ESSCA

MASTER'S THESIS 2024-2025

Reinforcement learning for Dynamic Assets Allocation to fulfill futures Liabilities: T3D algorithm with Digital Portfolio Theory framework

Etienne, LARCHET
Finance, Risk & Compliance
2001775

Supervisor: Bushra GHUFRAN

Table of Contents

Introduction

Asset analysis targets the ideal portfolio allocation by minimizing risk while maximizing returns. One of the eminent figures in portfolio strategy is Harry Markowitz, whose 1952 paper introduced Modern Portfolio Theory (MPT), widely recognized as the foundation of portfolio optimization. MPT provides a framework to construct investment portfolios that maximize expected returns for a given level of risk. It emphasizes diversification and reduces risk by combining assets with small correlations. It leads to an efficient frontier of optimal portfolios, that is drawn using Sharpe ratio.

For banks and especially life insurance companies, effective portfolio optimization is crucial, as it must balance both asset returns, and future liabilities given by the client contracts.

To address this, the Asset Liability Management (ALM) department is primary focused on optimizing asset allocation while managing liabilities. As defined by Wekwete et al. (2023: 1), ALM aims to "*[…] derive an optimal investment asset allocation strategy for reducing interest rate risk exposure by considering both current and future liabilities*". [1]

A classic ALM approach is Redington's Immunization model, introduced in 1952 (Redington, 1952, cited by Wekwete et al., 2023 [1], and Shiu, 1990 [2]). This model focuses on matching the duration of assets and liabilities so that changes in interest rates affect both and therefore reduce interest rate risk. However, the traditional ALM framework has notable limitations. One significant drawback is the heavy reliance on human judgment for investment decisions and reallocations [1]. This reliance makes traditional approaches vulnerable to behavioral biases, such as *"[…] confirmation bias, overconfidence, recency bias, availability bias, and many other biases.*" (Syed & Bansal, 2018, Rabbani et al., 2021, Chiu et al., 2022, Bondt et al., 2013, cited by Wekwete et al., 2023 : 2) [1]. These biases can distort decision-making and affect the overall performance of ALM strategies.

One way to balance these issues is to reduce human involvement in the decision-making process. Many researchers have demonstrated that the use of machine learning offers superior results in

complex decision environments like ALM (Wekwete et al., 2023, Jang & Seong, 2023, Fontoura et al., 2019, Lim et al., 2022, Ruyu et als., 2024). [1] [3] [4] [5] [6]

Machine learning (ML), a branch of artificial intelligence, focuses on developing systems that can learn from data and improve their performance over time without explicit programming. To our knowledge, three types of learning in ML coexist: supervised learning, unsupervised learning, and reinforcement learning.

This paper will not elaborate on the first two, as they are not well suited for portfolio optimization due to their lack of learning abilities once the model has been trained. Reinforcement learning, on the other hand, is highly suited for ALM and portfolio optimization tasks. RL is a type of machine learning where an agent interacts with its environment and learns by receiving feedback based on its actions. The learning process is iterative, where the agent refines its strategy (that is called policy) to maximize the cumulative reward over time. Several strategies have been implemented, the main one being the Bellman equation that handles the times series and discount factor. This equation is detailed in Ashin and Tikhon 2022 paper [7]. The underlying structure that governs RL is the Markov Decision Process (MDP), which formalizes the decision-making framework.

The MDP is defined as a "*class of stochastic sequential decision processes in which the cost and transition functions depend only on the current state of the system and the current action"* (Puterman, 1990: 1) [8].

A sequential decision process being a model where the decision maker (called agent) observes the state (S) of the system and performs an action (A) from a set of available actions. Puterman explains the consequences of choosing an action as *"[…] twofold; the decision maker receives an immediate reward, and specifies a probability distribution on the subsequent system state. If the probability distribution is degenerate, the problem is deterministic. The decision maker's objective is to choose a sequence of actions called a policy, that will optimize the performance of the system over the decision making horizon. Since the action selected at present affects the future evolution of the system, the decision maker cannot choose his action without taking into account future consequences.*" (Puterman, 1990: 1) [8].

In other words, the agent's goal is to learn the optimal policy that maximizes the expected cumulative reward over time. In financial markets, this translates into continuously adjusting a portfolio to maximize returns while minimizing risk, all in response to a changing market environment.

In the context of ALM, Deep Deterministic Policy Gradient (DDPG), a variant of reinforcement learning, has been widely used within the Modern Portfolio Theory framework to optimize asset allocations. DDPG is an actor-critic algorithm specifically designed to handle continuous action spaces, which makes it ideal for tasks such as portfolio rebalancing. The algorithm learns a deterministic policy (the actor) and evaluates it using a value function (the critic), improving its decisions over time based on feedback from the environment. This type of deterministic algorithm is privileged over machine learning stochastic policy gradient as they require less computing power in high dimensional environment (Silver et al., 2014) [9].

Gap in literature

Despite the advances brought by DDPG in ALM and portfolio optimization, existing research has primarily focused on its application within the MPT framework. However, the gap in literature lies in the fact that MPT is inherently myopic and not suited for long-horizon risk management, particularly when incorporating multi-period objectives or mean-reverting risks. This has been studied by C. Kenneth Jones that compared the MPT with Intertemporal Portfolio Theory (IPT) and Digital Portfolio Theory (DPT) (Kenneth Jones, 2017) [10]. The paper put in lights DPT as a hybrid framework, that gives a single-period non-myotic solution and estimates mean-reversion risk levels.

Current research has yet to explore the potential of advanced reinforcement learning algorithms, such as Twin-Delayed Deep Deterministic Policy Gradient (TD3) that is an extension of DDPG that addresses some of its shortcomings and introduces improvements like using two critics to mitigate overestimation bias. A consensus of researchers considers TD3 as an extend and improvement of *"the DDPG algorithm by introducing new features to make it increasingly stable during training and to improve convergence speeds"*(Jiang et al., 2024: 4) [11]. Its usage in finance

is still not widely implemented, with few papers using it, however as of my knowledge, no one has used it with modern framework like Digital Portfolio Theory which is designed for long-term portfolio optimization.

This thesis addresses this gap by investigating how TD3 can be integrated into ALM within the context of DPT, providing a more dynamic and flexible approach to asset allocation. Unlike MPT, DPT accommodates long-horizon risks and mean-reversion, making it a more suitable framework for institutional investors aiming to balance returns and liabilities over time.

Through this exploration, the research aims to answer the following question:

Can the TD3 reinforcement learning algorithm, within the DPT framework, improve asset allocation outcomes in ALM by optimizing portfolio performance and fulfilling its long-term liabilities?

Research Objectives and Approach

The goal is to build and train a model using the TD3 algorithm, which is particularly well-suited for high-dimensional and continuous action spaces, such as those found in financial portfolios. The model will be trained using the DPT framework, which emphasizes the long-horizon, mean-reversion nature of financial returns, as opposed to the short-term, risk-return trade-offs inherent in MPT. The model will follow two guiding principles for the design of the policy and reward-punishment function:

- Short-term goal: maximize returns while simultaneously reducing systemic risk. The model will have to optimize asset allocations dynamically, adjusting based on market signals to capitalize on opportunities while keeping systemic risk under control.
- Long-term goal: address long-term liabilities and progressively allocate assets to less volatile instruments as the liabilities approach maturity. This will require the model to balance short-term portfolio growth with the need to manage long-term obligations and ensure that sufficient assets are available when liabilities are due.

This research will provide insights into the advantages of adopting a long-horizon, mean-reversion approach and the potential for reinforcement learning for asset allocation strategies in ALM.

Academic literature

Modern Portfolio Theory

Modern Portfolio Theory (MPT), developed by Harry Markowitz in his works of 1952 and 1959, fundamentally changed the way investors approach portfolio construction. Before MPT, investment decisions were largely based on individual asset characteristics, such as returns and risks, without a structured framework for considering how assets interact within a portfolio. Markowitz introduced a mathematical and systematic approach to investing, emphasizing that asset selection must account for the interrelationships between assets, particularly their covariances, to optimize the risk-return tradeoff.

Markowitz's mean-variance framework established two fundamental principles:

- For a given level of risk, investors should maximize expected return.
- For a given level of expected return, investors should minimize risk (variance).

As summed by Elton and Gruber, these principles led to the concept of the efficient frontier and the mean-variance, which represent a method to determine optimal portfolios offering the highest expected return for a given level of risk (Elton & Gruber, 1997) [12]. The efficient frontier became the cornerstone of MPT, illustrating that diversification reduces risk not just by holding more assets but by selecting assets with low or negative correlations.

At the heart of MPT lies the recognition that asset returns do not move independently, and therefore in the construction of a portfolio, an investor "*had to consider how each security co-moved with all other securities*" (Elton & Gruber, 1997:2) [12]. This interdependence means that well-diversified portfolios can achieve a lower total risk than the sum of individual risks. By quantifying both expected returns and risks, MPT provides a framework for constructing portfolios that balance an investor's risk tolerance with their return objectives.

MPT operates under the assumption of normally distributed returns and a single-period investment horizon, simplifying mathematical modeling but limiting its applicability in more complex or multi-period scenarios.

Limitations

While MPT remains a foundational framework, several extensions and modifications have addressed its limitations:

1. **Mean-variance simplification**: MPT assumes that the investor's sole objective is to maximize returns for a given level of risk, measured by variance. The simplification of focusing only on mean and variance in the portfolio optimization problem was criticized by Tobin in 1958 (cited by Elton & Gruber, 1997) [12]. Other factors, such as skewness and kurtosis, are ignored, which led to the emergence of other portfolio theories including such indicators (Lee, 1977, Kraus & Litzenberger, 1976, cited by Elton & Gruber, 1997) [12].

2. **Liabilities**: The theory primarily focuses on asset returns without adequately considering liabilities. In their 1992 paper, Elton and Gruber emphasized the importance of incorporating liabilities into portfolio optimization (Elton & Gruber, 1992, cited in their 1997 paper) [12]. This approach recognizes that for institutional investors, such as pension funds or insurance companies, the asset allocation process must account for future cash flow obligations, considering the uncertainty and systematic risks inherent to these liabilities.

3. **Stationarity of Inputs**: MPT assumes that returns are independent and identically distributed (IID), with no predictive time variation, and focuses solely on the trade-off between risk and return in the immediate future (Kenneth Jones, 2009) [13]. This myopic approach fails to account for long-term risks, such as mean-reversion effects or horizon-dependent risks, making it less suitable for investors with extended time horizons

4. **Quadratic computation**: In the portfolio optimization problem, MPT's reliance on quadratic programming often results in unstable and extreme portfolio weights, which are sensitive to estimation errors in means and covariances. The quadratic computation is inherent to the MPT, that relies on a covariance matrix for the mean-variation calculation. Furthermore, optimizers of the mean-variance problem are estimation error amplifiers (Michaud, 1989, cited by Kenneth Jones, 2009) [13] .

5. **Single period problem:** The estimation of mean return and mean variance for each asset is over a single period. To solve the multi period problem, several researchers processed with a sequence of single period problems (Fama, 1970, Hakansson, 1970 and 1974, cited by Elton & Gruber, 1997) [12]. However, this method follows the assumption that returns are independent between periods, which were proven later to be dependents (Fama and French, 1989, Cambell and Shiller, 1988, cited by Elton & Gruber, 1997) (Kenneth Jones, 2017) [12] [10].

Digital Portfolio Theory

To address these gaps in the MDP, C. Kenneth Jones introduced a theorical portfolio optimization framework named Digital Portfolio Theory (DPT) (Kenneth Jones, 2001) [14]. This framework extends the foundational concepts of Modern Portfolio Theory (MPT) by incorporating advanced techniques from digital signal processing to address the limitations of MPT, particularly for long-term investment horizons. While MPT optimizes portfolios by balancing expected returns and risk in a single period, DPT addresses the complexities of long-term investment by incorporating mean-reversion risks, holding periods, and the dynamic nature of financial markets.

The Innovation of Digital Portfolio Theory

DPT incorporates insights from digital signal processing, a field traditionally associated with engineering and communications. At its core, DPT decomposes portfolio variance into systematic and unsystematic components while also accounting for periodic risks, such as calendar anomalies and mean-reversion effects. Anomalies returns on months (Linn & Lockwood, 11988, Hensel & Ziemba, 1996, Penman, 1987), seasons (Wachtel, 1942) and presidential elections periods (Booth & Booth,1999), have been cited by Kenneth Jones in his 2001 [14] and 2009 [13] papers on the DPT. Mean reversion refers to the tendency of asset returns to revert to their long-term average over time (Poterba & Summers, 1988, Fama & French, 1998, cited by Kenneth Jones, 2001) [14]. This decomposition enables investors to manage risks across different time horizons, aligning portfolio decisions with long-term goals and expectations.

A fundamental technical innovation in DPT is its use of the Fourier transform, which allows for the decomposition of financial time series into frequency components. By treating asset returns as signals, the Fourier transform translates these time-domain data into the frequency domain, enabling the identification of periodic risk components such as short-term volatility and long-term mean-reversion trends. Low frequencies represent stable, long-term risks, while high frequencies capture short-term fluctuations. This granular view of risk allows investors to manage portfolio exposures to specific frequencies, tailoring their portfolios to align with anticipated cycles, such as quarterly earnings effects or multi-year mean-reversion patterns.

DPT's integration of Fourier analysis facilitates a refined decomposition of risk into systematic and unsystematic components. Systematic risk, which reflects market-wide influences, and unsystematic risk, which pertains to individual securities or sectors, are analyzed in the context of multiple time horizons. By using autocovariance data derived from the Fourier-transformed signals, DPT enables precise risk tuning. Investors can explicitly control the proportion of systematic and unsystematic risks they wish to bear, ensuring that their portfolios align with their risk tolerance and investment objectives. This decomposition also helps achieve efficient diversification by mitigating unsystematic risk without over-diversifying into excessively large portfolios.

Another notable advancement of DPT is its ability to impose constraints on portfolio size, addressing practical challenges associated with over-diversification and the management of large portfolios. Unlike MPT, which often leads to portfolios with many negligible allocations, DPT employs mixed integer programming (MIP) to set a specific limit on the number of assets in the portfolio. This constraint is implemented using a zero-one integer variable. By allowing investors to specify their preferred portfolio size, DPT accommodates diverse investment styles, whether emphasizing concentrated active management or broader passive diversification. This is crucial because no consensus has been found by research on the ideal portfolio number assets to achieve optimal diversification (Kenneth Jones, 2009) [13], but also because the DPT can be used by individuals investors, which compose small portfolios, and by institution investors that diversify in hundreds of different assets.

Another critical technical feature of DPT is its prevention of extreme portfolio weights, a common drawback of MPT (Green & Hollifield, 1992, cited by Kenneth Jones, 2017) [10]. MPT's quadratic optimization approach amplifies estimation errors in expected returns and covariances, often resulting in highly unstable and concentrated allocations. DPT addresses this by utilizing linear programming, which avoids quadratic terms and is less sensitive to input errors. Furthermore, DPT incorporates shrinkage techniques to stabilize estimates of means and covariances, pulling them toward more robust, generalized values. These enhancements reduce the likelihood of extreme allocations and ensure that portfolio weights remain stable and realistic. Additional constraints, such as upper and lower bounds on individual asset weights, allow investors to further control their allocations, preventing any single security from dominating the portfolio or being allocated an insignificant share.

Applications in Portfolio Management

DPT is particularly well-suited for institutional investors, such as pension funds, that manage long-term liabilities. By incorporating mean-reversion effects and horizon-dependent risks, DPT helps these investors construct portfolios that align with their extended holding periods and risk tolerances. Furthermore, DPT's ability to manage periodic risks makes it an effective tool for tactical adjustments, such as positioning portfolios to benefit from anticipated seasonal trends or macroeconomic cycles.

Limits of the DPT

Digital Portfolio Theory (DPT), while innovative, has several limitations that may hinder its practical application. One significant challenge is its reliance on large datasets. Empirical tests of DPT often require extensive historical data, requiring 16 years or more data prices to accurately capture long-term mean-reversion patterns and periodic risks. This requirement can be restrictive, particularly for emerging markets or new asset classes with limited historical records.

Also, since its introduction in 2001, there appears to be no significant evidence of its adoption in practical financial applications or academic benchmarks. Unlike the MPT, which remains the dominant framework for portfolio optimization, DPT has yet to be tested extensively in real-world scenarios or compared rigorously against the established methodologies. As a result, while DPT

offers intriguing potential, further empirical testing and validation are essential to evaluate its practical effectiveness and determine whether it can truly outperform or complement MPT in portfolio optimization.

A Comparison with MPT

When compared to MPT and intertemporal portfolio choice models, DPT emerges as a superior framework for long-term portfolio optimization. While MPT is limited to single-period decisions and intertemporal models are often computationally complex and reliant on restrictive assumptions, DPT strikes a balance between practicality and sophistication. Its linear programming approach allows for efficient optimization, while its integration of mean-reversion risks ensures relevance for long-term investment strategies.

	Digital Portfolio Theory	Modern Portfolio Theory
Time Horizon	Long-term focus; incorporates mean-reversion and periodic risks.	Short-term focus; assumes a single-period optimization.
Risk	Separates risk into systematic, unsystematic, and periodic components.	Low control on systematic and unsystematic risks
Data Requirements	Requires extensive historical data (16+ years) to analyze long-term trends and autocorrelations.	Requires shorter historical datasets to estimate means and covariances for a single period.
Portfolio Control	Control on the number and weights of assets	No direct control over portfolio size and assets weights requirements, with tendencies to extreme weights
Mean-Reversion Handling	Explicitly incorporates mean-reversion risk, allowing for hedging and speculative strategies across different time horizons.	Assumes IID returns, ignoring mean-reversion and long-term trends.
Periodic Risk Management	Integrates periodic risk (e.g., calendar anomalies) using frequency-domain analysis.	Does not account for periodic risks or time-dependent autocorrelation.
Computational Efficiency	More scalable for large universes due to linear programming, even with constraints like portfolio size.	Computationally expensive for large universes due to the inversion of large covariance matrices and quadratic programming.
Practical Applications	Not widely adopted; requires further empirical testing to validate its practical effectiveness.	Well-established and widely used by both researchers and companies.

Reinforcement learning

In recent years, many researchers have proven the efficiently of machine learning in portfolio weights optimization, most of them using the mean-variance theory of the MPT as base to the reward function. This section will recompose the advancement over time of empirical studies on the resolution of the portfolio optimization problem with the leverage of machine learnings algorithms.

Machine learning with the MPT

In 2022, Pinelis and Ruppert propose a machine learning algorithm that implements the MPT (Pinelis & Rubbert, 2022) [15]. They created a utility-maximizing framework that optimizes portfolio weights between a market index and a risk-free asset using two Random Forest models to capture expected returns and volatility. The first Random Forest model predicts the expected monthly excess returns using macroeconomic and financial variables such as payout yields, term spreads, and inflation rates, while the second estimates prevailing volatility using lagged realized volatility and similar predictors. Empirical results validate the efficacy of this implementation, demonstrating a 28% improvement in Sharpe ratios over traditional buy-and-hold strategies, with a heavy annualized alpha of 3.4%.

Type of reallocation strategy

The same year, in 2022, some researchers begin the implementation of reinforcement learning to solve the portfolio optimization problem (Lim et al, 2022) [5]. The focus is on the assets allocation strategies with the RL algorithm testing strategic and tactical asset allocation (SAA and TAA). As explained by Bouyé, SAA is a medium-to-long term strategy that bonds each asset with minimal and maximal allocation, while TAA is a short term strategy that shifts the assets weights depending on the market conditions (Bouyé, 2018) [16]. The challenges of investors, being the conciliation of the two strategies (Aglietta et al., 2007, cited by Bouyé, 2018) [16].

The selected reinforcement learning is a Q-learning type of algorithm, which unlike most similar studies is a value-based algorithm that uses neurons layers (in opposition with policy based and actor critic algorithms). The research uses inputs derived from technical indicators, including

EMAs and MACD values. The implementation demonstrated that a gradual rebalancing method incorporating Long Short-Term Memory (LSTM) models for price prediction yielded the best results, improving NAV returns by 27.9% to 93.4% over a full rebalancing strategy without predictions. The explanation was the gradual adjustments reduced transaction costs and penalties associated with abrupt changes, leading to superior risk management and enhanced overall returns compared to individual assets within the portfolios.

DRL with the MPT using Tucker decomposition

Then, in 2023, Jang and Seong successfully implemented the MPT in a Deep Reinforcement Learning (DRL) algorithm with risk-adjusted returns and diversification principles (Jang & Seong, 2023) [3]. For the first time, a reinforcement algorithm utilizes the Tucker decomposition to bridge the gap between historical price correlations and technical indicators, such as moving averages, RSI, and MACD. This multimodal approach processes high-dimensional tensors (multi-dimensional vectors) representing the covariance of returns and technical features. These tensors are then subjected to 3D convolutional neural networks for feature extraction and dimensionality reduction, followed by a deep deterministic policy gradient (DDPG) reinforcement learning framework. The reward function was designed to maximize the portfolio's terminal value while accounting for transaction costs and risk, reflected through metrics like the Sharpe ratio and maximum drawdown.

Deep Reinforcement Learning

More recently, another study was conducted by Yan et al. in 2024, that also integrates the MPT for portfolio optimization. The article especially fills the gaps in the literature about the lack of consideration for transaction costs and risk volatility in existing RL-based portfolio optimization models. The proposed framework uses the Deterministic Policy Gradient (DPG) algorithm, which is well-suited for continuous decision-making and robust against the inherent nonstationarity of financial markets. Inputs to the model include normalized price vectors of assets, comprising historical data for opening, closing, highest, and lowest prices across multiple timeframes. Additionally, convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) are employed to extract respectively asset correlations and temporal patterns. Unlike

the precedent study, authors choose not to provide any technical indicators, to give more control for the CNNs and LSTM algorithms with raw prices data. The MPT was implemented in the reward function with a risk-cost reward function to optimize the trade-off between portfolio returns and risk. The model incorporates transaction costs by modeling them using the one-norm of portfolio transaction vectors. Specifically, the function is designed to penalize frequent trading and estimation errors while rewarding stable accumulative portfolio returns.

Implementation of the TD3 algorithm

Jiang et al. achieve another breakthrough in the portfolio optimization problem by utilizing the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, providing a comparison with the well-established DDPG approach (Jiang et al., 2024) [11]. This algorithm is employed to address the complexities of portfolio optimization in high-dimensional and dynamic financial markets by improving upon the DDPG with the introduction of three key enhancements: target policy smoothing, double Q-learning, and delayed policy updates. These updates reduce overestimation bias and improve training stability (Fujimoto et al, 2018) [17]. The TD3 framework utilizes actor-critic architecture, where the actor network outputs deterministic actions, and two critic networks estimate the Q-values (expected cumulative reward function) to avoid overestimation. Innovations include target policy smoothing, which applies clipped Gaussian noise to the action output to improve generalization, and a delayed update mechanism, where the actor is updated less frequently than the critics to enhance convergence stability.

The method also models the portfolio optimization problem as MDP, with states defined by historical asset prices and portfolio weights, actions representing asset reallocation decisions and rewards incorporating mean-variance framework, as well as transaction costs, and risk aversion. The proposed framework successfully balances risk and return, achieving reduced maximum drawdowns and higher cumulative returns compared to other deep reinforcement learning methods.

Multiagent DRL

Lastly, Cheng and Sun proposed an advance algorithm that also relies on TD3 algorithm, but implementing a multiagent structure (Cheng & Sun, 2024) [18]. They underscore many gaps in

the literature, including the inability of single-agent frameworks to handle chaotic, multi-asset environments and the tendency of traditional models to overfit or produce unstable returns. The framework allows multiple agents to independently explore distinct assets while sharing learned parameters with a global network. Each agent focuses on specific stock features, with two primary modules: the Trading Action Module (TAM) and the Trading Portfolio Module (TPM). The TAM integrates CNNs and LSTMs to analyze candlestick patterns (open, close, high low of daily prices) generating asset-specific trading actions (long, short, hold). TPM evaluates these actions, assigns asset scores based on technical indicators, and determines portfolio weights dynamically. This modular design avoids the pitfalls of single-agent models by enabling robust exploration of diverse market conditions. The reward function uses the Sortino ratio, which prioritizes downside risk management by focusing on negative deviations from expected returns, thereby enhancing stability in volatile markets. The multi-agent frame introduces asynchronous training via Asynchronous Advantage Actor-Critic (A3C), wherein each agent optimizes its environment-specific policy before aggregating results globally. This decentralized approach accelerates training and reduces inter-agent correlation issues.

Application in Assets Liabilities Management

This last section synthesizes insights from two seminal works on Wekwete et al. (2023) [1] and Fontoura et al. (2019) [4], which apply DRL frameworks to revolutionize ALM practices. As of our knowledge, these 2 studies constitute the overall work done on the subject, highlighting the important gap in literature, and hindrance compared to the explored dynamic assets allocation models.

Traditional ALM approaches, particularly duration matching through Redington Immunization (Shiu, 1990) [2], aim to align the timing of asset cash flows with liability outflows. However, as explained by Wekwete, this method requires frequent rebalancing, which is both labor-intensive and prone to human biases, such as overconfidence and recency effects (Wekwete et al., 2023). Therefore, the need for automated, adaptive, and robust solutions, is paving the way for DRL-based approaches.

In Wekwete et al. (2023) [1], the state space includes liability durations and asset maturities derived from Monte Carlo simulations, and actions represent asset allocation decisions, such as the weights assigned to short- and long-term bonds. The reward function is designed to minimize the absolute mismatch between asset and liability durations, thus ensuring effective duration matching.

Similarly, Fontoura et al. (2019) leverage the DDPG algorithm. Unlike Wekwete et al. approach that discretize state spaces into scenario trees, DDPG enables the use of continuous state and action spaces, which better reflect the stochastic and dynamic nature of ALM. The algorithm comprises an actor-critic architecture: the actor maps the actions (buy, short, hold asset allocations), while the critic evaluates the quality of these actions based on Q-values. By incorporating a discount factor in the reward function, the model emphasizes immediate gains while maintaining a forward-looking perspective on liability management.

Academic conclusion and model introduction

This thesis lays the groundwork for a novel approach to Asset–Liability Management by integrating Digital Portfolio Theory with a multiagent Deep Reinforcement Learning framework. While Modern Portfolio Theory has traditionally shaped portfolio optimization, it struggles with long-term and mean-reversion dynamics that are critical for institutional investors. DPT addresses this gap by incorporating Fourier transforms, frequency-domain analysis, and advanced constraints, making it more suitable for horizon-dependent risks and large-scale asset universes.

On the machine learning side, Deep Reinforcement Learning has demonstrated robust performance in dynamic asset allocation, although existing studies predominantly employ MPT-based reward functions. Moreover, single-agent RL models may struggle with the complexity of multi-asset environments. As such, the multiagent extension of RL, supported by techniques like Twin Delayed Deep Deterministic Policy Gradient offers a framework that can better handle multiple asset dynamics, reduce overfitting, and address scalability challenges.

Against this backdrop, the goal of our research is to develop a multiagent DRL model that leverages the DPT framework to adaptively manage asset weights and positions. By doing so, the system aims to maximize returns while considering long-term liabilities, thereby ensuring that portfolios remain well-positioned to meet future obligations.

Model description

The portfolio optimization model developed in this thesis is built upon a modular architecture, drawing inspiration from the task division frameworks proposed the work by Cheng and Sun [18]. However, our implementation introduces a refined three-module structure: Forecasting Module, Optimization Module, and Decision Module. This chapter details the theoretical underpinnings and practical implementation of each module.

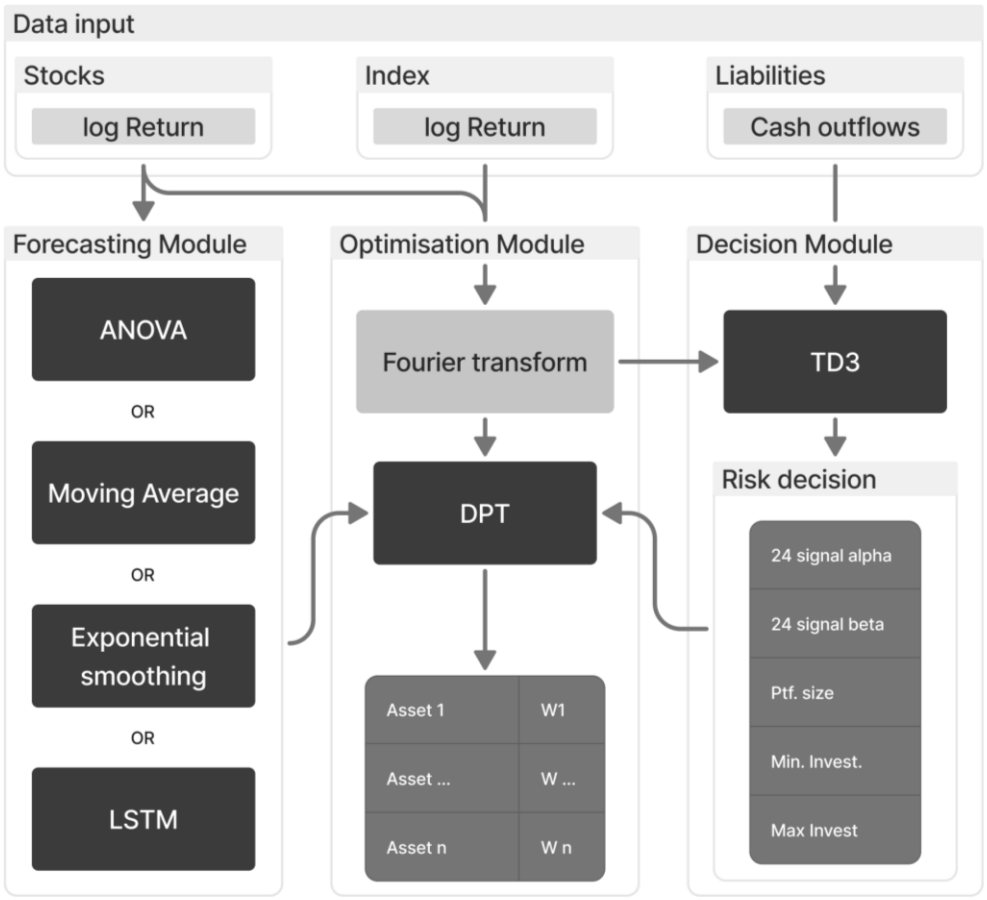
Overall Model Description

The overarching goal of the implemented framework is to dynamically manage a portfolio of assets to meet a series of future liabilities while maximizing returns under defined risk constraints. The system operates through the interplay of three specialized modules:

1. **Forecasting Module:** This module is responsible for generating predictions of future asset returns. It employs a suite of econometric and machine learning techniques to estimate the expected monthly performance of the assets under consideration.
2. **Optimization Module:** At the heart of the framework lies the Optimization Module. This module implements the DPT. It processes the return forecasts and raw market data to construct an optimal portfolio according to the DPT principles and constraints set by the Decision Module.
3. **Decision Module:** This module leverages a RL agent, specifically the TD3 algorithm. The agent is tasked with making high-level strategic decisions, such as defining the risk parameters (Cαk,Cβk), the desired number of assets (S), and minimum/maximum holding constraints (L_j,M_j) for the DPT optimization. The constraint ,M_j is not inherent to the original DPT formulation but was introduced in this model to enhance stability and promote equilibrium in the portfolio weights.

A significant innovation of this framework is the explicit integration of liabilities into the decision-making process. The TD3 agent in the Decision Module receives information about upcoming liabilities and incorporates this into its policy, guiding the Optimization Module to construct portfolios that are cognizant of these future obligations.

Figure 1 TD3-DPT Model Framework



The model is inherently dynamic. At each discrete time step (month), the entire system is re-evaluated. Logarithmic returns of the constituent assets and a chosen market index serve as primary inputs. These are fed into the Forecasting Module for return prediction and directly into the Optimization Module for the signal processing steps crucial to DPT. Based on the new forecasts and the policy dictated by the TD3 agent, the Optimization Module recalculates the optimal portfolio composition.

The model is initialized with a starting portfolio value and a predefined schedule of liabilities. As time progresses, if a liability matures, its value is deducted from the portfolio. The episode concludes either when all liabilities are met or when the portfolio value is depleted. Premature termination can also occur if the RL agent fails to converge on a stable policy.

Like Markowitz's mean-variance framework, Digital Portfolio Theory requires an estimation of future asset returns. However, DPT significantly departs from traditional models in its conceptualization and quantification of risk. While the Forecasting Module addresses the estimation of returns, the Optimization Module employs Fourier Transform techniques to decompose and analyze risk across various frequencies, corresponding to different investment horizons. These components—predicted returns and frequency-decomposed risk—are then used by a solver within the Optimization Module to construct portfolios that adhere to the specified constraints.

Forecasting Module

The accurate estimation of future asset returns is a cornerstone of any portfolio optimization strategy. As noted by C. Kenneth Jones [13] in his 2009 seminal works on Digital Portfolio Theory, there is no universally prescribed method for determining these expected returns. Our Forecasting Module, therefore, was designed to accommodate and evaluate several techniques, with an initial focus on established time series models. The implementation, detailed in the *Forecasting.py* script, currently prioritizes two primary methods for generating one-step-ahead monthly return forecasts:

Auto-ARIMA (Auto-Regressive Integrated Moving Average)

The Auto-ARIMA model is a widely used statistical method for time series forecasting that automatically discovers the optimal order of an ARIMA model. An ARIMA(p,d,q) model combines autoregressive (AR) components, differencing (I for integrated), and moving average (MA) components to capture the temporal structure of a time series. The p refers to the number of lag

observations included in the model, *d* is the number of times raw observations are differentiated, and *q* is the size of the moving average window.

Our implementation utilizes the Auto-ARIMA class from the *statsforecast* Python library. Key configurations include specifying a seasonal length of 12 months to capture potential annual cyclical patterns in monthly asset returns. This allows the Auto-ARIMA function to also consider Seasonal ARIMA (SARIMA) models.

To enhance computational efficiency, an approximation method is enabled within the Auto-ARIMA function. This is particularly crucial given that forecasts must be generated for each asset at every time step of the simulation, which can be computationally intensive. While this offers speed, it may come at a minor cost to precision. The Auto-ARIMA algorithm typically employs an information criterion to select the best model. In this context, the Akaike Information Criterion with correction (AICC) is the metric used to guide the selection of the optimal ARIMA orders. The forecasting process for Auto-ARIMA can be conceptually outlined with the following pseudo-code:

Pseudo-code 1 TD3-DPT Model Framework

```
# The model automatically searches for optimal (p,d,q)(P,D,Q) orders
model = AutoARIMA_Model(season_length=seasonal_length,
                        approximation=True,
                        information_criterion='aicc')
# Fit the model to the historical log returns for each asset

fitted_model = model.fit(time_series_data)
prediction = fitted_model.predict(steps=1) # Predict one step ahead
```

Moving Average

A simpler, yet often effective, forecasting technique implemented is the moving average. This method calculates the average of the logarithmic returns over a specified trailing window. The

moving average function in *Forecasting.py* allows for a flexible window size, W. The forecast for the next period, $\hat{r}_{t+1}$, is given by:

$$\hat{r}_{t+1} = \frac{1}{W}\sum_{i=0}^{W-1} r_{t-i}$$

where r_{t-i} are the past logarithmic returns and W is number of past periods to include in the average. This method was explored with W ranging from 5 months up to the maximum observation period of 192 months.

Future Enhancements: LSTM and Exponential Smoothing

The initial design of the Forecasting Module also included provisions for more advanced techniques like Long Short-Term Memory (LSTM) networks and various Exponential Smoothing methods. LSTMs, a type of recurrent neural network, are particularly adept at learning long-range dependencies in time series data. Exponential Smoothing methods assign exponentially decreasing weights to past observations. However, these were not fully implemented and tested within the current scope of this thesis. They remain promising avenues for future research and potential enhancement of the model's predictive capabilities.

Optimization Module (DPT implementation)

The Optimization Module is the cornerstone of this research, responsible for the practical implementation of Digital Portfolio Theory (DPT) as pioneered by C. Kenneth Jones. DPT extends traditional portfolio theory by providing a more granular view of risk, decomposing it across multiple investment horizons through the application of digital signal processing techniques, particularly the Fourier Transform.

Data Preparation

The foundational data for the DPT module consists of a time series of asset prices. The first step is the conversion of these adjusted closing prices into logarithmic returns *r*:

$$\mathrm{r} = \ln(P_t - P_{t-1})$$

where P_t is the adjusted closing price at time t. Logarithmic returns are preferred for their additive property over time and their tendency towards normality compared to simple returns.

Signal Calculation via Fourier Transform

A core tenet of DPT is the treatment of these logarithmic return series as digital signals. This allows for the application of Fourier analysis to decompose the variance (risk) of each asset's returns into components associated with different frequencies, and thus, different time horizons.

The Fourier Transform in Financial Time Series

The Fourier Transform is a mathematical operation that converts a signal from time domain to a representation in the frequency domain. For a time series of asset returns, which exists in the time domain, the Fourier Transform reveals the underlying cyclical components of various frequencies. A low frequency corresponds to a long-period cycle (e.g. US presidential election), while a high frequency corresponds to a short-period cycle (e.g. monthly fluctuations).

For a discrete, finite time series $\hat{r}[n]$ consisting of N samples, the Discrete Fourier Transform (DFT) is defined as:

$$X[k] = \sum_{n=0}^{N-1} \hat{r}[n] e^{-i2\pi \frac{k}{N} n}$$

Each X[k] is a complex number representing the amplitude and phase-shift of the signal component at the kth frequency, $f_k = k\,(f_s/N)$, where f_s is the sampling frequency. The magnitude squared of these coefficients, $|X[k]|^2$, is proportional to the Power Spectral Density (PSD), indicating the power (or variance) of the signal concentrated at that frequency f_k. The phase component indicates the timing or alignment of that frequency component. DPT leverages this decomposition to analyze how much risk is associated with different investment horizons.

Table 1 Calendar and Non-Calendar Holding Period Return Horizons

Table I from C. Kenneth Jones, 2009 [13]

k	Period p_k (months)	Calendar Based Risk	Calendar Effect	k	Period p_k (months)	Calendar Based Risk	Calendar Effect
1	48.0	4-yr	Presidential	13	3.7		
2	24.0	2-yr	Election	14	3.4	8-yr	
3	16.0			15	3.2	16-yr	
4	12.0	1-yr	Annual/Summer	16	2.0	1/4-yr	Quarterly
5	9.6			17	2.8		
6	8.0	8-yr		18	2.7		
7	6.9			19	2.5		
8	6.0	1/2-yr	Six Month	20	2.4	1-mo	January
9	5.3			21	2.3		
10	4.8	8-yr	Business Cycle	22	2.2	8-yr	
11	4.4			23	2.1		
12	4.0	1-mo		24	2.0	1-mo	

Welch's Method for Power Spectral Density (PSD) Estimation

Financial time series are often noisy. A direct DFT of such a series can result in high variance, making it difficult to discern the true underlying spectral characteristics. Welch's method provides a more robust and statistically stable PSD estimation by averaging segmented periods. This method, as utilized by Jones (2009, p. 29) [13] reduces the variance of the PSD estimate by averaging modified periodograms obtained from overlapping segments of the original time series As implemented using the *scipy* Python library, and following the parameters of Jones Kenneth, the steps are:

1. **Segmentation:** The input time series (e.g., 192 months of logarithmic returns for an asset) is divided into S_{seg} overlapping segments. In our configuration, the length of each segment, nperseg is set to 48 months. This four-year segment length is chosen to capture a range of business and economic cycles relevant to financial markets. A 50% overlap is used, meaning each subsequent segment starts halfway through the previous one. This

increases the number of segments available for averaging. Therefore, S_{seg} is equal to 192/ (48 * 0.5) -1 = 7 segments.

Table 2 Welch segments

S_{seg}	Period (months)
1	0-48
2	24-72
3	48-96
4	72-120
5	96-144
6	120-168
7	144-192

Pseudo-code 2 Welch parameters

```
# Conceptual Python for Welch's parameters as aligned with Jones (2009)
total_signal_length_Ns = 192 # 16 years of monthly returns

segment_length_T_seg = 48 # months

overlap_points = segment_length_T_seg // 2 # 50% overlap

window_type = "boxcar" # Rectangular window

sampling_interval_dt = 1 # month

sampling_frequency_fs = 1.0 / sampling_interval_dt # 1 cycle/month
```

2. **Windowing:** Each segment is multiplied by a window function. Our implementation uses a rectangular window. This window applies equal weighting (a weight of 1) to all data points within the segment and zero elsewhere. While other windows (e.g., Hanning, Hamming) taper the ends of the segments to reduce spectral leakage, the boxcar window is simpler and is specified in the DPT literature context.

3. **Periodogram Calculation:** The DFT is computed for each windowed segment. The squared magnitude of the DFT result for each segment yields its periodogram, which is an estimate of the power spectrum for that segment.
4. **Averaging:** The periodograms from all S_{seg} are then averaged point-by-point for each frequency bin. This averaging process is key to Welch's method, as it reduces the variance of the final PSD estimate, providing a smoother and more statistically reliable representation of the signal's power distribution across frequencies compared to a single periodogram of the entire signal.

The welch function in *scipy.signal* returns an array of sample frequencies and an array of corresponding PSD values for each asset.

Harmonics (R_{kj} values)

Sample frequencies are discarded, and only PSD estimates are retained. These are used to derive the harmonic amplitudes (standard deviation), denoted as R_{kj}. For each asset *j* and each harmonic frequency *k* (excluding the 0th harmonic, which corresponds to the signal mean and is not considered part of the periodic risk in DPT), R_{kj} is calculated as the square root of the PSD at that frequency:

$$R_{kj} = \sqrt{PSD_{kj}}$$

For a segment length $T_{seg} = 48$ months and monthly data ($\delta_t = 1$ month), there are $\mathrm{K} = T_{seg} / (2\delta_t) = \frac{48}{2} = 24$ such harmonic components considered in the optimization.

Phase Shift (ϑ_{kmj}), Alpha, and Beta Determination

To distinguish between systematic (market-related) and unsystematic (asset-specific) risk at each harmonic frequency, DPT analyzes the relationship between each asset and a designated market benchmark (index). This is achieved through:

- **Cross-Spectral Density (CSD):** The *scipy.signal.csd* function calculates the CSD between the logarithmic returns of each asset *j* and the market index *m*. The CSD_k is a complex

number that measures the shared power and the phase relationship between the two signals at each frequency *k*.

- **Phase Angle (ϑ_{kmj}):** The phase angle ϑ_{kmj} is extracted from the CSD using *numpy.angle*. This angle represents the phase lead or lag of the kth harmonic of asset *j*'s returns relative to the kth harmonic of the market index m's returns. It is the measure of correlation with index on each harmonic.

From these quantities, the risk components are derived:

- **Systematic Risk (Beta component) Coefficient:** The term $R_{kj}\cos(\vartheta_{kmj})$ quantifies the portion of asset *j*'s kth harmonic risk that corresponds to systematic risk.
- **Unsystematic Risk (Alpha component) Coefficient:** The term $R_{kj}\sin(\vartheta_{kmj})$ quantifies the portion of asset *j*'s kth harmonic risk that is in quadrature (90 degrees out-of-phase) with the market index's kth harmonic that corresponds to unsystematic risk.

Optimization Problem Formulation

With the expected returns (μ_j) from the Forecasting Module and the DPT risk parameters (R_{kj}, $\cos(\vartheta_{kmj})$, $\sin(\vartheta_{kmj})$) calculated, the Optimization Module formulates and solves a portfolio optimization problem. This is implemented using the *PuLP* library in Python, which allows for the definition of linear and mixed-integer linear programming problems (MIP). The solver employed is the default open-source CBC (COIN-OR Branch and Cut) solver.

In the 2009 improved DPT framework, Jones Kenneth defined the solver objective and constraints as follows. The objective is to maximize the expected portfolio return:

$$E(\widehat{r_p}(t) = \sum_{j=1}^{N} W_j \mu_j$$

Subjects to constraints :

$$k = 1,2,3,\ldots,K$$

$$\left|\sum_{j=1}^{N} W_j R_{kj}\cos(\vartheta_{kmj})\right| \leq C\beta_k$$

$$\left| \sum_{j=1}^{N} W_j \, R_{kj} \sin(\vartheta_{kmj}) \right| \leq C\alpha_k$$

$$\sum_{j=1}^{N} z_j = S$$

$$w_j - z_j \leq 0$$

$$w_j - L_j z_j \geq 0$$

$$w_j \geq 0$$

$$z_j = 0 \; or \; 1$$

where $C\beta_k$ and $C\alpha_k$ are right-hand-side (RHS) constants that constrain portfolio risk for each of the 24 harmonics k, W_j demotes the weight of asset *j* in the portfolio (assuming no short-selling), Z_j is a binary variable indicating whether asset *j* is included in the portfolio, thereby controlling portfolio size, and L_j represents the minimum allowable weight for asset *j* in the portfolio.

Divergences with Jones Kenneth DPT implementation

An empirical adjustment introduced during the implementation involve applying a scaling factor to the risk constraint bounds Cβk and Cαk. Empirical results indicated that the raw values of R_{kj} were significantly smaller than those reported by Jones Kenneth. This discrepancy may reflect a diminished mean-variance effect in recent years. To address this, the risk coefficients were multiplied by a scaling factor of 10, ensuring that their magnitudes remain numerically stable for the solver. This adjustment also facilitates more intuitive specification of the Cβk and Cαk bounds by the RL agent. The scaling helps prevent numerical instability due to extremely small coefficients and avoids excessive sensitivity of the risk constraints to minor variations in asset weights.

Additionally, to prevent the solver from disproportionately favoring a particular asset in the resulting portfolio composition, an extra constraint was introduced:

$$w_j - M_j z_j \leq 0$$

where M_j represents the maximum allowable weight for asset *j* in the portfolio.

Decision Module: Reinforcement Learning for DPT continuous set up

The Decision Module elevates the DPT framework from a static optimization tool to a dynamic, adaptive system by employing a RL agent. This agent is tasked with learning an optimal strategy for determining the crucial parameters that govern the DPT optimization at each time step. The objective is to adapt these parameters—such as risk constraints and portfolio composition rules—in response to evolving market conditions and the portfolio's progress towards meeting its liability obligations. The core of this module is a custom-designed environment, *DPTEnv*, built upon the *Gymnasium* (active fork of OpenAI Gym) standard. This environment facilitates the interaction between the DPT model and the TD3 RL agent, implemented using the *Stable-Baselines3* library.

Gymnasium Standard

The Gymnasium library provides a standardized API for creating and interacting with RL environments. This standardization is essential for reproducibility and allows seamless integration with various RL algorithms. Each Gymnasium environment must define an action space (the set of all possible actions the agent can take at a given time step) and an observation_space (the set of all possible states the agent can observe at a given step). The primary methods an environment implement are:

- *reset()*: This method is called at the beginning of each episode. It reinitializes the environment to a starting state and returns the initial observation.
- *step(action):* This method advances the environment by one time step. It takes an action from the agent as input, processes this action, updates the environment's state, calculates the reward, and determines if the episode has terminated. It returns a tuple with the new observations, the reward and status as well as some info for debugging. The terminated flag indicates if the episode ended due to reaching a goal or a failure state intrinsic to the task, while truncated indicates an external reason for ending, like a time limit.

DPTEnv Environment: Interfacing DPT with Reinforcement Learning

The *DPTEnv* class is a custom Gymnasium environment specifically designed to allow an RL agent to control the DPT portfolio optimization process. It encapsulates the simulation of managing a portfolio against a set of liabilities over time.

Environment Initialization : Upon instantiation, *DPTEnv* is provided with the complete historical logarithmic returns, and a *liabilities_window_size* parameter that defines the number of upcoming liabilities visible to the agent in its observation state. This last parameter is needed as the *Gymnasium* observation space requires a fixed data size.

Internally, it manages critical state variables such as the current portfolio value, the detailed schedule of outstanding liabilities, a sliding window mechanism to provide historical return data for DPT calculations, and an instance of the DPT class itself.

Defining the Agent's Actions (action_space) : The agent's decisions directly influence the DPT optimization. The action_space is continuous, defined using *spaces.Box of Gymmasium*. It consists of a vector of 51 floating-point values structured as follows:

1. **Portfolio Size (**S**):** The first element specifies the number of assets to include in the DPT portfolio. Since the entire action_space is composed of floating-point values (due to the limitations of *Box*, which does not support mixed data types), this value is rounded to the nearest integer when integrated into the DPT framework.
2. **Minimum Holding (**L**):** The second element sets the minimum allowable weight for any asset selected in the portfolio.
3. **Maximum Holding (**M**):** The third element sets the maximum allowable weight for any selected asset.
4. **Systematic Risk Constraints ($C\boldsymbol{\beta}_k$):** The next 24 elements define the upper limits for the 24 systematic risk components .
5. **Unsystematic Risk Constraints ($C\alpha_k$):** The final 24 elements define the upper limits for the 24 unsystematic risk components.

This action vector allows the RL agent to dynamically adjust the DPT model's risk posture and diversification at each decision step.

Constructing the Agent's Perception (observation_space) : The information provided to the agent at each step, the observation, is structured as a *spaces.Dict* to accommodate heterogeneous data types and shapes. In Stable-Baselines3, dictionary spaces are only available for observation space. Observation space is as follows :

- **Portfolio value**: The current market value of the portfolio.
- **In-view liabilities**: 2D array. Each row details an upcoming liability: the first column indicates the number of months remaining until its maturity, and the second column its value. If fewer active liabilities exist than the *liabilities_window_size*, the array is padded with zeros to respect observation space dimensions.
- $\boldsymbol{R_{kj}}$, $\boldsymbol{\cos(\vartheta_{kmj})}$ **and** $\boldsymbol{\sin(\vartheta_{kmj})}$

This observation structure provides the agent with a comprehensive view of its financial status, future obligations, and the current market risk landscape as characterized by the DPT model.

Episode Initialization (reset) : The reset method prepares the environment for a new episode. It resets internal variables. Simultaneously, a new, randomized schedule of liabilities is generated with time and amount constraints. The total nominal value of these liabilities is targeted to be a certain percentage above the initial portfolio value, simulating a funding goal. These liabilities are then transformed into the array format required for observations.

Environment Dynamics (step method): The step (action) method orchestrates the core interaction loop. Upon receiving an action from the RL agent, the DPT optimization is executed using these parameters and the latest forecasted returns.

A critical feature of this environment is its handling of potentially infeasible DPT solutions. If the DPT solving method fails to find an optimal portfolio (e.g., due to overly restrictive parameters chosen by the agent), a negative reward, exponential to the number of retries, is issued to penalize the agent for choosing infeasible parameters. If the number of retries surpasses a specific threshold (set to 8), the episode is truncated, forcing the agent to learn to avoid such parameter

combinations. During retries the environment does not advance its internal state (e.g., time, portfolio value), allowing the agent to attempt a different action in the same state.

If the DPT optimization is successful, the environment transitions:

1. The returns slider advances, providing the next window of historical returns. The DPT signals are then recalculated using the historical portion of this *new* window. These signals will form part of the *next* state's observation.
2. Any liabilities that have matured (i.e., their month remaining count has reached zero in the previous step) are settled by adjusting the portfolio value.
3. The portfolio value is then updated based on the arithmetic return achieved by the DPT portfolio in the current step.
4. The months remaining field of all outstanding liabilities is decremented by one.

Following these state updates, the reward for the successful action is computed. The episode terminates if the portfolio value drops to zero or below, or if all liabilities have been successfully met.

Reward Function Engineering : The reward function is meticulously designed to guide the RL agent's learning process. It is a scalar value that reflects the desirability of the agent's actions in the context of both portfolio performance and liability management. The components of the reward signal in DPTEnv are:

- **Return Maximization:** A significant positive component is directly proportional to the achieved portfolio return encouraging the agent to select DPT parameters that lead to higher returns.
- **Liability-Driven Objective:** Calculates the sum of the present values of all remaining liabilities. A discount factor is defined as constant and applied to all liabilities, following the Markov Decision Process. A higher portfolio value relative to these discounted future obligations results in a higher reward.

- **Portfolio Value Preservation:** A tiered bonus/penalty system based on the ratio of the current portfolio value to its starting amount incentivizes capital preservation and growth. Substantial penalties are applied if the portfolio value drops significantly (e.g., -50 if below 50% of the start), while bonuses are awarded for strong growth (e.g., +40 if above 140% of the start).
- **Timeliness of Liability Settlement:** A penalty is applied based on the time remaining until the furthest liability matures, encouraging the agent to survive until maturity of the last liability
- **Risk Parameter Sophistication:** To encourage intelligent control over the DPT risk parameters, the reward includes terms related to the dispersion and magnitude of the chosen $C\beta_k$ and $C\alpha_k$ values. A bonus is given for higher standard deviations of these parameters, promoting active differentiation of risk constraints across different time horizons. Conversely, penalties are applied to the sum of these parameters, discouraging the agent from selecting unnecessarily high-risk constraints.

This multi-faceted reward function aims to train an agent that not only seeks high returns but also prudently manages risk and liabilities over the investment horizon.

TD3 Agent and Training process

The TD3 algorithm, provided by the *Stable-Baselines3* library, was chosen for this task as specifically designed for environments with continuous action spaces.

Given that the Environment observation space is a dictionary, a *MultiInputPolicy* network architecture is employed. Following the recommendations of *Stable-Baselines3*, the default hyperparameters are retained.

A custom *TensorboardCallBack* facilitates the logging of training progress and custom environment metrics to *TensorBoard*, enabling visualization and monitoring of the learning process. The trained model is saved periodically, allowing for the resumption of training or for later evaluation.

Explanation of population

This chapter details the methodology employed for selecting the market, asset universe, and specific data used in the empirical application of the DPT framework within this thesis. The choices made were influenced by the theoretical underpinnings of DPT, practical considerations of data acquisition, and the desire to test the model in a relevant and dynamic market environment.

Context from Original DPT Implementations

In the foundational implementations of the DPT framework, C. Kenneth Jones tested his model across a diverse range of markets and asset classes. These included broad market indices like the Dow Jones Industrial Average, various specific sector indices (e.g., property, automobile, hardware), and different commodities such as non-alcoholic beverages, tobacco, oil, electricity, and gas (Jones, 2009). This broad testing demonstrated the potential applicability of DPT across different financial domains.

Rationale for Market and Asset Class Selection in This Study

For the empirical analysis conducted in this thesis, the scope was deliberately focused on the United States equity market. This decision was primarily driven by a key characteristic of the DPT framework: its emphasis on mean-reversion cycles, particularly a 48-month cycle. As indicated in the DPT literature, this cycle is defined in concordance with the United States presidential election cycle, which occurs every four years. To align the empirical test as closely as possible with this theoretical consideration and to provide a clear context for interpreting the framework's behavior, the U.S. market was deemed the most suitable choice.

While Jones's work explored various asset classes, this study restricts its analysis to equities only. This limitation was introduced for reasons of simplicity in model implementation and due to the inherent complexities associated with sourcing, cleaning, and harmonizing high-quality historical data across multiple disparate asset classes (e.g., commodities, bonds, real estate) with the requisite granularity and length.

Data Requirements and Sourcing

The DPT framework, particularly for its signal processing component using Welch method, necessitates a substantial history of financial data. As indicated by Jones (2001), the methodology typically requires at least 16 years of monthly data to reliably estimate the spectral components and phase relationships crucial for risk decomposition.

To meet this requirement, historical financial data was sourced via the EOD Historical Data (EODHD) provider.

The primary focus for asset selection was the S&P 500 index, which is the leading benchmark for large-capitalization U.S. equities. The portfolio simulation period for this study was set to start on January 31, 2021. To obtain the necessary 16 years of historical monthly data preceding any DPT calculations relevant to this start date, data was retrieved extending back to approximately the end of 2004. This historical window is particularly rich in terms of market dynamics, encompassing significant financial events such as the 2008 subprime mortgage crisis, the European sovereign debt crisis around 2011, and the market volatility induced by the COVID-19 pandemic in 2020. The inclusion of these diverse market regimes provides a robust testing ground for the DPT model and the adaptive capabilities of the RL-driven Decision Module. The forward-looking simulation period for the trading algorithm thus spans from early 2021 to the end of 2024, covering a distinct four-year cycle to analyze.

Population Construction and Survivorship Bias Mitigation

A critical consideration in constructing the investment universe from an index like the S&P 500 is the potential for survivorship bias. The composition of the S&P 500 changes frequently due to corporate actions such as mergers, acquisitions, bankruptcies, and companies meeting or failing to meet inclusion criteria. Using only the constituents of the S&P 500 as of the end of the study period would exclude companies that were part of the index earlier but were subsequently delisted, leading to an overly optimistic assessment of historical performance.

To mitigate this bias, the initial population for this study was defined as the constituents of the S&P 500 index as of January 31, 2021. This fixed list of companies forms the basis of the eligible investment universe. For simplifying the simulation environment and focusing on the DPT and RL

aspects, the composition of this S&P 500 cohort was kept static throughout the 2021-2024 simulation period, even though the actual S&P 500 index would have undergone changes during this time.

Data Cleaning and Exclusion Criteria

On January 31, 2021, the S&P 500 comprised 508 individual companies. A rigorous data cleaning and validation process was applied to ensure the quality and suitability of the historical data for each of these potential constituents. This process led to the exclusion of 113 companies, based on the following strict criteria:

1. **Insufficient Historical Data Length:** Companies for which less than 16 years of continuous monthly historical price data (i.e., data extending back to at least end of 2004 for a January 2021 start) could not be retrieved were excluded. This is a direct consequence of the DPT methodology's requirement for long time series and its major weakness.
2. **Complete Data Unavailability:** Some tickers were not available through the EODHD service for the full 16-year period or change in ticker name, complexifying the trace. Such companies were removed from the sample.
3. **Partial Data Absence (Missing Data Tolerance):** To maintain the integrity of the time series analysis, a stringent tolerance for missing data was applied. Any company exhibiting more than 1% missing monthly data points over the 16-year historical period (meaning 2or more missing data points) was excluded.

The decision to apply such strict exclusion criteria, rather than employing data imputation techniques like backfilling for missing observations, was made to preserve the precision of the DPT signals. While backfilling might have allowed for a larger sample size, it could also introduce artificial patterns or dampen the true underlying cyclical components that DPT aims to capture. The priority was thus placed on data quality and the fidelity of the spectral analysis.

Following this filtering process, the final investment universe for the empirical study consisted of 395 companies. This refined panel provides a robust and high-quality dataset for testing the DPT

framework and the associated RL-driven decision-making module. The full list of selected and non-selected tickers can be found in Annex A.

Results

Fourier transform

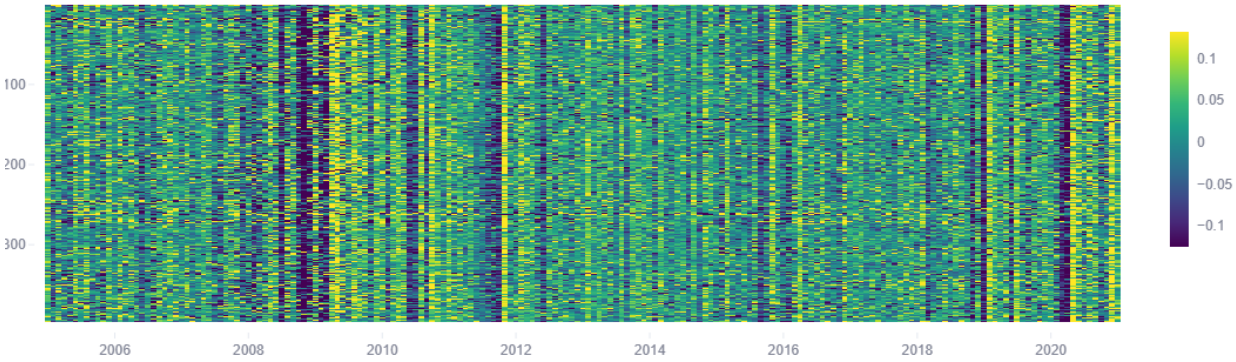
Empirical Analysis of DPT Inputs

This chapter presents an empirical analysis of the key inputs to the DPT framework. The initial step involves the transformation of historical asset prices into logarithmic returns. Subsequently, these returns are processed using Welch's method to derive their Power Spectral Density (PSD), which reveals the distribution of variance across different harmonic frequencies. Finally, an analysis of coherence between individual assets and the market benchmark (S&P 500) across these harmonics is discussed. These analyses provide insights into the risk characteristics of the selected U.S. equity universe over the historical period of 2004-2021.

Analysis of Logarithmic Returns

The foundational data for DPT signal processing are the logarithmic returns of the 395 selected companies from the S&P 500. The visual representation of these returns over the period spanning from late 2004 to early 2021 highlight periods of significant market stress and volatility. Th heatmap clearly demarcates historical financial crises, including the subprime mortgage crisis (2008-2009), the sovereign debt crisis (2011), and the COVID-19 pandemic market shock (2020). These events are characterized by pronounced negative returns across a broad range of assets. Conversely, periods of strong market recovery, such as the rebound following the subprime crisis, are also evident, with some assets exhibiting exceptionally high monthly logarithmic returns, reportedly exceeding 10% of monthly returns in certain instances during that recovery phase. This historical context, rich with diverse market regimes, forms the basis for the subsequent spectral analysis.

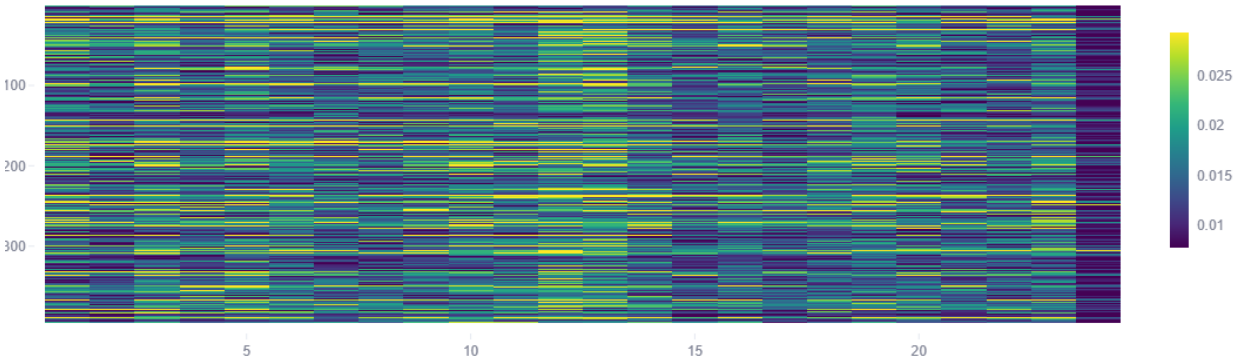
Figure 2 Log returns of Selected S&P 500 Components (Monthly data)



Power Spectral Density (PSD) Analysis

The logarithmic return series for each of the 395 companies were processed using Welch's method, to obtain their Power Spectral Density. The resulting PSD plots, when displayed in the heatmap across all assets, reveal common cyclical risk patterns within the market.

Figure 3 Power Spectral Density of the S&P 500 Components



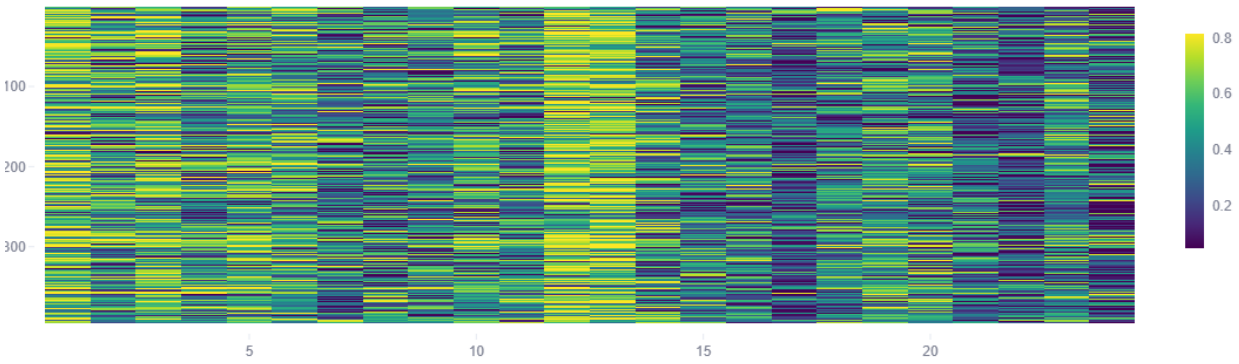
- Prominent power at harmonics 12 and 13 (periods of 4 and 3.7 months), suggesting significant variance is driven by short-term cycles, with harmonic 12 tied to calendar-related risks.
- Lower frequency harmonics (1–5) show alternating patterns of spectral density:

  - Harmonic 1 (48 months) reflects the 4-year U.S. presidential cycle and shows a slight increase in risk.
  - Harmonic 2 (24 months), tied to midterm elections, appears less influential.
  - Harmonics 3 and 5 indicate intermediate cycles without direct calendar associations.
- Harmonic 10 (4.8 months) relates to an 8-year business cycle component.
- High-frequency harmonics generally show diminishing variance contribution, with harmonic 24 (2 months) linked to minor very low short-term calendar effects.

Coherence Analysis

Although the coherence values themselves are not directly used as inputs in the DPT optimization problem formulated in this thesis, analyzing the coherence between each asset and the market benchmark (S&P 500 index, ticker GSPC) provides valuable insights. The Coherence measures the degree of linear relationship between asset *j* and the market index *m* at each harmonic *k*.

Figure 4 Coherence Between S&P 500 Components and the Reference Index



The analysis reportedly reveals very significant correlations (high coherence) at harmonics 12 and 13, as well as for harmonic 3. This observation is consistent with the PSD analysis, where these harmonics also exhibited notable power concentrations. High coherence at specific harmonics indicates that the cyclical behaviors observed in individual assets at these frequencies are strongly

synchronized with the broader market's cyclical behavior at those same frequencies. This reinforces the idea that the risks associated with the 4-month cycle (harmonic 12), the 3.7-month cycle (harmonic 13), and the 16-month cycle (harmonic 3) are, to a significant extent, systematic in nature for the analyzed S&P 500 constituents.

Forecasting

Figure 5 One-month Forecast S&P 500 Components Using a 48-Month Moving Average

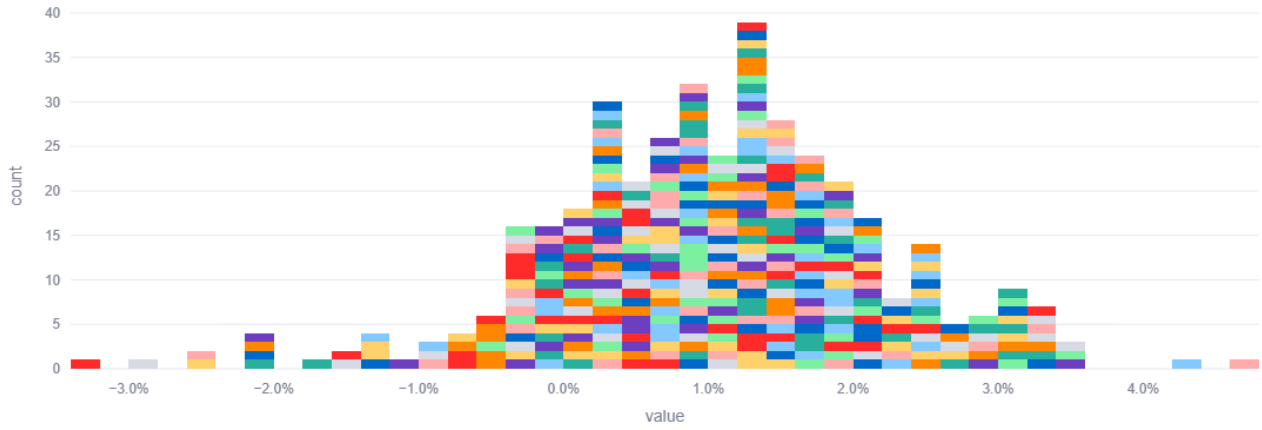


Figure 6 Treemap One-month Forecast S&P 500 Components Using a 48-Month Moving Average

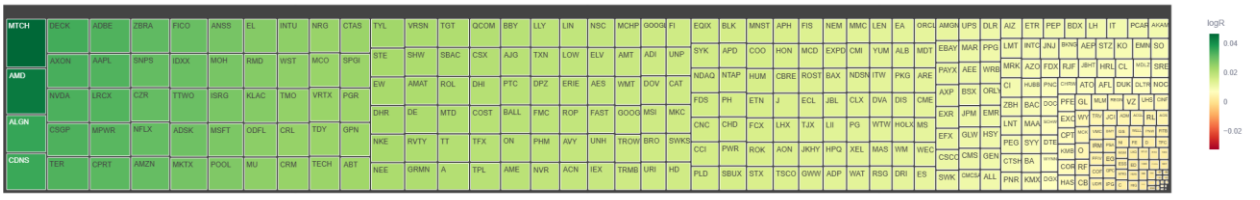


Figure 7 One-month Forecast S&P 500 Components Using Auto-ARIMA

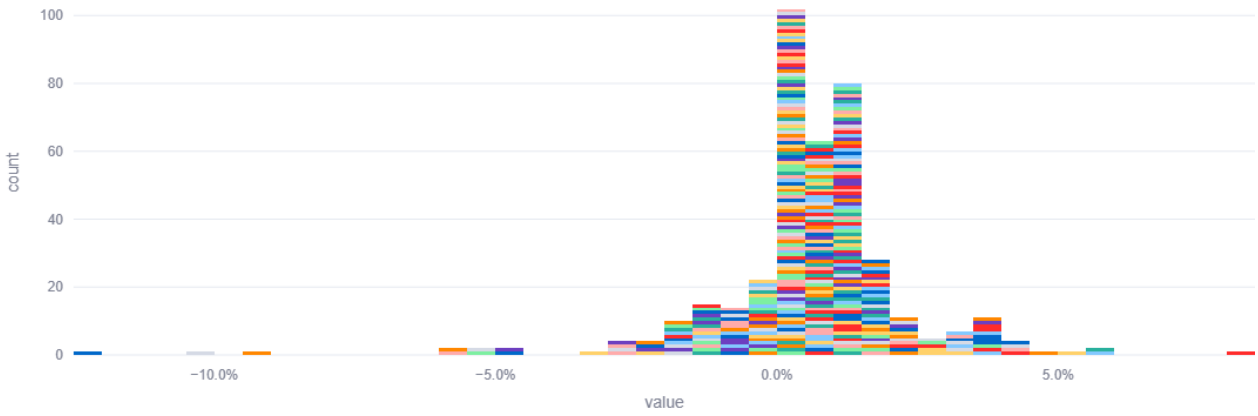
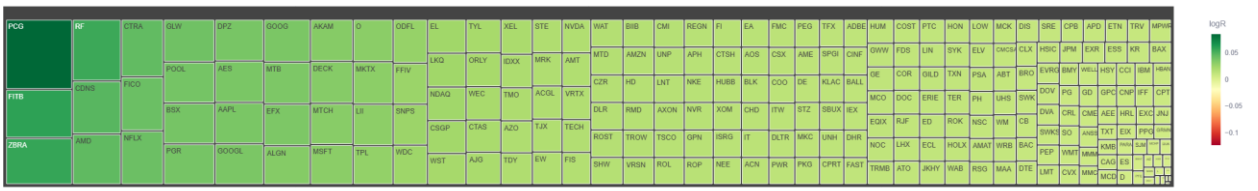


Figure 8 Treemap One-month Forecast S&P 500 Components Using a 48-Month Moving Average



The one-month-ahead return predictions derived from a 48-month forecast moving average and those generated by Auto-ARIMA models yield markedly different statistical distributions. Distribution of moving average returns exhibits low dispersion around the mean, with small Z-scores and limited variability, reflecting a stable and smooth signal (low-kurtosis profile where extreme values are rare).

Conversely, Distribution of Auto-ARIMA forecasts return, displays a leptokurtic structure characterized by a sharp central peak and heavy tails. From a practical point of view, this distribution is more realistic and therefore will be used in the latter calculation. Despite these differences, both models consistently forecast high returns for several major technology stocks, notably within the GAFAM and FAANG groups. Apple, NVIDIA, Microsoft, Amazon, Netflix, and AMD.

DPT Solver

Table 3 Configuration of DPT Parameters and Corresponding Optimized Portfolio

id	1		2		3		4	
Forecasting method	**Moving average 48 months**		**Auto-ARIMA**		**Auto-ARIMA**		**Auto-ARIMA**	
Min Invest.	3%		3%		5%		5%	
Max Invest.	15%		15%		20%		10%	
Ptf. size	15		15		10		15	
Betas	0.5		0.5		$*C\beta_k$		$**\ C\beta_k$	
Alphas	0.35		0.35		0.4		$*C\alpha_k$	
	Stock	**Weight**	**Stock**	**Weight**	**Stock**	**Weight**	**Stock**	**Weight**
	GIS	15%	GIS	15%	WEC	17.3%	PCG	10%
	ED	15%	ED	14%	ED	15.1%	XEL	8.97%
	SO	11.5%	SO	13.7%	GIS	13.3%	AMT	8.19%
	WMT	10.3%	NEM	8.43%	CHD	11.7%	PGR	7.74%
	NEM	9.38%	CHC	7.32%	FIS	8.58%	MCD	7.52%
	MKC	5.91%	WEC	6.52%	NEM	8.42%	WEC	7.25%
	FIS	5.64%	CAH	5.49%	MCD	8.21%	WMT	7.1%
	ISRG	4.41%	FIS	5.02%	EW	7.34%	ORLY	7.09%
	CAH	4.05%	AKAM	4.08%	CAH	5%	EFX	6.14%
	TJX	3.81%	MKC	3.96%	DGX	5%	CHD	5%
	AKAM	3%	ROL	3.65%			CTRA	5%
	BAX	3%	BAX	3.65%			EW	5%
	CHD	3%	ISRG	3.2%			FICO	5%
	EW	3%	TJX	3%			GIS	5%
	ROL	3%	WMT	3%			NEM	5%
Expected log Return	18.5%		13.6%		11.8%		34.4%	

kth	$*C\beta_k$	$**C\beta_k$	$*C\alpha_k$
1-6	0.40	0.55	0.20
7-18	0.50	0.80	0.50
19-24	0.70	0.55	0.30

Analysis Portfolio 1

The final portfolio includes firms primarily from defensive sectors such as consumer staples, utilities, and health care, with top allocations to General Mills (GIS), Consolidated Edison (ED), and Southern Company (SO). The sectoral composition reflects a preference for stability, while the inclusion of firms like Akamai (technology) and Newmont (materials) ensures diversification.

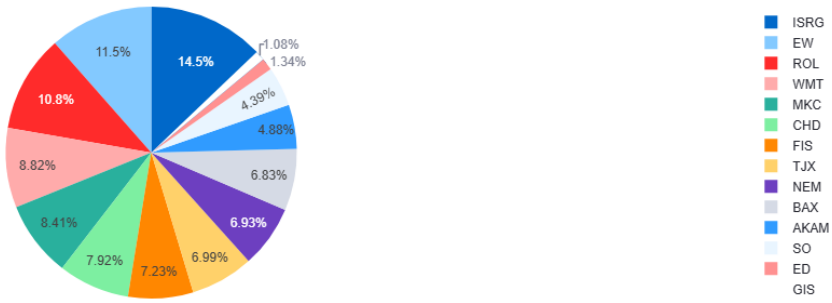
Table 4 Portfolio 1: $C\beta_k$ and $C\alpha_k$

kth	Ptf beta	Ptf alpha	kth	Ptf beta	Ptf alpha
1	0.50	-0.07	13	0.50	-0.10
2	0.48	0.32	14	0.48	-0.31
3	0.50	0.02	15	0.50	-0.12
4	0.29	0.00	16	0.40	-0.27
5	0.50	0.18	17	0.48	0.03
6	0.48	-0.25	18	0.50	0.12
7	0.43	-0.30	19	0.47	-0.09
8	0.50	-0.10	20	0.45	-0.06
9	0.50	-0.02	21	0.41	-0.33
10	0.50	0.08	22	0.27	-0.02
11	0.18	-0.35	23	0.46	0.33
12	0.50	0.01	24	0.49	0.00

Risk decomposition confirms that portfolio betas are tightly clustered around the 0.5 target, indicating stable systematic exposure. In contrast, alpha values range from -0.35 to +0.33, showing selective exposure to non-calendar anomalies. This asymmetry highlights DPT's ability to avoid certain high-frequency risks while exploiting others for return enhancement.

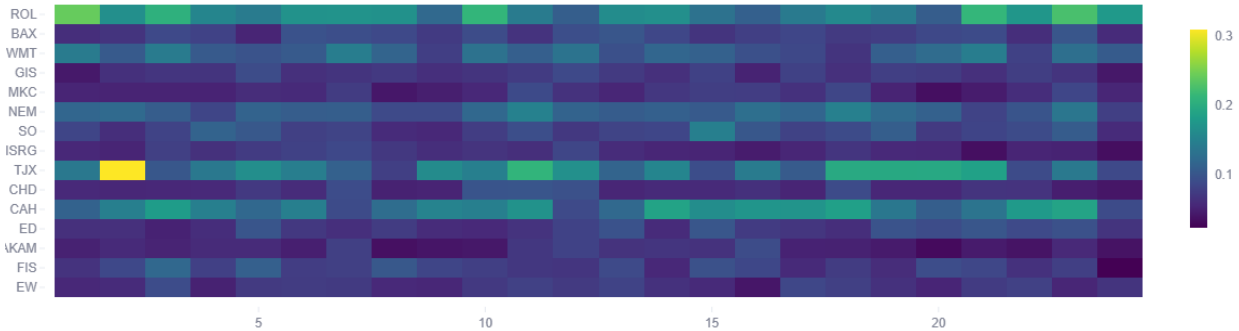
In constructing this portfolio, the algorithm deliberately selected assets whose alphas and betas exhibit low correlation, thereby enhancing diversification across different risk dimensions. A notable example is TJX, which stands out due to its significant power spectral density (PSD) component. Despite its modest allocation of 3.81%, TJX contributes an unweighted return component of 6.99%. This highlights a key characteristic of the portfolio: some assets offer attractive standalone returns, but their weights are moderated to respect overall risk constraints.

Figure 9 Portfolio 1: Unweighted returns



Unweighted return refers to the portion of the portfolio's total return attributable to a given asset if the portfolio were equally weighted across all holdings. This measure allows us to isolate the intrinsic return contribution of each asset independent of its actual portfolio weight, providing deeper insight into how the optimization balances return potential with risk exposure.

Figure 10 Portfolio 1: Power Spectral Density



A similar case can be seen with ROL, which delivers a notably high unweighted return of 10.8%, yet is assigned a low portfolio weight of 3%. These assets are returns boosters, and therefore in small ponderation.

Overall, this portfolio suits a sophisticated investor with moderate risk tolerance and a preference for exploiting structural inefficiencies. By integrating spectral analysis, DPT provides a richer framework for diversification and return optimization beyond traditional time-domain models.

Analysis Portfolio 2

The portfolio constructed using the Auto-ARIMA forecasting method maintains similar risk and investment constraints as the previous one, with position limits set between 3% and 15%, a total portfolio size of 15 assets, and target spectral exposures of beta = 0.5 and alpha = 0.35. This approach leverages the Auto-ARIMA model's adaptive ability to capture evolving time-series dynamics and structural breaks, providing a potentially more responsive signal than the moving average method.

The resulting portfolio reflects a diversified selection of 15 stocks across multiple sectors. General Mills (GIS) and Consolidated Edison (ED) again hold the largest allocations of 15% and 14%, respectively, followed closely by Southern Company (SO) at 13.7%. This sectoral distribution continues to favor stable, defensive industries while incorporating a balanced exposure to growth-oriented sectors.

The expected annualized log return of the portfolio is 13.6%, slightly lower than the previous portfolio based on a moving average forecast. This difference could be explained by the model's increased sensitivity to recent data, potentially leading to more conservative asset weightings under the same risk constraints.

Table 5 Portfolio 2: $C\beta_k$ and $C\alpha_k$

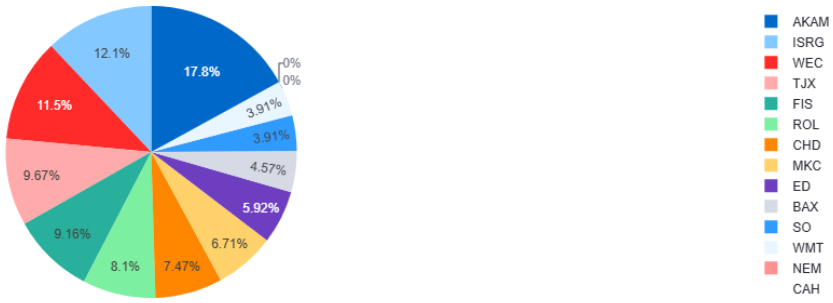
kth	Ptf beta	Ptf alpha	kth	Ptf beta	Ptf alpha
1	0.50	-0.14	13	0.50	-0.04
2	0.50	0.30	14	0.45	-0.28
3	0.48	0.01	15	0.50	-0.10
4	0.29	0.04	16	0.40	-0.27

5	0.45	0.32	17	0.35	0.14
6	0.44	-0.25	18	0.50	0.10
7	0.43	-0.33	19	0.50	-0.05
8	0.50	-0.08	20	0.41	0.02
9	0.50	0.08	21	0.33	-0.32
10	0.43	0.06	22	0.34	-0.08
11	0.09	-0.35	23	0.44	0.35
12	0.50	0.01	24	0.44	0.00

Risk decomposition across the 24 spectral components reveals that portfolio betas consistently cluster around the target 0.5 level, confirming adherence to systematic risk targets across frequencies. Several spectral components show negative alphas, suggesting cautious avoidance of specific frequency-domain anomalies, while positive alphas on other components highlight selective exploitation of recurring market inefficiencies. This nuanced balance between spectral alpha and beta exposures demonstrates the flexibility of the DPT framework to accommodate diverse signal extraction methods while maintaining robust risk control.

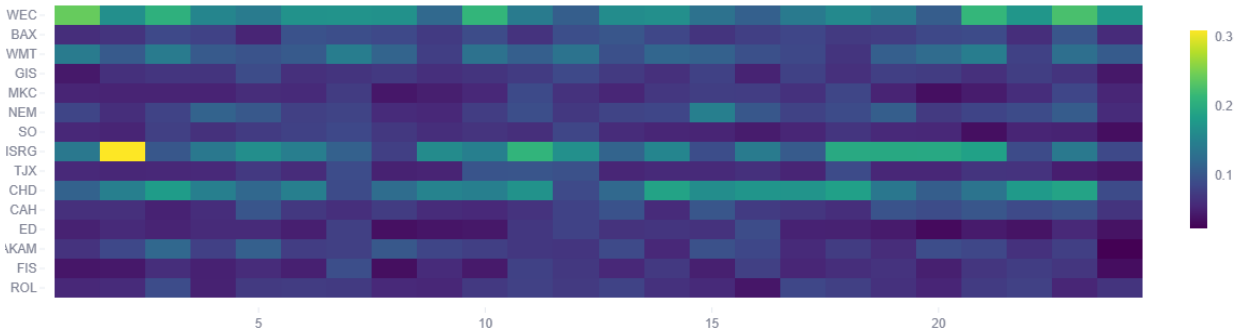
This portfolio is appropriate for investors or institutions valuing adaptive forecasting techniques integrated with frequency-aware risk decomposition to achieve efficient, signal-driven asset allocation.

Figure 11 Portfolio 2: Unweighted returns



Once again, the riskiest assets, such as WEC and ISRG, are carefully weighted despite their high expected returns.

Figure 12 Portfolio 2: Power Spectral Density



Analysis Portfolio 3

The third portfolio was also built with Auto-ARIMA forecasts, but with adjusted constraints: a smaller portfolio size of 10 assets, wider investment bounds between 5% and 20%, and varying target betas across spectral components. This design reflects a more tailored risk management approach, recognizing that different frequency bands contribute distinctively to portfolio risk.

The selected assets come from defensive and stable sectors, including utilities (WEC, ED), consumer staples (GIS, CHD, MCD), health care (CAH, DGX), and industrials (FIS, EW, NEM). Weights are concentrated but remain within constraints, with the largest allocation at 17.3% (WEC) and the smallest at 5% (CAH, DGX). The portfolio's expected annualized log return is 11.8%, reflecting a conservative tilt consistent with the higher beta targets on longer-term spectral components.

Figure 13 Portfolio 3: Unweighted returns

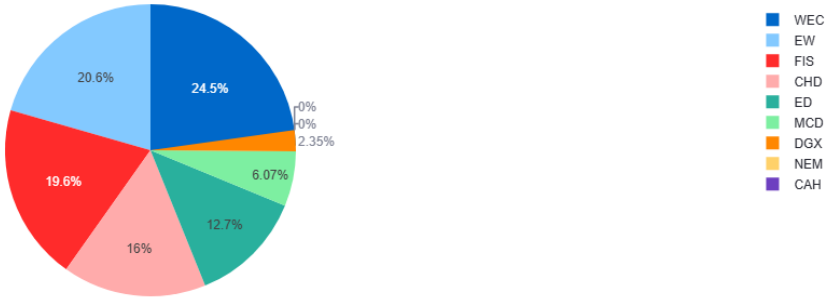
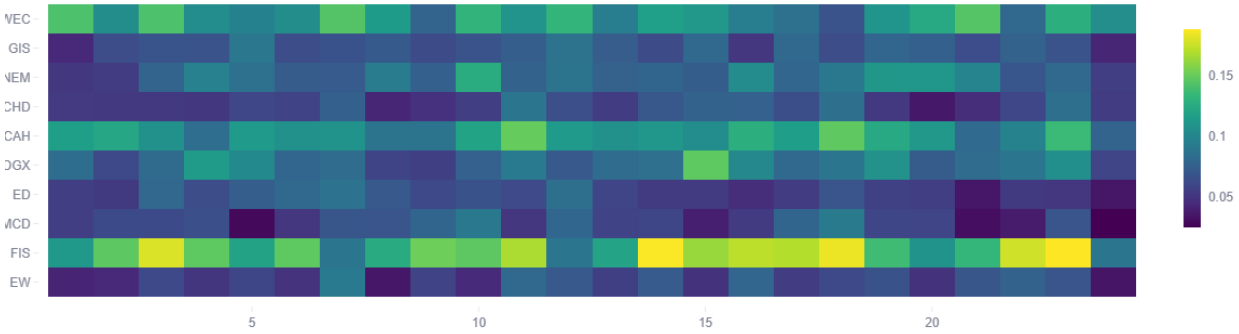


Figure 14 Portfolio 3: Power Spectral Density



Risk decomposition confirms adherence to the prescribed beta structure: betas vary as intended across frequency bands, from around 0.33–0.40 at lower frequencies to 0.70 at higher frequencies, indicating greater systematic exposure to longer-term market trends. Notably, several components have negative alphas, reflecting risk avoidance, while positive alphas highlight opportunities to exploit market inefficiencies in specific spectral bands. Fidelity National Information Services (FIS) here present a large Power Spectral Density on most of the 24th harmonics, highlighting the direct increase in risk in this portfolio.

This portfolio is suited for investors willing to accept moderate systematic risk that increases with investment horizon, emphasizing robustness against short-term volatility while capitalizing on longer-term trends.

Table 6 Portfolio 3: $C\boldsymbol{\beta}_k$ and $C\boldsymbol{\alpha}_k$

kth	Ptf beta	Ptf alpha	kth	Ptf beta	Ptf alpha
1	0.40	-0.15	13	0.50	-0.07
2	0.35	0.27	14	0.50	-0.15
3	0.40	0.04	15	0.50	-0.22
4	0.36	0.04	16	0.45	-0.29
5	0.39	0.33	17	0.19	0.05
6	0.33	-0.29	18	0.60	0.36
7	0.48	-0.39	19	0.56	-0.08
8	0.47	0.12	20	0.31	-0.05
9	0.50	0.19	21	0.35	-0.17
10	0.29	0.10	22	0.49	-0.20
11	0.10	-0.39	23	0.50	0.40

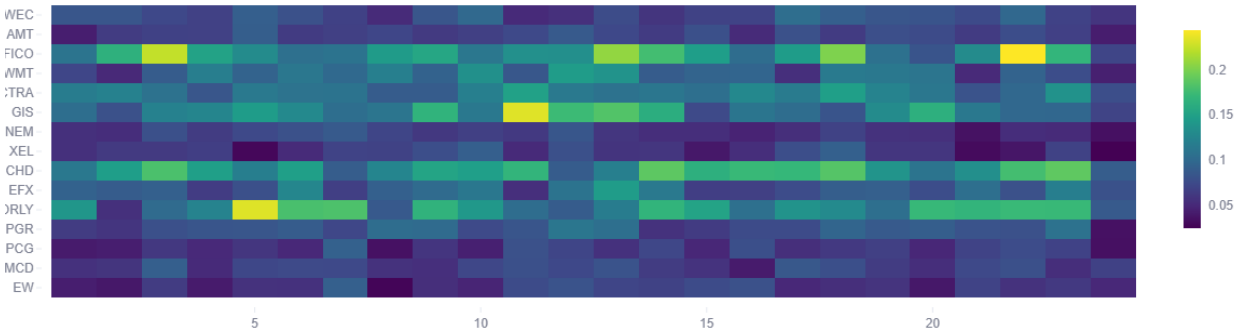
12	0.50	0.12	24	0.46	0.00

Analysis Portfolio 4

This last portfolio was constructed using Auto-ARIMA forecasts under a more complex constraint regime that varies both spectral betas and alphas across calendar frequencies.

The resulting portfolio combines utility stocks (PCG, XEL, WEC), consumer defensives (GIS, MCD, WMT, CHD), and growth-oriented picks (FICO, AMT, ORLY), yielding an expected annualized log return of 34.4%—the highest among tested configurations. This extreme performance stems from concentrated yet constraint-respecting allocations, especially in stocks with strong mid-frequency alpha contributions, which benefit from the higher alpha targets imposed in this spectral band.

Figure 15 Portfolio 4: Power Spectral Density



The assets' Power Spectral Density is the most colored in the portfolio, due to relaxed constraints on betas and alphas.

Table 7 Portfolio 4: $C\beta_k$ and $C\alpha_k$

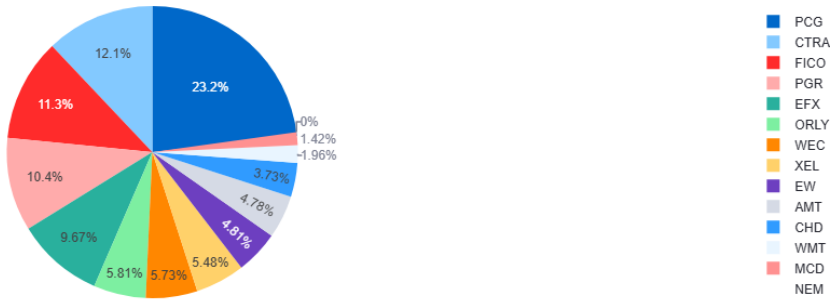
kth	Ptf beta	Ptf alpha	kth	Ptf beta	Ptf alpha
1	0.55	-0.17	13	0.80	-0.10
2	0.34	0.16	14	0.76	-0.16
3	0.55	-0.08	15	0.65	-0.07
4	0.52	-0.10	16	0.46	-0.24
5	0.47	0.20	17	0.50	0.01

6	0.55	-0.05	18	0.58	0.18
7	0.66	-0.19	19	0.54	0.05
8	0.62	0.18	20	0.41	-0.07
9	0.78	0.05	21	0.42	-0.07
10	0.64	0.10	22	0.29	0.07
11	0.36	-0.32	23	0.55	0.30
12	0.77	-0.01	24	0.40	0.00

Risk decomposition confirms close alignment with the constraint structure. Low-frequency components exhibit betas near 0.55 and modest alphas, while mid-frequency bands show stronger market exposure (betas around 0.8) and a mix of positive and negative alpha contributions. This spectral tuning suggests the model is capturing both short-term resilience and mid-horizon alpha potential while limiting overexposure to high-frequency noise.

Logically, PG&E Corp (PCG) is the most heavily weighted asset, contributing over 23.2% to unweighted returns despite having a low power spectral density

Figure 16 Portfolio 4: Unweighted returns



This portfolio highlights the flexibility of spectral optimization: by varying constraints across calendar frequencies, it is possible to fine-tune exposure to different market dynamics and target alpha generation with temporal precision.

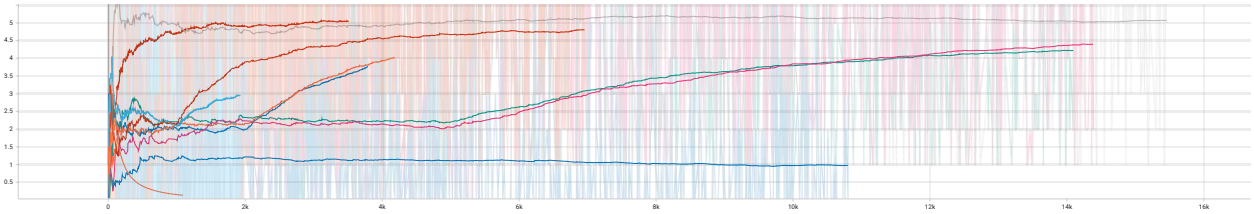
TD3-Driven DPT Framework Control

Algorithm names	
TD3_V01	TD3_V06
TD3_V02	TD3_V07
TD3_V03	TD3_V08
TD3_V04	PPO_V01
TD3_V05	PPO_V02

The development of the reinforcement learning environment for agent-market interaction underwent several iterative refinements. Initially, the observation space was limited to the discounted future value of liabilities. Subsequently, this was expanded to incorporate the sine and cosine of theta of return signals for each constituent of the S&P 500 index.

Various reward functions, characterized by differing coefficient values, were evaluated. However, the average reward metric failed to achieve stable convergence. To ensure proper episode execution by the learning algorithm, several key indicators were monitored. Notably, the *Retry Count* tracked instances where the agent's proposed actions failed to yield feasible portfolios due to solver constraints. This metric exhibited erratic increases across different algorithm trials, displaying an overall upward trajectory.

Figure 17 Metric Retry Count



Standardization of Alpha and Beta inputs to the model was measured to detect any upward bias stemming from the assigned rewards. Disappointingly, after a certain learning time, most algorithms converged to singular values for these inputs and ceased to exhibit further variation. This suggests a need for further refinement of the reward function to achieve a stable trade-off between exploration and exploitation.

Figure 18 Metric Beta Standard Deviation

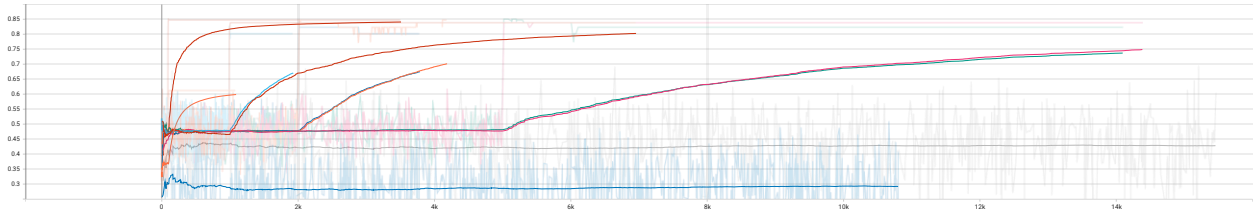


Figure 19 Metric Beta sum

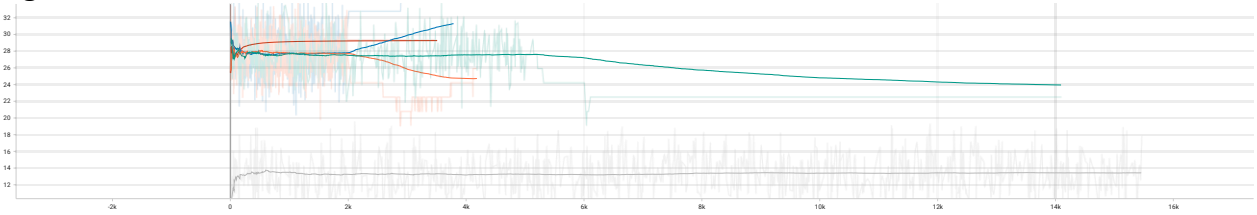


Figure 20 Metric Alpha Standard Deviation

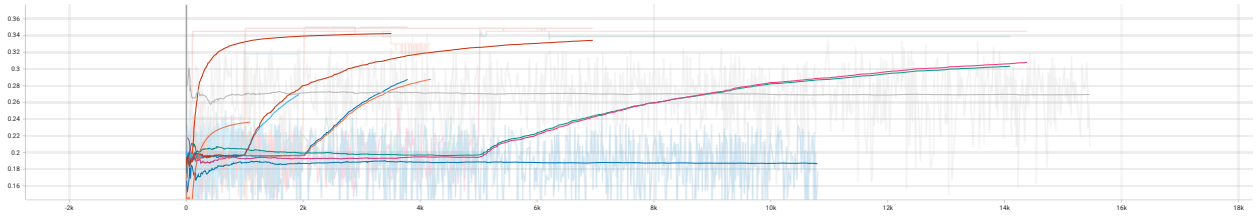
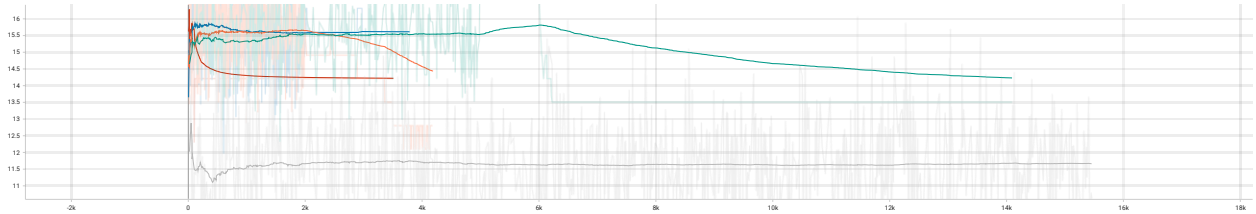
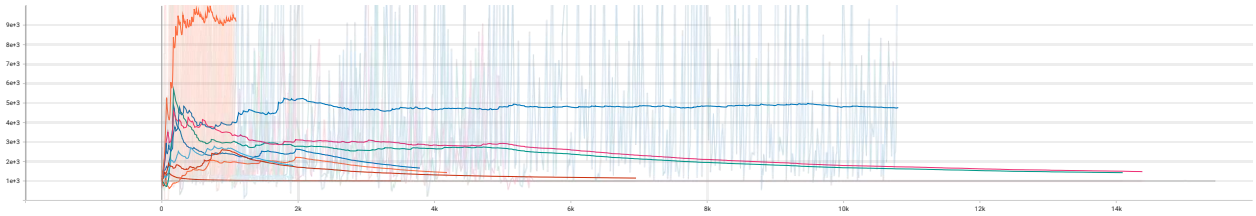


Figure 21 Metric Alpha sum



The *Portfolio Value* was also tracked, frequently demonstrating an upward trend. This was anticipated due to the incorporation of actual historical data within the training set, enabling the agent to know the future one month ahead and so achieve substantial returns. But still, the graphs reveal a substantial increase in portfolio value for most algorithms, reaching up to 900% for the V1 variant, with an average of 300% for other algorithms within the initial few steps, followed by stagnation for the remainder of the training period. This outcome warrants further investigation. The anticipated results would be a consistently increasing curve over the steps, potentially exhibiting sawtooth declines corresponding to the realization of future liabilities, which were not consistently observed.

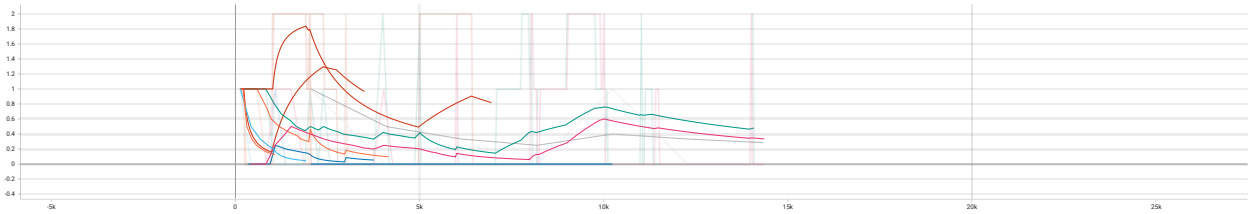
Figure 22 Metric Portfolio Value



This deliberate technical choice to use future returns as inputs, rather than estimated forecasts, was intended to segment the learning process by first focusing on familiarizing the model with the framework's mechanics. The subsequent intention was to transition utilizing previously generated return forecasts (e.g., exponential smoothing, LSTM, moving average, ARIMA) following a sufficient training period and data convergence. This phased approach was deemed prudent given the inherent complexity of the framework, where task decomposition and input simplification can be critical during the initial learning stages.

A significant constraint and challenge encountered during model training was the computational cost associated with each step. Depending on the complexity of the solver parameters, particularly for the *PuLP* solver which could require up to 30 seconds for convergence, a single step could range from two seconds to several minutes.

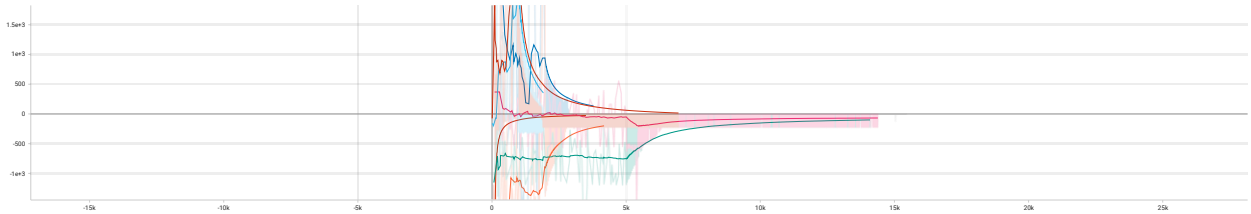
Figure 23 Metric Fps Step Per Second



Furthermore, a considerable number of invalid input proposals impeded convergence towards optimal portfolio allocation. Consequently, the estimated duration for a single step was approximately 10 seconds. Given this, achieving adequate model training necessitated hundreds of thousands of iterations on high-performance servers equipped with substantial CPU processing power. The model was trained on an OVH AI Notebook instance with 13 CPU vCores, an NVIDIA Tesla V100S GPU, and 32 GB of RAM. Approximately ten models were trained for a period of between 2 to 20 hours. Beyond this timeframe, further training proved unproductive as the models entered a state of consistently selecting identical alpha and beta inputs.

The most recently trained model, utilizing the previously described *Reward Function*, demonstrated a notable increase in its average reward value. Commencing from a highly negative value or positive value, they all converged towards zero. This demonstrated anissue with the reward function, further improvement on adding complexity as well as tweaking reward coefficient is needed.

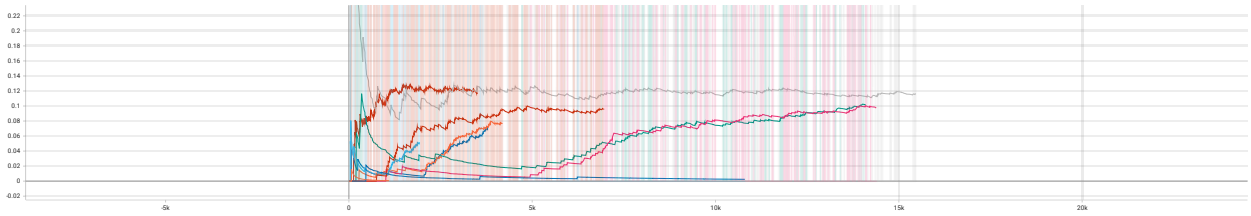
Figure 24 Metric Reward Function



The *Truncated* parameter signifies the premature termination of an episode due to non-convergence or invalid parameter inputs for the DPT solver. This parameter is an incremental

counter that increases by one at each step if the actions proposed by the model do not permit the creation of a valid portfolio. When this counter reaches the limit value (fixed at 8 in the code), the episode is terminated, and a malus is given.

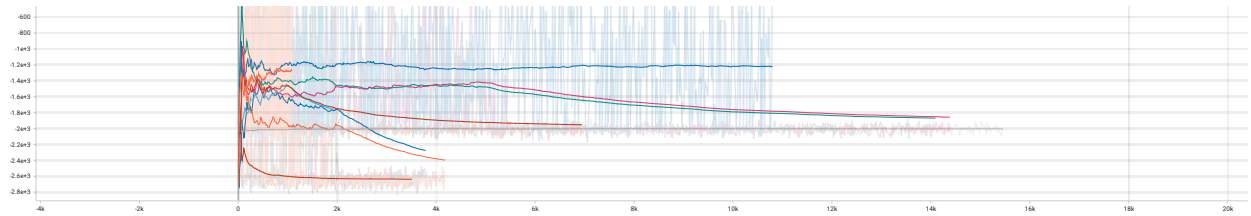
Figure 25 Metric Truncated



The corresponding graph reveals a trend towards closure across all algorithms, including the TD3_V5 variant which initially showed a decrease before subsequently increasing. Ideally, this parameter should converge towards zero, indicating a problem in the learning process.

The *Discounted Liabilities Sum* graph illustrates the evolution of the discounted future liability values at each time step. These values were largely stagnant for most algorithms, with a downward trend, which is consistent. Some algorithms exhibited sharp declines in the average discounted liability value over time, although the underlying explanation remains unclear. Ideally, this value should remain stable over time.

Figure 26 Metric Discounted Liabilities Sum



The *Next Liabilities Amount* graph tracks the evolution of the liabilities. Its trend is highly consistent, with all graphs showing an upward trajectory. This is expected as it represents the discounted value of liabilities, which increases over time. The sawtooth pattern observed is a

result of the exit of a liability, the corresponding cash inflow, and the immediate transition to the next liability. This sawtooth pattern is also evident in the *Next liabilities Months* graph.

Figure 27 Metric Next Liability Month

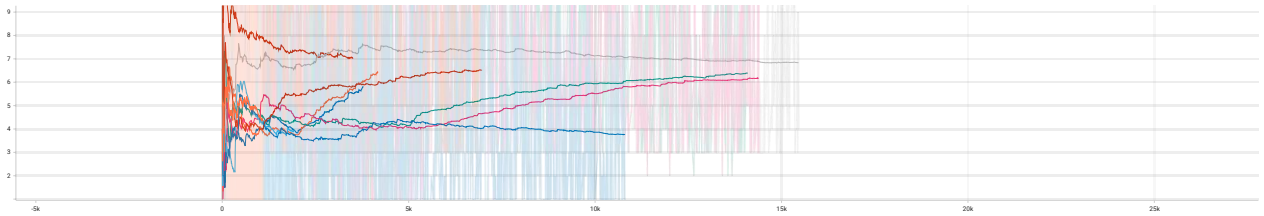
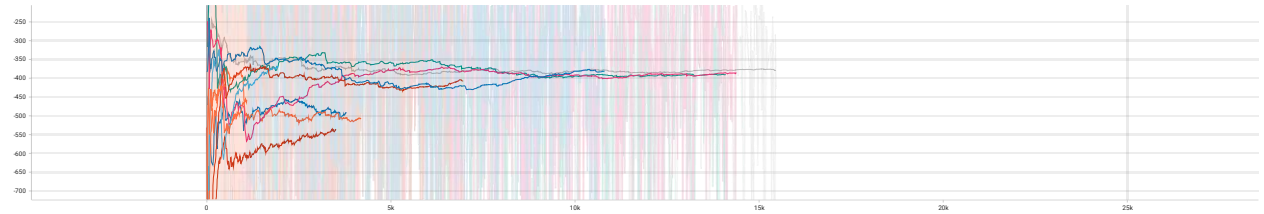
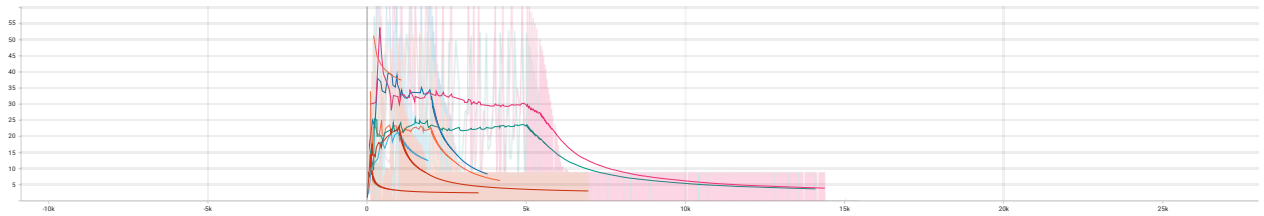


Figure 28 Metric Next Liability Amount



The *Episode Length Min* graph measures the average duration of an episode. This is a native function of Stable Baselines 3 but cannot be considered a reliable indicator of model learning quality in this context. This is because it also accounts for truncated steps, which do not advance the time step but merely represent retries by the model to propose valid parameters for the solver. The observed downward trend is, however, a positive sign, as can be interpreted as a reduction in unsuccessful attempts to parameterize the DPT solver.

Figure 29 Metric Episode Length Min



Conclusion

This research has made significant strides in establishing a novel computational framework for dynamic asset allocation, rooted in the principles of Digital Portfolio Theory (DPT) and enhanced through the integration of reinforcement learning techniques. One of the core achievements of this work lies in the translation of the theoretical foundations laid by C. Kenneth Jones into a fully operational Python-based framework. This implementation enables the practical application of DPT in portfolio optimization by leveraging Fourier Transform techniques to decompose risk across different investment horizons.

To bridge the gap between the DPT framework and reinforcement learning agents, a dedicated environment was developed using the Gymnasium library. This environment serves as a testbed for training and evaluating reinforcement learning strategies in a dynamic, risk-aware asset allocation setting. Its design allows agents to interact with the portfolio optimization logic of DPT, facilitating the learning of allocation policies under time-varying risk conditions.

The implemented DPT framework extends the original formulation in meaningful ways. A scaling factor was introduced into the risk constraint bounds to address the discrepancies between empirical data and the theoretical coefficients proposed by Jones. This adjustment enhances numerical stability and allows for more intuitive manipulation of risk levels by learning agents. Additionally, a maximum holding constraint was incorporated to prevent the solver from assigning excessively large weights to any single asset, thereby encouraging greater diversification and more realistic portfolio configurations.

Looking ahead, there are several promising directions for further research. A natural next step is the development of a parallel implementation of Modern Portfolio Theory (MPT), which would enable empirical comparisons between the two frameworks under varying market conditions and investment horizons. Such a comparative analysis could shed light on the relative strengths and limitations of DPT and MPT, enriching the discourse on risk-aware portfolio construction.

Another important avenue involves refining the reward function used in the reinforcement learning process. Since the learning outcomes of reinforcement learning agents are highly sensitive to the reward structure, a more robust and carefully designed function could significantly enhance policy convergence and agent performance. Relatedly, extending the training duration—potentially supported by more powerful computational infrastructure—might yield better results by allowing agents to explore more complex strategies and develop deeper representations of the environment.

A key limitation of the current work is its reliance on historical return data. Future iterations of this framework should aim to incorporate forecasting modules that allow agents to act on predicted returns, thereby aligning more closely with real-world portfolio management practices. The integration of advanced time series forecasting models—such as Long Short-Term Memory (LSTM) networks or Exponential Smoothing techniques—could further improve the reliability and performance of return predictions, ultimately enhancing the quality of portfolio decisions made by the agent.

In conclusion, this thesis offers a valuable contribution to the computational implementation of Digital Portfolio Theory and opens a pathway toward reinforcement learning-based dynamic asset allocation. The groundwork laid here provides a robust foundation for future developments, with substantial potential for both academic exploration and real-world financial applications.

References

[1] T. A. Wekwete, R. Kufakunesu and G. van Zyl, "Application of deep reinforcement learning in asset liability management," *Intelligent Systems with Applications 20,* 2023.

[2] E. S. Shiu, "On Redington's theory of immunization," in *Insurance: Mathematics and Economics*, vol. 9, 1990, pp. 171-175.

[3] J. Jang and N. Seong, "Deep reinforcement learning for stock portfolio optimization by connection with modern portfolio theory," *Expert Systems with Applications,* vol. 218, 2023.

[4] A. Fontoura, E. Bezerra and D. Haddad, "A Deep Reinforcement Learning Approach to Asset-Liabilities Management," 2019.

[5] Q. Y. E. Lim, Q. Cao and C. Quek, "Dynamic portfolio rebalancing though reinforcement learning," *Neural Computing and Applications,* vol. 34, pp. 7125-7135, 2022.

[6] Y. Ruyu, J. Jiafei and H. Kun, "Reinforcement learning for deep portfolio optimization," *Electronic Research Archive,* 2024.

[7] R. Ashwin and J. Tikhon, Foundations of Reinforcement Learning with Applications in Finance, 2022.

[8] M. L. Puterman, "Markov Decision Processes," in *Handbooks in Operations Research and Management Science*, vol. 2, 1990, pp. 331-434.

[9] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra and M. Riedmiller, "Deterministic Policy Gradient Algorithms," *Proceedings of Machine Learning Research,* vol. 32, pp. 387-395, 2014.

[10] C. Kenneth Jones, "Modern Portfolio Theory, Digital Portfolio Theory and Intertemporal Portfolio Choice," *American Journal of Industrial and Business Management,* vol. 7, pp. 833-854, 2017.

[11] Y. Jiang, J. Olmo and M. Atwi, "Deep reinforcement learning for portfolio selection," *Global Finance Journal,* vol. 62, 2024.

[12] E. J. Elton and M. J. Gruber, "Modern portfolio theory, 1950 to date," *Journal of BANKING & FINANCE,* pp. 1743-1759, 1997.

[13] C. Kenneth Jones, "Digital Portfolio Theory: Portfolio Size versus Alpha, Beta, and Horizon Risk," 2009.

[14] C. Kenneth Jones, "Digital Portfolio Theory," *Computational Economics,* vol. 18, pp. 287-316, 2001.

[15] M. Pinelis and D. Ruppert, "Machine learning portfolio allocation," *The Journal of Finance and Data Science,* vol. 8, pp. 35-54, 2022.

[16] É. BOUYÉ, "Allocation stratégique des actifs et gestion de l'investissement à long terme par les investisseurs institutionnels," *Risque et économie,* vol. 69, pp. 505-531, 2018.

[17] S. Fujimoto, H. Hoof and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *Proceedings of Machine Learning Research,* vol. 80, pp. 1587-1596, 2018.

[18] L.-C. Cheng and J.-S. Sun, "Multiagent-based deep reinforcement learning framework for multi-asset adaptive trading and portfolio management," *Neurocomputing,* vol. 594, 2024.

Annexes

Annex A – S&P 500 Components

(grey cells are excluded from perimeter)

A	AAPL	ABBV	ABNB	ABT	ACGL	ACN	ADBE	ADI	ADM	ADP	ADSK	AEE	AEP	AES
AIG	AIZ	AJG	AKAM	ALB	ALGN	ALL	ALLE	AMAT	AMCR	AMD	AME	AMGN	AFL	AMP
AMT	AMZN	ANET	ANSS	AON	AOS	APA	APD	APH	APO	APTV	ARE	ATO	AVB	AVGO
AVY	AWK	AXON	AXP	AZO	BA	BAC	BALL	BAX	BBY	BDX	BEN	BF.B	BG	BIIB
BK	BKNG	BKR	BLDR	BLK	BMY	BR	BRK.B	BRO	BSX	BWA	BX	BXP	C	CAG
CAH	CARR	CAT	CB	CBOE	CBRE	CCI	CCL	CDNS	CDW	CE	CEG	CF	CFG	CHD
CHRW	CHTR	CI	CINF	CL	CLX	CMCSA	CME	CMG	CMI	CMS	CNC	CNP	COF	COO
COP	COR	COST	CPAY	CPB	CPRT	CPT	CRL	CRM	CRWD	CSCO	CSGP	CSX	CTAS	CTRA
CTSH	CTVA	CVS	CVX	CZR	D	DAL	DAY	DD	DE	DECK	DELL	DFS	DG	DGX
DHI	DHR	DIS	DLR	DLTR	DOC	DOV	DOW	DPZ	DRI	DTE	DUK	DVA	DVN	DXCM
EA	EBAY	ECL	ED	EFX	EG	EIX	EL	ELV	EMN	EMR	ENPH	EOG	EPAM	EQIX
EQR	EQT	ERIE	ES	ESS	ETN	ETR	EVRG	EW	EXC	EXPD	EXPE	EXR	F	FANG
FAST	FCX	FDS	FDX	FE	FFIV	FI	FICO	FIS	FITB	FMC	FOX	FOXA	FRT	FSLR
FTNT	FTV	GD	GDDY	GE	GEHC	GEN	GEV	GILD	GIS	GL	GLW	GM	GNRC	GOOG
GOOGL	GPC	GPN	GRMN	GS	GWW	HAL	HAS	HBAN	HCA	HD	HES	HIG	HII	HLT
HOLX	HON	HPE	HPQ	HRL	HSIC	HST	HSY	HUBB	HUM	HWM	IBM	ICE	IDXX	IEX
IFF	INCY	INTC	INTU	INVH	IP	IPG	IQV	IR	IRM	ISRG	IT	ITW	IVZ	J
JBHT	JBL	JCI	JKHY	JNJ	JNPR	JPM	K	KDP	KEY	KEYS	KHC	KIM	KKR	KLAC
KMB	KMI	KMX	KO	KR	KVUE	L	LDOS	LEN	LH	LHX	LII	LIN	LKQ	LLY
LMT	LNT	LOW	LRCX	LULU	LUV	LVS	LW	LYB	LYV	MA	MAA	MAR	MAS	MCD
MCHP	MCK	MCO	MDLZ	MDT	MET	META	MGM	MHK	MKC	MKTX	MLM	MMC	MMM	MNST
MO	MOH	MOS	MPC	MPWR	MRK	MRNA	MS	MSCI	MSFT	MSI	MTB	MTCH	MTD	MU
NCLH	NDAQ	NDSN	NEE	NEM	NFLX	NI	NKE	NOC	NOW	NRG	NSC	NTAP	NTRS	NUE
NVDA	NVR	NWS	NWSA	NXPI	O	ODFL	OKE	OMC	ON	ORCL	ORLY	OTIS	OXY	PANW
PARA	PAYC	PAYX	PCAR	PCG	PEG	PEP	PFE	PFG	PG	PGR	PH	PHM	PKG	PLD
PLTR	PM	PNC	PNR	PNW	PODD	POOL	PPG	PPL	PRU	PSA	PSX	PTC	PWR	PYPL
QCOM	RCL	REG	REGN	RF	RJF	RL	RMD	ROK	ROL	ROP	ROST	RSG	RTX	RVTY
SBAC	SBUX	SCHW	SHW	SJM	SLB	SMCI	SNA	SNPS	SO	SOLV	SPG	SPGI	SRE	STE
STLD	STT	STX	STZ	SW	SWK	SWKS	SYF	SYK	SYY	T	TAP	TDG	TDY	TECH
TEL	TER	TFC	TFX	TGT	TJX	TMO	TMUS	TPL	TPR	TRGP	TRMB	TROW	TRV	TSCO
TSLA	TSN	TT	TTWO	TXN	TXT	TYL	UAL	UBER	UDR	UHS	ULTA	UNH	UNP	UPS
URI	USB	V	VICI	VLO	VLTO	VMC	VRSK	VRSN	VRTX	VST	VTR	VTRS	VZ	WAB
WAT	WBA	WBD	WDAY	WDC	WEC	WELL	WFC	WM	WMB	WMT	WRB	WST	WTW	WY
WYNN	XEL	XOM	XYL	YUM	ZBH	ZBRA								

Annex B - Code Repository

The complete code supporting this thesis is available at:

https://github.com/Etienne-larchet/smartptf

Please refer to the commit titled "*Master thesis complete project*" to access the version of the code corresponding to this thesis. Future commits may be added for further improvements.


ESSCA