: **init** parameters: θ_0,ϕ_0, γ_0, δ_0 # policy, value function, 1st & 2nd moment estimate
2: **for** $k = 0, 1, \ldots$ **do**
3: sample trajectories $\mathcal{D}_k = \{\tau_i\}$ with policy $\pi_k = \pi(\theta_k)$ in ϵ for T time steps; at each trajectory start sample risk preference $\lambda_i \sim U(a, b)$.
4: Update risk estimator function $\hat{f}_k(\cdot,\cdot) = \sqrt{\hat{M}_{2,\delta_k}(\cdot,\cdot) - \left(\hat{M}_{1,\gamma_k}(\cdot,\cdot)\right)^2}$.
5: Calculate the est. risk $\hat{f}_k(s_t, a_t)$ and then the risk adjusted reward r'_{t+1}.
6: Calculate advantage estimates, $\hat{A}_t$ based on the current value function V_{ϕ_k}.
7: Update policy by maximizing the PPO-Clip objective:

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

8: $\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - r'_{t+1}\right)^2$. # update ϕ
9: $\gamma_{k+1} = \arg\min_{\gamma} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(\hat{M}_{1,\gamma_k}(s_t, a_t) - r_{t+1}\right)^2$. # update γ
10: $\delta_{k+1} = \arg\min_{\delta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(\hat{M}_{2,\delta_k}(s_t, a_t) - r_{t+1}^2\right)^2$. # update δ
11: **end for**

distribution. The head of the critic network on the other hand is a basic linear layer without activation function. We further need to estimate the covariance matrix in order to estimate the risk associated with an action by estimating the elements of the multivariate expressions of the first and second moment, i.e. $\mathbb{E}[\Theta]$ and $\mathbb{E}[\Theta\Theta^\intercal]$. Since this is a standard supervised learning problem, we apply a standard transformer architecture to estimate a multivariate time series as described by [31]. In our setting, this architecture consists of four encoder layers and four decoder layers, each utilizing eight attention heads with an embedding size of 512. Note that the actor and critic network are trained together, having a joint loss function using the Adam optimizer with a learning rate of $5.0 \cdot 10^{-5}$. The moment estimating networks use a separate loss function and utilize the Adam optimizer with a learning rate of $1.0 \cdot 10^{-3}$.

5 Experiments

5.1 Environment

We use the qlib package[2] to fetch and process real-world financial data for the US market contained in the Nasdaq-100. The Nasdaq-100 is a modified market value-weighted index containing the shares from the 100 largest non-financial companies traded on the Nasdaq stock exchange. Over time, the composition of the index changes. This is due to the (de)listing of shares and changes in the market value of companies, which can then – according to the guidelines of the Nasdaq-100 – lead to removal from or addition to the Nasdaq-100. We consider the monthly single share closing prices for the period from January 1, 2010 to December 31, 2020. In order to avoid having to deal with missing data, we filter out the companies that were not included in the Nasdaq-100 throughout the entire period. From the remaining 35 companies, we randomly choose 16 to represent the investable universe in the RL environment.

In literature, there is a multitude of approaches modelling the dynamics in the time series of financial returns. One such approach is the application of classical time series models, e.g., by [5,17]. Another approach is the usage of deep learning based methods, e.g., by [15,20]. Furthermore, hidden markov models (HMMs) are applied, e.g., by [14,19]. In our setup, we decide to model the market dynamics by applying a HMM. However, any method capable of modelling the dynamics in a time series of financial returns could be used interchangeably.

To choose the HMM fitting best, we follow [19] and use two criteria, namely the Akaike information criterion (AIC) by [2] and the Bayesian information criterion (BIC) by [27]. Both criteria suggest the use of a two state HMM. In our environment, we set the length of a trajectory to twelve time steps, reflecting the investment horizon of a year. The transaction costs are set to 0.2% of the traded volume.

5.2 Experimental Setup

The implementation of our approach is based on the RLlib framework[3] and the agents were trained on a cluster utilizing various types of commercially available single GPUs. For each evaluation step, we sample 1000 trajectories to calculate the corresponding statistics.

For the implementation of our benchmark RL algorithms, we base [32] on the publicly available GitHub code[4] while for the approach proposed by [1], we rebuild the architecture as described in their paper.

5.3 Evaluation

Benchmarking with other approaches. For our evaluation setup, we compare our approach with three alternative approaches. The first one is an Equal

[2] `https://github.com/microsoft/qlib/tree/main`
[3] `https://docs.ray.io/en/master/rllib/index.html`
[4] `https://github.com/ShangtongZhang/DeepRL`

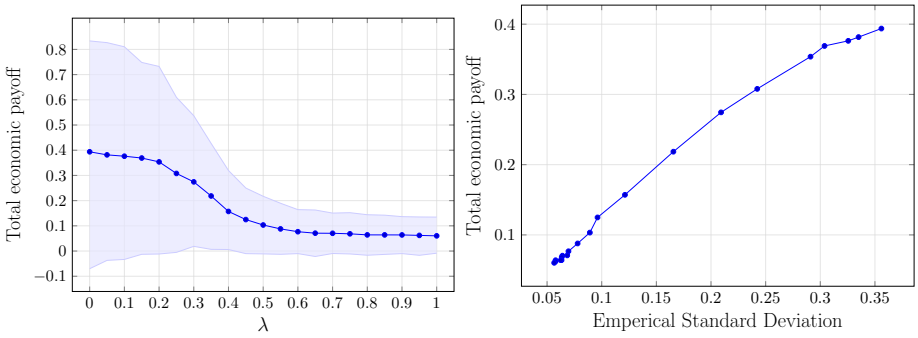
Table 1: Evaluation results of 1000 trajectories from the environment (I) and backtesting on the Nasdaq-100 data trajectory of 2021 (II).

	Sharpe Ratio (ex-post)	Total Econ. Payoff	Est. Std. Dev.
(I) Environment			
Equal Weight B&H	1.232	0.218	0.177
Ours	**1.347**	**0.240**	0.178
Zhang et al. (2021)	1.283	0.229	0.178
Abrate et al. (2021)	1.158	0.209	0.180
(II) Backtesting			
Equal Weight B&H	1.968	0.336	0.171
Ours	**2.039**	**0.344**	0.168
Zhang et al. (2021)	1.921	0.335	0.174
Abrate et al. (2021)	1.908	0.331	0.173

Weight Buy and Hold (Equal Weight B&H) policy, which is a simple investment heuristic. At the beginning of the investment horizon, the funds are distributed equally to all available assets. After buying the assets they are held until the end of the investment horizon without any allocation adjustments. Despite its low complexity, an Equal Weight policy is considered to be a performant allocation policy. The second approach is a risk-aware RL DDPG based method described by [1] which in their paper is specifically applied to the task of generating an optimized asset allocation policy for a single level of risk preference. In the following, we will refer to their approach as Abrate et al. (2021). The third approach is MVPI-TD3 by [32]. It is a state-of-the-art risk-aware RL method based on the TD3 algorithm, originally introduced by [12]. In the following, we will refer to the third approach as Zhang et al. (2021).

We evaluate two different settings: in setting (I) we evaluate the policies' performances for 1000 unseen trajectories generated by the environment. Setting (II) follows a backtesting approach by evaluating the policies' performances for the unseen historical trajectory of the Nasdaq-100 data for the entire year of 2021. To allow for a consistent comparison of the asset allocation policies, every approach needs to be adjusted to bear a comparable amount of risk. All of our evaluated RL approaches are able to control the risk of the optimized asset allocation policy by adjusting their specific risk preference level parameter λ. In contrast, the Equal Weight B&H approach does not have this feature, resulting in the use of the risk level of the Equal Weight B&H approach as the baseline level of risk to which the other approaches have to adapt. Accordingly, the λ in the other approaches are set in such a way to generate strategies with a standard deviation in returns comparable to the one produced by the Equal Weight B&H approach.

For (I), the policies' standard deviations in returns over the entire trajectory are estimated as the empirical standard deviations. To estimate the standard



(a) Approx. Pareto front of strategies' mean total economic payoff and their respective 90% confidence interval in relation to the risk preference level λ.

(b) Approx. Pareto front of strategies' mean total economic payoff in relation to their empirical standard deviation.

Fig. 1: Evaluation of a single PPO based meta agent for different levels of risk preference.

deviations in returns in (II), a different approach is needed, since the real-world data offer only a single observation per month, which makes it difficult to estimate the monthly variances in returns. To address this issue, we use the daily observations within a month. After estimating the daily variance, this value is scaled up by the number of trading days within the month in order to estimate the assets' monthly variance – a method commonly used in finance [9]. The root of the sum of the monthly variances is then used to obtain an estimate for the policies' standard deviations in returns in the backtesting evaluation setting.

Table 1 provides the evaluation results of our experiments. We evaluate the approaches in regards to their ex-post Sharpe ratio, an evaluation metric commonly used in finance to compare investment performances [8,28]. In addition, Table 1 shows the individual components of the Sharpe ratio, which in our setting are the total economic payoff and the estimated standard deviation. In both evaluation settings (I) and (II), our approach is able to provide the asset allocation policy scoring the highest Sharpe ratio and – under an approximately equal level of risk – therefore also the highest total economic payoff.

Note that the risk preference parameters λ of the different risk-aware RL approaches cannot be compared directly, due to different definitions of risk and different objective functions. For Zhang et al. (2021) we use a risk preference parameter value of 0.55, for Abrate et al. (2021) a risk preference parameter value of 0.3 and for our own approach a risk preference parameter value of 0.34.

Approximation of the Pareto front. A multi-period asset allocation policy for a given level of risk preference incorporates a suggested asset allocation for each single time step. Our meta agent approach generates an entire set of asset allocation policies, whereby each single one is linked to a specific level of risk preference within a continuous interval. Figure 1a shows the performance

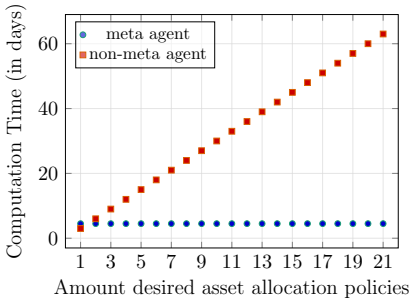


Fig. 2: Computation time required for training.

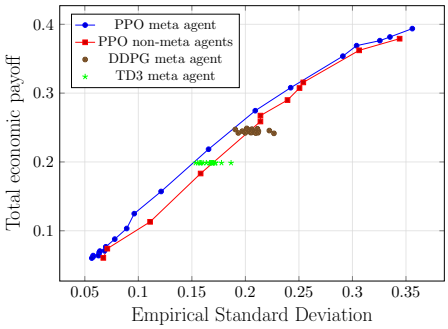


Fig. 3: Comparison of meta agent approaches and PPO non-meta agents.

of our approach with respect to different levels of risk preference λ. Each point represents an entire asset allocation policy evaluated over twelve time steps of the trajectory. The y-axis shows the economic return including the transaction costs, as defined in equation 1, cumulated over the entire trajectory. In the following, this value will be referred to as the *total economic payoff*. In Figure 1a we are evaluating 21 different asset allocation strategies with corresponding risk preference levels in the interval of 0.0 to 1.0 in steps of 0.05 generated by the same meta agent. The figure shows that our method is capable of approximating a monotonic decreasing Pareto front with increasing levels of risk preference. In order to illustrate the measured uncertainty of the total economic payoff, Figure 1a also includes the empirical 90% confidence interval. As in Figure 1a, in Figure 1b the 21 asset allocation strategies, evaluated in relation to their empirical standard deviation, form a Pareto front.

Stability during training. In order to find a suitable asset allocation policy, the RL based approaches use their model specific risk preference parameter λ. To allow for a consistent comparison of the approaches, each approach needs to generate a policy with a comparable level of risk, i.e, a comparable level of standard deviation in returns measured over the trajectory. From this arises the need to identify for each approach the corresponding individual risk preference parameter λ which produces a certain level of standard deviation. For the non-meta agent approaches by Zhang et al. (2021) and Abrate et al. (2021), the identification of a suitable risk preference parameter is done manually via an iterative interval search. Thereby, single agents need to be trained and evaluated. Both the DDPG and TD3 based approaches require a considerable amount of hyperparameter tuning for each single agent. When a suitable set of hyperparameters is found, it is then often not transferable between agents with different levels of risk preference. This leads to unstable training results combined with repeatedly extensive hyperparameter tuning.

To further investigate the stability properties of the DDPG and TD3 algorithm in our setting, we also implement our meta agent for the DDPG and the

TD3 algorithm. This allows for a direct comparison of all three implementations trained with the *same* objective function with the *same* definition of risk. During evaluation, neither the DDPG nor the TD3 implementation of a meta agent are able to generate a meaningful Pareto front. Their proposed asset allocation policies are strictly dominated by the asset allocation policies generated by the PPO implementation as shown in Figure 3. We also apply the PPO based approach to a non-meta agent, i.e., to the optimization of a single level of risk preference solely. We emphasize that for this PPO based approach, we are able to use a single set of hyperparameters for training, thereby transferable between agents for different levels of risk preference. Figure 3 further shows the comparison to the PPO meta agent. Due to computational limitations, we only train and evaluate 11 optimized allocation policies with PPO non-meta agents. Nevertheless, it can be seen that for the PPO based methods, both the meta agent as well as the non-meta agents are able to approximate a Pareto front, with the non-meta agents performing slightly worse. We hypothesize that the superior stability in hyperparameters for a PPO based approach over the DDPG and the TD3 based approaches plays an important role when successfully training a meta agent.

Efficiency. One advantage of our method when approximating the Pareto front is its computational efficiency. Once the meta agent has been trained, we are able to generate any number of optimized asset allocation strategies by simply changing the risk preference levels as an inference parameter. Thereby, the respective asset allocation strategies can be evaluated without further training. In contrast, previous approaches would need to train a different agent for each level of risk preference. Figure 2 shows the training time required to generate different optimized asset allocation strategies on the machine used for our experiments. While the time required to train a single agent for an optimized asset allocation using one single level of risk preference takes roughly 3 days, the training of a meta agent for an interval of levels of risk preference takes roughly 4.5 days on a system with an NVIDIA RTX 8000. When training multiple agents, the cumulative computation time increases linearly with the amount of desired optimized asset allocation strategies. In contrast, the training time of our approach stays constant due to the need of only training a single meta agent to cover an entire interval of risk preference levels.

Performance of Risk Measure Estimation. With our approach, we further introduce a method to estimate the risk per time step, which can be done independently from the agent's current policy. The experiments show fast convergence for both the first and the second moment estimators after roughly 6% of the total training time, i.e., after 150 out of a total of 2500 training iterations.

6 Conclusion

In this paper, we train an agent to invest a given amount of wealth into a set of assets on a monthly basis. In order to control the risk of the investment, the agent receives a risk preference parameter constraining the standard deviation in the financial returns received per time step. This in turn also indirectly controls the

risk of the financial returns over the entire trajectory. Our method of estimating the risk in a time step is independent of the agent's current policy and only requires the agent's current action as well as an estimate of the market risk. In our approach a single meta agent is trained for any risk preference level within a continuous interval, enabling a computationally efficient approximation of the Pareto front. We evaluate our PPO based approach combined with a Dirichlet action distribution against other state-of-the-art risk-aware RL approaches in a setting based on real-world Nasdaq-100 data. The results show that our new method outperforms compared approaches w.r.t. stability during training as well as generating asset allocation policies with better risk-return profiles. For future work, we want to explore the setting of multiple competing meta agents able to influence the market prices and their resulting interactions.

7 Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibility for its content.

References

1. Abrate, C., Angius, A., Francisci Morales, G.D., Cozzini, S., Iadanza, F., Puma, L.L., Pavanelli, S., Perotti, A., Pignataro, S., Ronchiadin, S.: Continuous-action reinforcement learning for portfolio allocation of a life insurance company. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 237–252. Springer (2021)
2. Akaike, H.: A new look at the statistical model identification. IEEE transactions on automatic control **19**(6), 716–723 (1974)
3. Almahdi, S., Yang, S.Y.: An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. Expert Systems with Applications **87**, 267–279 (2017)
4. André, E., Coqueret, G.: Dirichlet policies for reinforced factor portfolios. arXiv preprint arXiv:2011.05381 (2020)
5. Ariyo, A.A., Adewumi, A.O., Ayo, C.K.: Stock price prediction using the arima model. In: 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. pp. 106–112. IEEE (2014)
6. Bisi, L., Sabbioni, L., Vittori, E., Papini, M., Restelli, M.: Risk-Averse Trust Region Optimization for Reward-Volatility Reduction. In: Twenty-Ninth International Joint Conference on Artificial Intelligence Special Track. pp. 4583–4589. International Joint Conferences on Artificial Intelligence Organization (2020)
7. Black, F., Litterman, R.: Global portfolio optimization. Financial analysts journal **48**(5), 28–43 (1992)
8. Boyd, S., Busseti, E., Diamond, S., Kahn, R.N., Koh, K., Nystrup, P., Speth, J.: Multi-period trading via convex optimization. Foundations and Trends in Optimization **3**(1), 1–76 (2017)
9. Brigham, E.F., Ehrhardt, M.C.: Financial management: Theory & practice. Cengage Learning (2019)

10. Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M.: Risk-constrained reinforcement learning with percentile risk criteria. The Journal of Machine Learning Research **18**(1), 6070–6120 (2017)
11. Costa, G., Kwon, R.: A regime-switching factor model for mean-variance optimization. Journal of Risk (2020)
12. Fujimoto, S., Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: International conference on machine learning. pp. 1587–1596. PMLR (2018)
13. Guercio, D.D., Reuter, J.: Mutual fund performance and the incentive to generate alpha. The Journal of Finance **69**(4), 1673–1704 (2014)
14. Hassan, M.R., Nath, B.: Stock market forecasting using hidden markov model: a new approach. In: 5th International Conference on Intelligent Systems Design and Applications (ISDA'05). pp. 192–196. IEEE (2005)
15. Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., Soman, K.: Nse stock market prediction using deep-learning models. Procedia computer science **132**, 1351–1362 (2018)
16. Markowitz, H.: Portfolio selection. The journal of finance **7**(1), 77–91 (1952)
17. Munim, Z.H., Shakil, M.H., Alon, I.: Next-day bitcoin price forecast. Journal of Risk and Financial Management **12**(2) (2019)
18. Navon, A., Shamsian, A., Fetaya, E., Chechik, G.: Learning the pareto front with hypernetworks. In: International Conference on Learning Representations (2021)
19. Nguyen, N.: Hidden markov model for stock trading. International Journal of Financial Studies **6**(2), 36 (2018)
20. Pang, X., Zhou, Y., Wang, P., Lin, W., Chang, V.: An innovative neural network approach for stock market prediction. The Journal of Supercomputing **76**(3), 2098–2118 (2020)
21. Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R.L., Clark, A., Noury, S., et al.: Stabilizing transformers for reinforcement learning. In: International Conference on Machine Learning. pp. 7487–7498. PMLR (2020)
22. Pirotta, M., Parisi, S., Restelli, M.: Multi-objective reinforcement learning with continuous pareto frontier approximation. In: Twenty-ninth AAAI conference on artificial intelligence (2015)
23. Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R.Y., Chen, X., Asfour, T., Abbeel, P., Andrychowicz, M.: Parameter space noise for exploration. arXiv preprint arXiv:1706.01905 (2017)
24. Roijers, D.M., Vamplew, P., Whiteson, S., Dazeley, R.: A survey of multi-objective sequential decision-making. Journal of Artificial Intelligence Research **48**, 67–113 (2013)
25. Sato, M., Kobayashi, S.: Variance-penalized reinforcement learning for risk-averse asset allocation. In: International Conference on Intelligent Data Engineering and Automated Learning. pp. 244–249. Springer (2000)
26. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
27. Schwarz, G.: Estimating the dimension of a model. The annals of statistics pp. 461–464 (1978)
28. Sharpe, W.F.: The sharpe ratio. Streetwise–the Best of the Journal of Portfolio Management pp. 169–185 (1998)
29. Sobel, M.J.: The variance of discounted markov decision processes. Journal of Applied Probability pp. 794–802 (1982)

30. Wang, H., Zhou, X.Y.: Continuous-time mean–variance portfolio selection: A reinforcement learning framework. Mathematical Finance **30**(4), 1273–1308 (2020)
31. Wu, N., Green, B., Ben, X., O'Banion, S.: Deep transformer models for time series forecasting: The influenza prevalence case. arXiv preprint arXiv:2001.08317 (2020)
32. Zhang, S., Liu, B., Whiteson, S.: Mean-variance policy iteration for risk-averse reinforcement learning. In: AAAI (2021)