

AVTORJI: Andre Sealy, Jingrui Li

1

When Does Reinforcement Learning Beat Naive Diversification? Learning Architecture, Regime Information, and Trading Frictions

Andre Sealy* Jingrui Li[†]

September 11, 2026

Abstract

This paper examines when reinforcement learning (RL) can outperform naive diversification in portfolio choice. We compare online actor–critic reinforcement learning with an offline Decision Transformer and the equally weighted $1/N$ benchmark, focusing on three determinants of out-of-sample performance: learning architecture, regime information, and trading frictions. We find that conventional online actor–critic methods provide only modest improvements over $1/N$, whereas the Decision Transformer delivers substantially stronger risk-adjusted performance. The value of regime information is non-monotonic: selected macroeconomic signals improve portfolio performance, but combining multiple regime-conditioning variables can materially reduce out-of-sample Sharpe ratios, consistent with an information–complexity tradeoff. These findings remain robust to alternative transaction-cost assumptions, and portfolio holdings reveal economically distinct allocation policies across architectures and information sets. Overall, reinforcement learning does not universally dominate naive diversification; its effectiveness depends critically on the interaction between learning architecture, state representation, and implementation frictions.

Keywords: *Reinforcement Learning; Deep Reinforcement Learning; Decision Transformer; Portfolio Choice; Naive Diversification; Regime Information; Trading Frictions.*

JEL Classification: *G11, G12, C45, C61.*

*School of Business, Stevens Institute of Technology. Email: asealy@stevens.edu.

[†]Corresponding Author. School of Business, Stevens Institute of Technology. Email: jli264@stevens.edu.

1 Introduction

The equally weighted portfolio remains one of the most difficult benchmarks to beat in empirical portfolio choice. Despite decades of advances in portfolio optimization, sophisticated strategies often fail to deliver consistent out-of-sample improvements over the simple $1/N$ rule. DeMiguel, Garlappi, and Uppal (2009) show that, across 14 portfolio models and seven empirical datasets, none consistently outperforms $1/N$ in terms of Sharpe ratios, certainty-equivalent returns, or turnover. This persistence highlights a fundamental challenge in portfolio management: richer and more flexible models can exploit additional information, but they also introduce greater estimation error, instability, and trading activity. These concerns are especially important in financial markets, where expected returns are difficult to estimate, signal-to-noise ratios are low, and relationships between predictive variables and asset returns can change over time.

Recent developments in machine learning offer a different approach to this problem. Rather than estimating expected returns and covariance matrices and then solving a separate portfolio-optimization problem, reinforcement learning (RL) learns portfolio policies directly from sequential interactions with a financial environment. An RL agent observes market and portfolio states, chooses allocations, receives rewards based on subsequent portfolio performance, and updates its decision rule accordingly. This framework is attractive for portfolio choice because it can accommodate nonlinear relationships, high-dimensional state spaces, transaction costs, and changing market conditions. Yet the same flexibility that makes RL appealing also creates substantial risks of overfitting and unstable out-of-sample performance. More broadly, evidence from machine learning in asset pricing shows that greater model complexity does not necessarily dominate simpler alternatives when financial signals are weak, noisy, or unstable (Gu, Kelly, and Xiu 2020).

This paper asks when reinforcement learning can overcome these limitations and outperform naive diversification. We focus on three dimensions that determine whether RL generates economically meaningful portfolio improvements: the learning architecture, the information supplied to the agent, and the trading frictions associated with active portfolio adjustment. In particular, we distinguish conventional online actor-critic reinforcement learning from offline reinforcement learning implemented through the Decision Transformer.¹ Online actor-critic algorithms learn through repeated policy

¹The distinction between online and offline reinforcement learning concerns how the model learns from experience. Online RL updates its policy through repeated interaction with an environment: the agent observes a state, takes an action, receives a reward, and uses that feedback to revise its policy. Offline RL instead learns from a fixed historical

updates within the training environment, whereas the Decision Transformer reformulates sequential decision making as a conditional sequence-modeling problem and learns portfolio actions from a fixed set of historical trajectories (Chen et al. 2021). This distinction is economically important because different learning architectures may extract and process the same noisy and nonstationary financial information in fundamentally different ways.

A second question is whether providing reinforcement-learning agents with more information about the economic environment necessarily improves portfolio decisions. Financial markets are nonstationary, and the relationships among asset returns, volatility, and macroeconomic conditions can vary over time. We therefore augment the portfolio state with regime-conditioning variables based on oil prices, economic growth, and unemployment. These variables capture distinct dimensions of the broader economic environment and may allow learning algorithms to adapt portfolio allocations as conditions change. However, additional information also expands the dimensionality of the state space and increases the complexity of the learning problem. If regime signals are redundant, slow-moving, or only weakly informative for portfolio choice, adding them may increase estimation error and reduce out-of-sample performance. This creates an information–complexity tradeoff: richer state representations can improve learning when they contain useful conditioning information, but more information need not produce better portfolios.

This tradeoff is particularly relevant for sequence-based learning. The Transformer architecture introduced by Vaswani et al. (2017) uses self-attention to capture dependencies across sequences, while the Decision Transformer applies this architecture to reinforcement learning by conditioning actions on states, past actions, and desired returns-to-go (Chen et al. 2021). Such an architecture is potentially well suited to portfolio choice because investment decisions depend on sequences of market and portfolio states rather than isolated observations. At the same time, financial nonstationarity can make historical temporal relationships unstable out of sample. We therefore examine whether conditioning the Decision Transformer on macroeconomic information improves its ability to extract economically useful variation in the investment environment and, importantly, whether the benefits of additional information diminish as the state representation becomes increasingly complex.

Our empirical design compares the equally weighted $1/N$ portfolio with online actor–critic reinforcement-learning strategies and an offline Decision Transformer within a common portfolio

dataset of previously observed states, actions, and outcomes without generating new training experience through interaction. In our setting, the online actor–critic models learn recursively within the historical trading environment, whereas the Decision Transformer learns portfolio decisions from fixed historical trajectories.

environment. All strategies face the same investment universe, portfolio constraints, and implementation assumptions, allowing us to attribute performance differences more directly to learning architecture and information content. We evaluate out-of-sample returns, volatility, Sharpe ratios, turnover, and portfolio holdings while explicitly incorporating transaction costs. We then augment the learning environment with alternative combinations of regime-conditioning variables to determine whether additional economic information systematically improves performance. This design allows us to address three related questions: whether reinforcement learning can outperform naive diversification, whether offline sequence learning differs systematically from conventional online reinforcement learning, and whether the value of regime information depends on the architecture through which that information is processed.

Our results show that reinforcement learning does not universally dominate naive diversification. Conventional online actor–critic methods provide only modest improvements over the $1/N$ benchmark in the baseline specification, and their performance is highly sensitive to the macroeconomic information included in the state representation. Selected regime-conditioning variables materially improve risk-adjusted performance, whereas other combinations substantially underperform equal weighting. Offline sequence learning performs more strongly: the vanilla Decision Transformer produces higher risk-adjusted performance than both $1/N$ and the baseline online RL model.² Importantly, however, additional regime information does not monotonically improve the Decision Transformer either. Selected macroeconomic signals are useful, while combining a broader set of regime variables can materially reduce out-of-sample performance. Robustness tests further show that the main rankings persist under alternative transaction-cost assumptions, and portfolio holdings reveal economically distinct allocation policies across architectures and information sets. The evidence therefore indicates that successful portfolio learning depends not simply on supplying an algorithm with more information, but on whether the learning architecture can convert that information into stable and economically useful decisions.

The paper makes three contributions. First, we extend the portfolio-learning literature by placing online actor–critic reinforcement learning, offline sequence-based reinforcement learning, and naive diversification in a common empirical framework. Rather than asking only whether a particular RL

²The vanilla Decision Transformer is the baseline offline specification that uses only the portfolio, market, and technical information contained in the standard state representation, without incorporating any of the macroeconomic regime-conditioning variables. It therefore isolates the contribution of the Decision Transformer architecture itself before additional regime information is introduced.

algorithm can beat $1/N$, we examine whether the architecture through which portfolio policies are learned determines the ability to generate robust out-of-sample improvements. Second, we contribute to the literature on state representation and regime-dependent portfolio choice by documenting a non-monotonic relationship between macroeconomic information and portfolio performance. Richer information sets can improve learning when selected signals provide useful conditioning information, but they can also degrade performance when state complexity increases faster than economically relevant information. Third, we incorporate trading frictions and portfolio turnover directly into the comparison between learning-based and naive strategies and use realized portfolio holdings to show how differences in architecture and state information translate into distinct allocation policies. These tests establish that the strongest performance results are not driven solely by aggressive rebalancing or negligible implementation costs.

Our findings have a broader implication for the use of artificial intelligence in portfolio management. The relevant question is not whether increasingly sophisticated learning methods can process more financial information, but whether they can transform that information into portfolio decisions that remain economically valuable out of sample. The persistence of $1/N$ as a strong benchmark underscores the value of simplicity when estimation uncertainty is substantial. Reinforcement learning can improve on this benchmark, but its success depends critically on the interaction among learning architecture, state representation, and implementation frictions. In this sense, sophisticated learning is most valuable not when it maximizes the amount of information incorporated into a portfolio rule, but when it selectively extracts useful information while controlling the complexity and trading costs associated with active decision making.

The remainder of the paper proceeds as follows. Section 2 reviews the related literature on naive diversification, reinforcement learning, Transformer-based portfolio methods, and regime-dependent financial learning. Section 3 develops the conceptual framework linking learning architecture, information complexity, and trading frictions. Section 4 describes the asset universe, portfolio information set, regime-conditioning variables, and experimental design. Section 5 presents the common portfolio environment, online actor–critic reinforcement learning, the Decision Transformer, and the model variants. Section 6 reports the baseline comparison among online reinforcement learning, offline reinforcement learning, and the $1/N$ benchmark. Section 7 examines the value of regime information and the information–complexity tradeoff. Section 8 provides additional evidence on trading frictions, portfolio turnover, model robustness, and learned allocation patterns. Section 9 concludes.

2 Related Literature

This paper builds on the classical literature on portfolio choice and asset pricing. Markowitz (1952) establishes the mean–variance framework, while Sharpe (1964), Lintner (1969), and Mossin (1966) develop the Capital Asset Pricing Model linking expected returns to systematic risk. Black (1972) relaxes the assumption of unrestricted risk-free borrowing and lending, and Merton (1973) extends portfolio choice to time-varying investment opportunities. A persistent empirical challenge is that optimal portfolios depend on parameters estimated with substantial error. DeMiguel, Garlappi, and Uppal (2009) show that sophisticated portfolio rules fail to consistently outperform $1/N$ out of sample, establishing naive diversification as a demanding benchmark. We retain this benchmark but ask whether reinforcement learning can improve portfolio choice without separately estimating expected returns and covariances before optimizing weights.

A related literature uses machine learning to extract nonlinear and high-dimensional information from financial data. Gu, Kelly, and Xiu (2020) show that machine-learning methods can improve equity-premium prediction, while Kelly, Pruitt, and Su (2019) connect observable characteristics to time-varying factor exposures and expected returns. Gu, Kelly, and Xiu (2021) extend this approach with nonlinear autoencoder factor models, and Feng, Giglio, and Xiu (2020) emphasize redundancy and model selection in high-dimensional factor environments. Giglio, Kelly, and Xiu (2022) review the broader integration of machine learning into asset pricing. Deep-learning approaches further permit nonlinear interactions and temporal dependence: Chen, Pelger, and Zhu (2024) use neural networks and recurrent architectures to extract macroeconomic states, while Cong et al. (2020) show that sequence and attention-based architectures can capture nonlinearities and path dependence. Importantly, this literature also shows that flexible models require economically informative state representations; simply expanding the set of macroeconomic inputs can worsen out-of-sample performance when irrelevant information obscures useful signals. This insight directly motivates our information–complexity tradeoff.

Our paper is most directly related to reinforcement learning in financial markets. Early applications formulate trading and portfolio problems as sequential decision tasks in which current actions affect future wealth. Neuneier (1997) applies Q -learning to portfolio management, Moody and Saffell (1998) develops reinforcement-learning trading systems that directly optimize financial objectives, and Chan and Shelton (2001) applies an actor–critic framework to market making. More recent work combines reinforcement learning with deep neural networks to learn nonlinear trading and asset-allocation policies

from high-dimensional states. As summarized by Pippas, Ludvig, and Turkay (2025), actor–critic methods are now an important part of the financial RL literature. We contribute by comparing online actor–critic learning directly with both naive diversification and an offline sequence-based architecture under a common portfolio environment.

A complementary strand applies attention and Transformer architectures to financial decision making. Vaswani et al. (2017) introduce the Transformer and multi-head self-attention, while Cong et al. (2021) develop AlphaPortfolio, combining sequential representation and cross-asset attention to learn portfolio allocations directly from financial data. Li et al. (2023) extend related attention-based methods to longer and higher-frequency sequences in online portfolio selection. The Decision Transformer of Chen et al. (2021) provides a particularly relevant bridge between Transformers and reinforcement learning by reformulating sequential decisions as conditional sequence modeling over states, actions, and returns-to-go. We use this framework to compare offline sequence learning with conventional online actor–critic methods as alternative mechanisms for translating the same financial information into portfolio decisions.

We also relate to research on regime dependence, macroeconomic conditioning information, and nonstationarity. Merton (1973) motivates conditioning portfolio decisions on changes in investment opportunities, while Pelger and Xiong (2022) show that macroeconomic states are relevant for capturing time variation in high-dimensional factor models. Yan (2024) incorporate regime switching into a deep deterministic policy-gradient portfolio model, and Chen, Pelger, and Zhu (2024) show that extracting a small number of dynamic macroeconomic states can be more informative than feeding a large set of macro variables directly into a learning model. Related work emphasizes adaptation to changing feature distributions: Huang, Sahm, and Ng (2024) highlights the role of state information in reinforcement-learning environments, while Liu et al. (2022) shows that nonstationarity can impair Transformer-based forecasting. Motivated by this literature, we use oil prices, GDP, and unemployment as regime-conditioning signals and test whether their informational benefits exceed the complexity they introduce.

Our contribution lies at the intersection of these literatures. Classical portfolio theory establishes the benefits of conditioning allocations on investment opportunities, while empirical portfolio choice shows how estimation error can make simple diversification difficult to beat. Machine learning and reinforcement learning provide more flexible mechanisms for learning dynamic portfolio policies, but practical implementation introduces an additional constraint: Avramov, Cheng, and Metzker (2023)

show that machine-learning portfolio performance can deteriorate materially in the presence of trading frictions. We bring these strands together by comparing online and offline reinforcement learning against the same $1/N$ benchmark, examining how the value of macroeconomic regime information depends on learning architecture, and testing whether performance survives transaction costs and portfolio turnover.

3 Conceptual Framework

This section develops the framework for comparing reinforcement learning with naive diversification. Portfolio-learning performance depends on learning architecture, state complexity, and trading costs. The $1/N$ strategy provides a demanding benchmark because it requires little estimation, whereas RL can adapt dynamically to market information but is more exposed to overfitting, unstable policies, and turnover. We therefore view the comparison as a tradeoff between adaptive information processing and the costs of model complexity and active portfolio management.

3.1 Reinforcement Learning versus Naive Diversification

Consider an investment universe containing N risky assets. At time t , the investor chooses a portfolio allocation

$$a_t = \boldsymbol{\pi}_t = (\pi_{1,t}, \pi_{2,t}, \dots, \pi_{N,t})^\top, \quad (1)$$

subject to

$$\sum_{i=1}^N \pi_{i,t} = 1, \quad \pi_{i,t} \geq 0, \quad (2)$$

so that the portfolio is fully invested and short selling is not permitted. The naive diversification strategy sets

$$\pi_{i,t}^{1/N} = \frac{1}{N}, \quad i = 1, \dots, N, \quad (3)$$

and therefore does not condition portfolio weights on estimates of expected returns, covariance matrices, technical indicators, or macroeconomic states. Its primary advantage is simplicity. Because $1/N$ does not attempt to infer a complex mapping between observed market information and future portfolio outcomes, it is largely insulated from the estimation error that can undermine more sophisticated allocation rules.

Reinforcement learning instead allows portfolio allocations to depend on an observed state s_t . In

the portfolio environment used in this paper, the state contains information describing the current portfolio and market environment, including asset holdings, prices, technical indicators, and, in selected specifications, macroeconomic regime-conditioning variables. The RL policy can therefore be represented generally as

$$a_t = \pi_\theta(s_t), \quad (4)$$

where π_θ denotes a policy parameterized by θ . The agent seeks a policy that maximizes expected cumulative rewards,

$$\max_{\theta} \mathbb{E}_{\pi_\theta} \left[\sum_{t=1}^T r_t \right], \quad (5)$$

where r_t is the portfolio reward at time t . In contrast to conventional portfolio optimization, the RL policy need not separately estimate expected returns and covariance matrices before determining optimal weights. Instead, it learns a direct mapping from observed states to allocation decisions.

This flexibility creates the possibility that RL can outperform $1/N$ when the state variables contain economically useful information about future investment opportunities. Let $\mathcal{I}_t$ denote the information set available at time t . A reinforcement-learning strategy can condition on

$$\mathcal{I}_t = \{\mathbf{P}_t, \mathbf{H}_t, \mathbf{I}_t, \mathbf{Z}_t\}, \quad (6)$$

where $\mathbf{P}_t$ denotes asset prices, $\mathbf{H}_t$ represents portfolio holdings, $\mathbf{I}_t$ contains technical indicators, and $\mathbf{Z}_t$ contains regime-conditioning information. The potential advantage of RL therefore arises from its ability to transform variation in $\mathcal{I}_t$ into time-varying portfolio weights. If the information set contains signals that are relevant for subsequent returns or risks and the learning algorithm can extract those signals reliably, then an adaptive portfolio can dominate a static allocation.

However, greater flexibility does not imply superior out-of-sample performance. The $1/N$ portfolio avoids the estimation burden associated with learning the mapping $\mathcal{I}_t \rightarrow a_t$. RL must estimate this mapping from a finite and noisy historical sample. In financial markets, where predictive relationships are often weak and unstable, the additional flexibility of an RL policy can generate fitting error that offsets the benefits of conditioning on additional information. We therefore view the relative performance of RL and $1/N$ as depending on whether the incremental value of adaptive allocation exceeds the costs associated with learning and implementing a more complex portfolio rule.

This argument can be summarized by expressing the net performance of strategy j as

$$\mathcal{P}_j = \mathcal{B}_j - \mathcal{E}_j - \mathcal{F}_j, \quad (7)$$

where $\mathcal{B}_j$ denotes the gross benefit obtained from conditioning portfolio decisions on economically useful information, $\mathcal{E}_j$ captures losses arising from estimation error and model instability, and $\mathcal{F}_j$ represents implementation frictions. For the $1/N$ strategy, $\mathcal{B}_j$ may be limited because the portfolio does not adapt to changing market information, but both $\mathcal{E}_j$ and $\mathcal{F}_j$ are relatively small. Reinforcement learning can increase $\mathcal{B}_j$ by exploiting conditional information, but it can simultaneously increase both estimation error and trading costs. RL therefore outperforms naive diversification only when

$$\Delta\mathcal{B} > \Delta\mathcal{E} + \Delta\mathcal{F}. \quad (8)$$

This condition provides the basic organizing principle for the empirical analysis.

3.2 Online versus Offline Learning

The architecture through which the agent learns may materially affect this tradeoff. We distinguish between online actor-critic reinforcement learning and offline sequence-based reinforcement learning. Both approaches use historical financial information to learn portfolio policies, but they differ in how the learning problem is formulated and how actions are associated with subsequent rewards. Online RL agents update their policies through repeated interaction with the training environment. At each timestep, the agent observes the current state s_t , selects an action a_t , receives reward r_t , transitions to state s_{t+1} , and updates the policy using the observed transition.

The interaction can be summarized as

$$s_t \rightarrow a_t \rightarrow r_t, s_{t+1} \rightarrow \theta_{t+1}. \quad (9)$$

Actor-critic methods separate the learning problem into an actor, which selects portfolio actions, and a critic, which evaluates the quality of the current policy or action. This architecture can adapt flexibly to information in the training environment and learn nonlinear policies, but repeated policy updates can also become unstable when financial rewards are noisy.³

³In high-dimensional state spaces, even modest changes in the training sample or state representation can lead to

Offline reinforcement learning begins from a different learning problem. The agent receives a fixed dataset of previously observed trajectories and learns a policy without altering the underlying data distribution during training. Let

$$\mathcal{D} = \{\tau^{(1)}, \tau^{(2)}, \dots, \tau^{(M)}\} \quad (10)$$

denote a dataset of historical trajectories, where each trajectory can be written as

$$\tau = \{(R_t, s_t, a_t), \dots, (R_T, s_T, a_T)\}. \quad (11)$$

The return-to-go at time t is defined as

$$R_t = \sum_{k=t}^T r_k. \quad (12)$$

The Decision Transformer reformulates the RL problem as conditional sequence modeling and estimates the probability of an action conditional on desired returns, observed states, and previous actions:

$$a_t \sim p_\theta(a_t \mid R_t, s_{\leq t}, a_{< t}). \quad (13)$$

Rather than estimating a recursive value function and repeatedly updating a policy through actor–critic feedback, the Decision Transformer learns relationships among returns, states, and actions directly from observed sequences.

This distinction may be particularly important in finance. Historical market realizations are fixed, financial rewards are noisy, and structural changes can make recursive policy updates sensitive to particular portions of the training sample. Offline sequence modeling may provide a more stable mapping from market histories to portfolio actions because the learner conditions on complete historical trajectories rather than relying exclusively on recursive value estimation. Self-attention also permits the model to place different weights on information observed at different points in the trajectory, potentially allowing the Decision Transformer to capture long-range dependencies between financial states and portfolio outcomes. Offline sequence learning instead conditions on a fixed historical dataset and learns state–action relationships directly from observed trajectories.⁴

materially different policy updates, increasing sensitivity to estimation noise and initialization.

⁴Because offline policies are estimated from historical trajectories, their effective learning domain is naturally tied to the states and actions represented in the training data. This feature distinguishes offline learning from online actor–critic methods, which update policies recursively within the training environment. In our setting, the distinction provides a useful basis for comparing the stability and economic performance of the two architectures under the same asset universe,

3.3 The Information–Complexity Tradeoff

The second dimension of the framework concerns the amount and type of information supplied to the learning algorithm. Financial markets are nonstationary, and the statistical properties of asset returns can vary with broader economic conditions. We therefore augment the baseline state with macroeconomic variables intended to capture changes in the investment environment. In the empirical analysis, these regime-conditioning variables include West Texas Intermediate (WTI) crude oil prices, the U.S. unemployment rate, and U.S. gross domestic product. These variables are treated as additional components of the state representation rather than as deterministic labels that assign each observation to a discrete regime.

Let $s_t^{(0)}$ denote the baseline state containing portfolio, market, and technical information. Adding a regime-conditioning signal $Z_{j,t}$ creates the augmented state

$$s_t^{(1)} = [s_t^{(0)}, Z_{j,t}]. \quad (14)$$

Adding additional signals further expands the state representation:

$$s_t^{(K)} = [s_t^{(0)}, Z_{1,t}, Z_{2,t}, \dots, Z_{K,t}]. \quad (15)$$

The benefit of this expansion is that the learner can condition portfolio allocations on a richer description of the economic environment. If $Z_{j,t}$ contains information about future return distributions, volatility, or correlations, then conditioning on this information may produce better portfolio choices than treating all observations as if they arise from the same stationary environment.

However, the informational benefit is accompanied by an increase in model complexity. The previous implementation illustrates this directly: expanding the number of regime-conditioning variables increases the size of the state representation faced by the learning algorithm. The agent therefore must estimate portfolio decisions over a larger effective state space as additional variables are introduced. The relevant relationship can be written conceptually as

$$\mathcal{P} = f(\mathcal{I}, \mathcal{C}), \quad (16)$$

information set, constraints, and trading environment.

where $\mathcal{I}$ denotes the amount of economically useful information contained in the state and $\mathcal{C}$ denotes the complexity generated by representing and learning from that information. Adding a variable Z generally increases the available information,

$$\mathcal{I}^+ = \mathcal{I} \cup Z, \quad (17)$$

but it also increases the dimensionality of the learning problem,

$$\dim(s_t^+) > \dim(s_t). \quad (18)$$

Consequently, the marginal effect of additional information on portfolio performance need not be positive:

$$\frac{\partial \mathcal{P}}{\partial Z} \geq 0. \quad (19)$$

When a new state variable provides incremental information that can be learned reliably, the informational benefit may dominate the increase in complexity. When the variable is redundant, slow moving, noisy, or weakly related to portfolio opportunities, the additional complexity can instead reduce out-of-sample performance. The same problem may become more severe when several correlated or overlapping variables are introduced simultaneously.

We refer to this mechanism as the information–complexity tradeoff. It implies that richer state representations should not be expected to monotonically improve reinforcement-learning portfolios. Instead, the value of additional state information depends jointly on its economic content and on the ability of the learning architecture to extract that information without overfitting. This distinction is especially important when comparing online and offline RL. Actor–critic algorithms and Decision Transformer may respond differently to the same expansion in the state space because they learn portfolio policies through different mechanisms. A regime-conditioning variable that stabilizes or improves one architecture may provide little value, or even reduce performance, in another.

State normalization is related to this tradeoff. The individual components of a financial state can differ substantially in scale and distribution. Prices, technical indicators, macroeconomic variables, and portfolio holdings may exhibit different means, variances, and persistence. Let x_t denote a generic

state feature. Its normalized representation can be written as

$$\tilde{x}_t = \frac{x_t - \mu_x}{\sigma_x}, \quad (20)$$

where μ_x and σ_x denote the corresponding location and scale parameters estimated from the training data. Normalization does not eliminate the information–complexity tradeoff, but it places heterogeneous state variables on comparable scales and can improve numerical stability during training. In the empirical implementation, state statistics are therefore used to standardize the information supplied to the learning algorithms while preserving the underlying variation across market conditions.

The framework generates a clear empirical implication. If additional macroeconomic information is uniformly beneficial, portfolio performance should improve as regime-conditioning variables are added to the state. In contrast, the information–complexity mechanism predicts a non-monotonic relationship: selected signals may improve performance, while increasingly rich combinations may eventually reduce it. The empirical results therefore evaluate both individual regime-conditioning variables and combinations of these variables rather than assuming that a larger information set is necessarily preferable.

3.4 Trading Frictions

The final component of the framework concerns the cost of translating learned portfolio policies into actual trades. The $1/N$ benchmark requires relatively limited rebalancing, whereas reinforcement-learning strategies can adjust target weights whenever the state changes. Consequently, an RL strategy can generate superior gross performance yet fail to outperform naive diversification after accounting for turnover and transaction costs.

Let TC_t denote portfolio turnover at time t . The net performance advantage of an RL strategy relative to $1/N$ can be summarized as

$$\Delta R^{\text{net}} = \Delta R^{\text{gross}} - \Delta C^{\text{trade}}, \quad (21)$$

where ΔR^{gross} is the gross performance difference and ΔC^{trade} is the incremental trading cost generated by active portfolio adjustment. Reinforcement learning therefore creates economic value only when the benefits of dynamic reallocation are large enough to offset the implementation costs associated with that reallocation.

Trading frictions also interact with estimation error. A highly responsive policy can translate small changes in noisy state variables into portfolio trades, thereby increasing turnover without generating commensurate improvements in expected performance. We therefore impose transaction costs and limits on portfolio adjustment in the empirical environment, with the detailed implementation described in Section 5.1.

$$\mathcal{P}^{\text{net}} = f(\mathcal{M}, \mathcal{I}, \mathcal{C}, \mathcal{F}) \quad (22)$$

where $\mathcal{M}$ denotes learning architecture, $\mathcal{I}$ useful information, $\mathcal{C}$ state complexity, and $\mathcal{F}$ trading frictions. The framework does not predict that reinforcement learning should universally dominate naive diversification. Instead, RL is most likely to outperform $1/N$ when the learning architecture extracts economically useful information without generating excessive estimation complexity or implementation costs.

4 Data and Experimental Design

This section describes the asset universe, information set, regime-conditioning variables, and experimental design. All strategies are evaluated within the same portfolio environment, with common assets, constraints, transaction costs, and out-of-sample criteria. Performance differences can therefore be attributed more directly to learning architecture and state information rather than to differences in the underlying investment problem.

4.1 Asset Universe and Sample Construction

Our investment universe consists of $N = 393$ individual stocks. The same set of assets is used across the equally weighted benchmark, the online reinforcement-learning models, and the Decision Transformer so that all strategies face a common portfolio-selection problem. At each trading date, the agent allocates capital across these securities subject to full-investment and no-short-selling constraints. The common asset universe is particularly important for our analysis because the purpose of the paper is not to compare alternative security-selection universes, but rather to examine how different learning architectures transform the same underlying financial information into portfolio allocations.

For asset i at time t , let $P_{i,t}$ denote its observed market price and let $r_{i,t}$ denote its return. The

vector of asset prices is

$$\mathbf{P}_t = (P_{1,t}, P_{2,t}, \dots, P_{N,t})^\top, \quad (23)$$

and the corresponding vector of asset returns is

$$\mathbf{r}_t = (r_{1,t}, r_{2,t}, \dots, r_{N,t})^\top. \quad (24)$$

These market observations provide the basis for constructing the portfolio state and for computing the realized rewards received by the learning algorithms. Each portfolio begins with an initial value of

$$V_0 = \$1,000,000. \quad (25)$$

To ensure a common and economically neutral starting point, the initial portfolio is equally allocated across the N assets:

$$w_i(0) = \frac{1}{N}, \quad i = 1, \dots, N. \quad (26)$$

The reinforcement-learning strategies therefore begin from the same allocation as the naive benchmark and subsequently earn any relative advantage through learned reallocation decisions rather than through different initial portfolio positions. The portfolio environment is long-only and fully invested. Target portfolio weights satisfy

$$\sum_{i=1}^N \pi_{i,t} = 1, \quad 0 \leq \pi_{i,t} \leq 1, \quad (27)$$

where $\pi_{i,t}$ denotes the target allocation selected for asset i at time t . We additionally impose a maximum adjustment of $H_{\max} = 100$ shares per stock per day. This restriction prevents the learning algorithms from generating implausibly large reallocations and provides a common implementation constraint across the online and offline learning strategies.

4.2 Portfolio Information Set

The Reinforcement Learning agents receive a state representation that combines portfolio information, market prices, and technical indicators. Let C_t denote the portfolio's cash position, $\mathbf{H}_t$ the vector of current asset holdings, $\mathbf{P}_t$ the vector of asset prices, and $\mathbf{I}_t$ the collection of technical indicators. The baseline financial state can be represented as

$$s_t^{(0)} = [C_t, \mathbf{H}_t^\top, \mathbf{P}_t^\top, \mathbf{I}_t^\top]^\top. \quad (28)$$

This formulation allows the learning agent to condition portfolio decisions on both the current composition of the portfolio and contemporaneous information about the underlying assets. For each stock, we construct technical indicators intended to summarize different dimensions of market conditions, including price trends, momentum, and volatility. The indicator matrix is

$$\mathbf{I}_t = \begin{bmatrix} T_{11,t} & T_{12,t} & \cdots & T_{1M,t} \\ T_{21,t} & T_{22,t} & \cdots & T_{2M,t} \\ \vdots & \vdots & \ddots & \vdots \\ T_{N1,t} & T_{N2,t} & \cdots & T_{NM,t} \end{bmatrix}, \quad (29)$$

where $T_{ij,t}$ denotes technical indicator j for asset i at time t . The baseline implementation contains eight technical features. These variables are treated as components of the state rather than as stand-alone trading rules, allowing the reinforcement-learning algorithms to determine whether and how each signal should influence portfolio allocations.

The technical-information set includes Bollinger Bands, which summarize the position of prices relative to moving averages and provide a measure of local price variation; the Relative Strength Index (RSI), which captures recent price momentum; the Commodity Channel Index (CCI), which measures deviations in prices relative to their recent historical average; the Directional Index, which summarizes the strength of prevailing price trends; and short- and longer-horizon simple moving averages. Collectively, these variables provide information about volatility, momentum, and trend conditions without requiring the agent to rely exclusively on raw asset prices.

The action space corresponds to the vector of desired portfolio weights,

$$a_t = \boldsymbol{\pi}_t = (\pi_{1,t}, \pi_{2,t}, \dots, \pi_{N,t})^\top. \quad (30)$$

The distinction between state and action is therefore economically intuitive: the state summarizes the portfolio and market information available to the learner, while the action determines how the agent reallocates wealth across the available securities.

4.3 Regime-Conditioning Variables

We augment the baseline state with macroeconomic variables designed to provide information about changes in the broader economic environment. Specifically, we use WTI crude oil prices, the U.S. unemployment rate, and log U.S. gross domestic product. Rather than treating these variables as deterministic labels that assign each date to a particular discrete market regime, we interpret them as *regime-conditioning variables*. They provide the learning algorithms with additional information about economic conditions that may affect expected returns, volatility, correlations, and the attractiveness of different portfolio allocations.

Let

$$\mathbf{Z}_t = (Z_{1,t}, Z_{2,t}, Z_{3,t})^\top \quad (31)$$

denote the vector of regime-conditioning variables, where the three elements correspond to oil prices, unemployment, and GDP. The regime-conditioned portfolio state is then

$$s_t^{(Z)} = [s_t^{(0)}, \mathbf{Z}_t^\top]^\top. \quad (32)$$

This structure allows the reinforcement-learning algorithms to determine whether macroeconomic state information contains incremental information for portfolio choice beyond asset-level prices and technical indicators.

The empirical design varies the regime-conditioning information set rather than imposing a single preferred specification. We consider each of the three macroeconomic variables individually and in alternative combinations. The resulting specifications include Oil, GDP, Unemployment, Oil/GDP, Oil/Unemployment, GDP/Unemployment, and Oil/GDP/Unemployment. We also estimate a vanilla specification containing no regime-conditioning variables. This design permits a direct assessment of whether increasing the amount of macroeconomic information systematically improves portfolio performance.

Expanding the regime-information set also expands the state representation confronted by the learning algorithm. In our implementation, the baseline environment contains 3,931 state elements. Adding one regime-conditioning variable increases the state representation to 4,324 elements, adding

two increases it to 4,717, and including all three increases it to 5,110:

$$\dim(s_t) = \begin{cases} 3,931, & K = 0, \\ 4,324, & K = 1, \\ 4,717, & K = 2, \\ 5,110, & K = 3, \end{cases} \quad (33)$$

where K denotes the number of regime-conditioning variables. This variation is central to the information–complexity tradeoff developed in Section 3.3. The additional variables potentially contain useful information about changing economic conditions, but they simultaneously increase the complexity of the portfolio-learning problem.

[Insert Figure 1 Here]

Figure 1 illustrates the increase in the state representation as additional macroeconomic variables are incorporated. This feature of the experimental design allows us to evaluate a central prediction of the conceptual framework: if additional information is uniformly valuable, performance should improve as the state becomes richer. If, instead, the marginal informational benefit is eventually dominated by increased complexity, portfolio performance may deteriorate as multiple regime signals are combined.

5 Portfolio Learning Methods

This section presents the common portfolio environment and the two learning architectures. All strategies use the same asset universe, baseline information, constraints, and transaction-cost structure; they differ primarily in how financial states are mapped into portfolio actions. Online actor–critic methods learn recursively through interaction with the training environment, whereas the Decision Transformer learns from fixed historical trajectories using conditional sequence modeling.

5.1 Reinforcement Learning Environment

We formulate portfolio allocation as a Markov decision process (MDP),

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \Theta), \quad (34)$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P represents state transitions, R denotes the reward function, and Θ contains the portfolio-environment parameters. The main parameters are $N = 393$ assets, initial wealth $V_0 = 1,000,000$, proportional transaction cost $\tau = 0.005$, a maximum trade size of $H_{\max} = 100$ shares per asset, and reward-scaling parameter $\alpha = 10^{-4}$.

At time t , the agent observes the financial state

$$s_t = \left[C_t, \mathbf{H}_t^\top, \mathbf{P}_t^\top, \text{vec}(\mathbf{I}_t)^\top, \mathbf{Z}_t^\top \right]^\top, \quad (35)$$

where C_t denotes cash, $\mathbf{H}_t$ asset holdings, $\mathbf{P}_t$ asset prices, $\mathbf{I}_t$ technical indicators, and $\mathbf{Z}_t$ the regime-conditioning variables when included. The vanilla specifications omit $\mathbf{Z}_t$. The agent chooses a vector of target portfolio weights,

$$a_t = \boldsymbol{\pi}_t = (\pi_{1,t}, \dots, \pi_{N,t})^\top, \quad (36)$$

subject to

$$\sum_{i=1}^N \pi_{i,t} = 1, \quad 0 \leq \pi_{i,t} \leq 1. \quad (37)$$

Thus, all strategies are long-only and fully invested. Portfolio return over the next period is

$$r_{p,t+1} = \sum_{i=1}^N \pi_{i,t} r_{i,t+1}. \quad (38)$$

Trading frictions are incorporated directly into portfolio wealth. Turnover is defined as

$$TC_t = \sum_{i=1}^N |\pi_{i,t} - w_{i,t}|, \quad (39)$$

and portfolio value evolves according to

$$V_{t+1} = V_t (1 + r_{p,t+1}) (1 - \tau TC_t). \quad (40)$$

We additionally impose the maximum-share adjustment constraint described in Section 4.1. These features ensure that the learning algorithms are evaluated in a common frictional environment and prevent performance from being driven by unconstrained portfolio reallocation.

5.2 Online Actor–Critic Reinforcement Learning

Our online reinforcement-learning benchmarks use Advantage Actor–Critic (A2C), Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG). All three models are model-free, actor-critic algorithms, where the actor module takes the state S_t as an input and determines the action A_t at time t , while the Critic module subsequently receives the state S_t and the action A_t determined by the Actor module. The algorithm then evaluates the state-action pair, and estimates the value of the state-action pair, using the reward as input, adhering to the framework outlined in Section 3.

Recall, a policy can be written generally as

$$a_t = \pi_\theta(s_t), \tag{41}$$

where θ denotes the actor parameters. The agent collects experience using π_θ , uses those observations to update the policy, and then discards that rollout data. Once the policy changes substantially, it stops reusing old trajectories for training.

One important difference from PPO/A2C and DDPG is that the former generates data only from the current policy. This means only actions taken by the current policy generate data at each training iteration, which can make the data unstable after policy updates. This makes PPO/A2C relatively sample-inefficient; however, on-policy algorithms are typically more stable and exhibit lower variance as they learn directly from the policy they are improving. PPO/A2C is commonly used in stock trading because it can adapt to market changes.

On the other hand, off-policy models such as DDPG do not have this issue, which allows them to use data generated from different policies. DDPG becomes more sample-efficient, where they can reuse past experiences stored in memory. However, DDPG is not without its drawbacks, such as requiring significant amounts of memory storage to preserve past experiences, along with stability and variance issues associated with the complexity of updating policies from different data sources. DDPG is often utilized in high frequency trading due to their ability to process large amounts of data, although the algorithm often requires longer training times.

For each online model, we estimate a vanilla specification using portfolio and technical information and additional specifications that incorporate the regime-conditioning variables described in Section 4.3.⁵

⁵Detailed value-function and policy-gradient formulations for A2C, PPO, and DDPG are provided in the Online Appendix C.

This design allows us to evaluate whether adding macroeconomic information improves or destabilizes the learned portfolio policy.

5.3 Offline Reinforcement Learning with Decision Transformers

The Decision Transformer approaches reinforcement learning as a conditional sequence-modeling problem (Chen et al. 2021). Rather than estimating value functions recursively, it learns portfolio actions directly from historical trajectories containing states, actions, and returns-to-go. A trajectory is represented as

$$\tau = \{(R_1, s_1, a_1), \dots, (R_T, s_T, a_T)\}, \quad (42)$$

where

$$R_t = \sum_{k=t}^T r_k \quad (43)$$

is the return-to-go from timestep t . The Decision Transformer estimates the conditional policy

$$a_t \sim p_\theta(a_t \mid R_t, s_{\leq t}, a_{< t}). \quad (44)$$

Thus, the model predicts the current portfolio action conditional on a desired return-to-go, the sequence of states observed to date, and previous portfolio actions. This formulation allows the model to exploit temporal relationships in historical trajectories without relying on recursive value-function estimation.

Our implementation uses a GPT-2-based Transformer initialized with pretrained GPT-2 weights. At each timestep, return-to-go, state, and action are embedded into a common hidden representation and combined with timestep embeddings. We use a context window of 20 timesteps, corresponding to a maximum sequence of 60 tokens, and project each token into a 128-dimensional hidden representation. A causal attention mask ensures that portfolio decisions at time t depend only on information available through that timestep.

[Insert Figure 2 Here]

Figure 2 summarizes the architecture. The principal distinction from the online actor-critic models is that the Decision Transformer learns a conditional mapping from complete historical state-action-return sequences rather than updating portfolio decisions through recursively estimated value functions. This sequence-based structure provides the basis for our comparison of online and offline

reinforcement learning under the same portfolio environment. We estimate the same information variants for the Decision Transformer as for the online models. The vanilla specification excludes macroeconomic regime-conditioning variables, while the remaining specifications incorporate oil prices, GDP, unemployment, and their combinations. Holding the architecture fixed while varying the state information allows us to identify the marginal value of additional regime information.

5.4 State Normalization and Model Variants

The financial state contains variables with different scales and persistence. We therefore normalize continuous state variables using means and standard deviations estimated from the training sample and apply the same transformations to validation and out-of-sample observations. This prevents future information from entering the preprocessing stage and improves numerical stability across learning architectures.⁶

For both online and offline learning, we estimate the same eight information specifications:

$$\mathcal{V} = \begin{cases} \text{Vanilla, Oil, GDP, Unemployment,} \\ \text{Oil + GDP, Oil + Unemployment, GDP + Unemployment,} \\ \text{Oil + GDP + Unemployment.} \end{cases} \quad (45)$$

The vanilla model contains portfolio, market, and technical information but no macroeconomic conditioning variables. The remaining specifications progressively expand the state representation. The resulting two-dimensional design—learning architecture crossed with information set—allows us to determine whether out-of-sample performance depends primarily on the learning method, the state information, or their interaction.

6 Baseline Results

We begin the empirical analysis with the simplest comparison in the paper: whether reinforcement-learning strategies outperform the equally weighted $1/N$ portfolio when no macroeconomic regime-conditioning variables are included in the state space. This baseline comparison isolates the effect of the learning architecture before introducing additional information. We first evaluate the online actor–critic model relative to $1/N$, then examine the offline Decision Transformer, and finally compare online and

⁶Detailed normalization formulas and state statistics are reported in the Online Appendix B.

offline reinforcement learning directly. All strategies are evaluated over the same out-of-sample period and under the common portfolio environment described in Sections 4 and 5.

6.1 Dataset Construction

Our sample comprises daily stock prices for firms in the S&P 500 from January 2, 2009, through May 1, 2024.⁷ All financial data are obtained from *Wharton Research Data Services* (WRDS), with daily stock-price data drawn from the *Center for Research in Security Prices* (CRSP). Our preprocessing procedure constructs a balanced panel of closing prices and removes securities with missing observations over the sample period.⁸

We divide the resulting sample chronologically into training and out-of-sample trading periods. In our reinforcement-learning setting, each trading day represents one time step within a sequential episode rather than an independent observation. The training period extends from January 1, 2009, through June 30, 2022, yielding 3,396 time steps, while the out-of-sample trading period extends from July 1, 2022, through May 1, 2024, yielding 459 time steps.⁹

6.2 Summary Statistics

We begin by characterizing the cross-sectional and time-series variation in the asset universe used to train the portfolio-learning models. Because reinforcement-learning policies are learned from realized variation in returns and dependence across assets, the economic heterogeneity of the training environment is important for understanding the opportunities available to the algorithms. We classify the S&P 500 firms in our sample according to the Kenneth French 10-industry classification: Consumer Nondurables (NoDur), Consumer Durables (Durbl), Manufacturing (Manuf), Energy (Enrgy), Business Equipment (HiTec), Telecommunications (Telcm), Wholesale, Retail, and Services (Shops), Health (Hlth), Utilities (Utils), and Other (Other). Table 1 reports summary statistics for daily returns aggregated by industry over the training sample.

[Insert Table 1 Here]

The return distributions are broadly comparable across industries, although economically mean-

⁷Because we require a complete price history over the full sample period, the balanced-panel construction introduces survivorship bias and leaves 393 firms in the final sample.

⁸This restriction avoids complications associated with securities entering after an initial public offering or exiting through delisting. Filling pre-IPO or post-delisting observations with zeros would be inappropriate because a zero price does not represent a tradable equity value and could distort returns, covariance estimates, portfolio values and weights, and the trading logic of the reinforcement-learning models.

⁹This chronological separation preserves the temporal structure of the portfolio-learning problem and prevents observations from the out-of-sample evaluation period from entering model training.

ingful differences remain in their dispersion and tail behavior. In particular, the Energy sector exhibits relatively greater return variability, consistent with its exposure to commodity-price and macroeconomic shocks. Differences in skewness and extreme returns across industries further indicate that the learning environment is not characterized by homogeneous asset behavior. This heterogeneity is important because it gives the portfolio algorithms scope to learn allocations that respond to differences in risk and return rather than simply distributing wealth uniformly across securities.

Cross-sectional variation in individual return distributions, however, captures only one dimension of the portfolio problem. Diversification also depends critically on the extent to which assets move together.¹⁰ Thus, the potential benefit from learning non-uniform portfolio weights depends not only on differences in individual asset risk but also on the correlation structure of the investment universe. This consideration is especially relevant to our comparison with the $1/N$ benchmark, which assigns equal weights without explicitly conditioning allocations on the covariance structure. Figure 3 therefore reports the correlation matrix of sector-level returns in the reinforcement-learning environment.

[Insert Figure 3 Here]

Thus, the potential benefit from learning non-uniform portfolio weights depends not only on differences in individual asset risk but also on the correlation structure of the investment universe. This consideration is especially relevant to our comparison with the $1/N$ benchmark, which assigns equal weights without explicitly conditioning allocations on the covariance structure. Figure 3 therefore reports the correlation matrix of sector-level returns in the reinforcement-learning environment.

The correlations are uniformly positive but vary meaningfully across sectors, ranging approximately from 0.50 to 0.92. Manufacturing, Other, Shops, and High Technology display relatively strong comovement with several other sectors, whereas Utilities and Energy tend to exhibit weaker correlations with the broader market environment. Manufacturing is particularly highly correlated with Consumer Durables, High Technology, and several other economically connected sectors. These patterns imply that the asset universe contains substantial common market exposure while still preserving meaningful cross-sector diversification opportunities. The learning algorithms must therefore distinguish between common variation and sector-specific information when choosing portfolio weights.

A second source of heterogeneity comes from the macroeconomic information supplied to the regime-conditioned models. The empirical analysis uses WTI oil prices, U.S. GDP, and the U.S. unemployment

¹⁰Recall that portfolio variance depends on both individual asset volatilities and pairwise correlations: $\sigma_p^2 = \mathbf{w}^\top \Sigma \mathbf{w} = \sum_i w_i^2 \sigma_i^2 + 2 \sum_{i < j} w_i w_j \sigma_i \sigma_j \rho_{ij}$.

rate because these variables capture distinct dimensions of economic conditions and are observed at different frequencies. Before introducing them into the reinforcement-learning state representation, it is useful to establish the extent to which they provide overlapping versus complementary information. Table 2 reports the pairwise correlations among log U.S. GDP, the unemployment rate, and WTI oil prices.

[Insert Table 2 Here]

The correlations indicate that the three variables are related but far from redundant. Log GDP is negatively associated with unemployment, consistent with the tendency for stronger economic activity to coincide with stronger labor-market conditions. WTI exhibits substantially weaker relationships with the two conventional macroeconomic indicators, suggesting that oil prices contain a different component of economic information. The variables also differ in observation frequency: GDP is available quarterly, unemployment monthly, and WTI daily. Consequently, the three signals differ not only in economic content but also in how rapidly they can reflect changes in the investment environment.

These distinctions motivate their use as regime-conditioning variables in the subsequent analysis. GDP provides a broad measure of aggregate economic activity, while unemployment captures labor-market conditions that often evolve more slowly over the business cycle. WTI can respond much more rapidly to commodity-market developments, inflationary pressures, global demand conditions, and geopolitical supply disruptions. The training period spans episodes with markedly different financial and macroeconomic conditions, including the aftermath of the Global Financial Crisis and the pandemic-related contraction and recovery. This variation gives the learning models exposure to a wide range of historical states while allowing us to test whether explicitly adding macroeconomic information improves out-of-sample portfolio decisions.

Importantly, the imperfect correlation among these variables does not imply that adding more of them must improve portfolio performance. Each additional conditioning signal expands the state representation and therefore increases the amount of information that the model must process. The correlation evidence thus provides an empirical motivation for the information–complexity tradeoff developed in Section 3.3: macroeconomic variables may provide complementary information, but combining them can also increase the complexity of the learning problem enough to reduce out-of-sample performance.

6.3 Can Online Reinforcement Learning Beat $1/N$?

To establish whether reinforcement learning delivers economically meaningful gains over naive diversification, we first compare the baseline specifications before introducing any regime-conditioning variables. This comparison isolates the role of learning architecture itself by holding the investment universe, portfolio constraints, transaction-cost assumptions, and out-of-sample evaluation period fixed across strategies. Table 3 therefore provides the natural starting point for assessing whether online and offline reinforcement-learning methods can improve on the $1/N$ benchmark.

[Insert Table 3 Here]

Table 3 reports baseline out-of-sample performance for the equally weighted portfolio, the vanilla online reinforcement-learning model, and the vanilla Decision Transformer. The equally weighted strategy generates an annualized return of 14.94%, a cumulative return of 28.86%, an annualized volatility of 15.59%, and an out-of-sample Sharpe ratio of 0.971. The vanilla online RL model produces an annualized return of 15.30% and a cumulative return of 29.61%, only modestly higher than the corresponding values for $1/N$. Its annualized volatility increases slightly to 15.82%, resulting in an out-of-sample Sharpe ratio of 0.979.

The difference in risk-adjusted performance is therefore economically small. Relative to the equally weighted benchmark, the vanilla online RL model improves the out-of-sample Sharpe ratio by only 0.008 and annualized return by approximately 0.37 percentage points. Thus, although the online RL strategy generates slightly higher raw and risk-adjusted performance, the improvement is limited relative to the additional complexity required to estimate and implement the dynamic policy.

Figure 4 plots the out-of-sample performance of the online reinforcement-learning specifications relative to the $1/N$ strategy. In the baseline comparison, the vanilla online RL model tracks the equally weighted portfolio closely, reinforcing the conclusion that conventional online reinforcement learning does not generate a material advantage over naive diversification before additional regime-conditioning information is introduced.

[Insert Figure 4 Here]

This result reinforces the importance of $1/N$ as a benchmark for learning-based portfolio strategies. The online RL agent has access to considerably more information than the naive portfolio and can dynamically alter allocations in response to changes in the financial state. Nevertheless, the baseline results indicate that this flexibility does not translate into a material out-of-sample advantage. The

small difference between the online RL out-of-sample Sharpe ratio and that of $1/N$ suggests that the additional information captured by the learned policy provides only limited incremental value in the baseline specification.

Importantly, the baseline comparison should not be interpreted as evidence that reinforcement learning is unable to improve portfolio choice. Rather, it indicates that the way in which the policy is learned matters. The next subsection shows that changing the learning architecture while maintaining the same underlying portfolio problem produces a considerably larger difference in out-of-sample performance.

6.4 Can Offline Reinforcement Learning Beat $1/N$?

The results differ more substantially for the offline Decision Transformer. The vanilla Decision Transformer generates an annualized return of 19.44%, a cumulative return of 38.20%, an annualized volatility of 16.18%, and an out-of-sample Sharpe ratio of 1.179. Relative to $1/N$, annualized return increases by approximately 4.50 percentage points and cumulative return by approximately 9.34 percentage points.

The improvement is not simply the consequence of taking substantially more risk. Annualized volatility increases from 15.59% for $1/N$ to 16.18% for the Decision Transformer, an increase of approximately 0.59 percentage points. The increase in returns is therefore considerably larger than the increase in volatility, producing an improvement in the out-of-sample Sharpe ratio of approximately 0.208 relative to the naive benchmark. This improvement is economically much larger than the corresponding increase generated by the online RL model.

[Insert Figure 5 Here]

Figure 5 illustrates the out-of-sample cumulative performance of the Decision Transformer specifications relative to the equally weighted $1/N$ benchmark. The vanilla Decision Transformer already exhibits stronger cumulative performance than $1/N$ before any macroeconomic regime-conditioning variables are introduced. This visual pattern is consistent with the summary statistics reported in Table 3, which show that the vanilla Decision Transformer achieves higher annualized return and stronger out-of-sample risk-adjusted performance than both the equally weighted benchmark and the vanilla online reinforcement-learning model.

The performance of the vanilla Decision Transformer is particularly informative because its state representation excludes the macroeconomic regime-conditioning variables examined in Section 7. Its

baseline advantage therefore cannot be attributed to GDP, unemployment, or oil-price information. Instead, the result points to the learning architecture itself: relative to the online actor–critic approach, the Decision Transformer learns portfolio actions from historical sequences of states, previous actions, and returns-to-go, allowing it to exploit temporal relationships embedded in the baseline financial information set.

This architecture produces an economically meaningful improvement in both return and out-of-sample risk-adjusted performance without requiring an expanded macroeconomic state. Table 3 summarizes this comparison. The equally weighted and vanilla online RL strategies display similar baseline performance, whereas the vanilla Decision Transformer generates a noticeably higher annualized and cumulative return with only a modest increase in annualized volatility. As a result, its out-of-sample Sharpe ratio is materially higher than those of the other two baseline strategies. Because all three portfolios are evaluated over the same out-of-sample period and under common investment constraints and transaction-cost assumptions, the table provides a clean comparison of the incremental value associated with the learning architecture.

6.5 Online versus Offline Reinforcement Learning

The direct comparison between the two learning architectures provides the clearest baseline result. The vanilla online RL strategy generates an out-of-sample Sharpe ratio of 0.979, only slightly above the $1/N$ benchmark, while the vanilla Decision Transformer achieves an out-of-sample Sharpe ratio of 1.179. Relative to the online RL model, the Decision Transformer increases the out-of-sample Sharpe ratio by approximately 0.200 and annualized return by approximately 4.13 percentage points. Cumulative return increases by approximately 8.59 percentage points, from 29.61% for the online model to 38.20% for the Decision Transformer.

The higher return of the Decision Transformer is accompanied by only a modest increase in annualized volatility. Volatility rises from 15.82% for the online model to 16.18% for the offline model, a difference of approximately 0.35 percentage points. The improvement in the out-of-sample Sharpe ratio therefore reflects primarily stronger returns rather than a large increase in portfolio risk. This result suggests that the offline learning architecture extracts more economically useful information from the baseline state representation than the online actor–critic model.

The visual comparison in Figures 4 and 5 reinforces this distinction. The online RL strategy remains close to the $1/N$ benchmark in the baseline specification, whereas the Decision Transformer

exhibits a more pronounced improvement in cumulative out-of-sample performance. The figures therefore complement the summary statistics by showing that the difference between the two learning architectures is not driven solely by one performance metric.

One possible explanation follows directly from the distinction between recursive policy learning and conditional sequence modeling. The online model repeatedly updates its actor and critic as it moves through the training environment, whereas the Decision Transformer learns the relationship among states, actions, and cumulative future rewards from fixed trajectories. Its self-attention mechanism allows the model to condition current portfolio decisions on information distributed across a longer historical context. The baseline results are therefore consistent with sequence-based learning providing a more effective mapping from historical financial information to out-of-sample portfolio actions in our setting.

[Insert Table 4 Here]

Table 4 summarizes the performance differences relative to $1/N$. The vanilla online RL model improves the out-of-sample Sharpe ratio by only 0.008, compared with an improvement of 0.208 for the Decision Transformer. The results highlight that the value of reinforcement learning is architecture-dependent: simply replacing naive diversification with a flexible online actor–critic model produces little improvement in the baseline setting, whereas reformulating portfolio learning as an offline sequence-modeling problem produces a materially stronger out-of-sample result.

The baseline evidence provides two conclusions that motivate the remainder of the empirical analysis. First, reinforcement learning does not automatically outperform naive diversification. The online actor–critic specification delivers out-of-sample performance that is very close to $1/N$, despite relying on a substantially more complex dynamic portfolio policy. Second, offline sequence learning performs considerably better in the same baseline environment. This finding shifts the emphasis from a simple comparison between sophisticated and naive portfolio strategies toward the architecture through which financial information is processed.

7 The Value of Regime Information

The baseline results show that the value of reinforcement learning depends strongly on learning architecture. We next test whether adding macroeconomic information improves out-of-sample performance by conditioning on WTI oil prices, U.S. GDP, and unemployment, individually and in

combination. Rather than imposing discrete regimes, we allow the models to use these variables as continuous signals of changing economic conditions. This design directly tests the information–complexity tradeoff: richer states may improve performance when added signals are informative, but may reduce it when additional dimensionality and redundancy outweigh their benefits.

7.1 Regime Information in Online Reinforcement Learning

We first examine how regime-conditioning information affects the online actor–critic model. Figure 4 plots the out-of-sample cumulative performance of the online RL specifications relative to the $1/N$ benchmark. The figure reveals substantial heterogeneity across information sets. Some macroeconomic signals materially improve performance, whereas other specifications underperform both the vanilla online model and naive diversification.

The cumulative performance paths make clear that the effect of regime conditioning is not simply a uniform upward or downward shift relative to the baseline model. Instead, different information sets lead the online agent to learn materially different allocation policies over the out-of-sample period. Specifications containing GDP tend to produce stronger cumulative performance, while models conditioned on unemployment or the full set of macroeconomic variables generally lag the benchmark. This visual evidence suggests that the incremental value of regime information depends on the specific signal provided to the learning architecture rather than on the mere expansion of the state space.

[Insert Table 5 Here]

Table 5 provides the corresponding return, volatility, and risk-adjusted performance measures underlying these cumulative paths. The vanilla online RL specification generates an annualized return of 15.30%, cumulative return of 29.61%, annualized volatility of 15.82%, and an out-of-sample Sharpe ratio of 0.979, compared with 0.971 for the $1/N$ benchmark. Once regime-conditioning variables are introduced, performance becomes substantially more dispersed across specifications.

GDP provides the strongest standalone regime-conditioning signal for the online model. Conditioning on GDP raises the annualized return to 20.84% and the out-of-sample Sharpe ratio to 1.167, compared with 0.979 for the vanilla online model and 0.971 for $1/N$. The Oil/GDP specification performs even better, producing an annualized return of 21.53%, cumulative return of 42.63%, and the highest out-of-sample Sharpe ratio among the online specifications, at 1.217. Relative to $1/N$, the out-of-sample Sharpe ratio improves by approximately 0.247 for Oil/GDP and 0.196 for GDP alone. These results indicate that selected macroeconomic signals can materially improve the portfolio policy

learned by the online agent.

The remaining specifications show, however, that the benefits of regime information are far from uniform. Oil alone produces an out-of-sample Sharpe ratio of 0.597, while unemployment alone yields 0.661. The Oil/Unemployment and GDP/Unemployment specifications generate out-of-sample Sharpe ratios of 0.932 and 0.769, respectively, both below the $1/N$ benchmark. Most notably, the specification containing all three regime-conditioning variables produces an out-of-sample Sharpe ratio of 0.569, the lowest among the online RL specifications and approximately 0.410 below the vanilla online model.

The online results are therefore strongly non-monotonic. GDP improves performance, and combining GDP with oil improves it further, yet expanding the information set to include all three macroeconomic variables substantially weakens out-of-sample performance. The evidence suggests that the value of additional state variables depends on their incremental economic content rather than simply on the amount of information available to the agent. In this setting, richer macroeconomic conditioning is beneficial only when the additional information contributes enough independent signal to offset the greater complexity of the state representation.

7.2 Regime Information in Offline Reinforcement Learning

We next examine whether the Decision Transformer responds differently to the same regime-conditioning information. Figure 5 presents the corresponding out-of-sample cumulative performance relative to the $1/N$ benchmark. As with online RL, performance varies meaningfully across information sets, but the ranking of the regime specifications differs across learning architectures.

The cumulative performance paths indicate that the Decision Transformer is comparatively strong even without explicit macroeconomic conditioning. The vanilla specification remains among the best-performing models throughout the out-of-sample period, while selected regime-conditioned specifications, particularly Oil/Unemployment and GDP, also perform well. At the same time, richer conditioning does not uniformly improve performance. The fully conditioned specification lags materially, reinforcing the view that the value of additional regime information depends on how effectively the architecture converts that information into portfolio decisions.

[Insert Table 6 Here]

Table 6 reports the corresponding return, volatility, and risk-adjusted performance measures. The vanilla Decision Transformer generates an annualized return of 19.44%, cumulative return of 38.20%, annualized volatility of 16.16%, and an out-of-sample Sharpe ratio of 1.172. It therefore remains one

of the strongest specifications even before macroeconomic information is added to the state.

Several regime-conditioned Decision Transformer specifications also perform strongly. Oil/Unemployment generates an out-of-sample Sharpe ratio of 1.190, outperforming the vanilla Decision Transformer’s 1.1722 and remaining well above the $1/N$ benchmark of 0.971. The GDP specification produces the highest annualized return among the Decision Transformer models, at 20.04%, although its higher annualized volatility of 17.58% results in a somewhat lower out-of-sample Sharpe ratio of 1.132. These results indicate that macroeconomic conditioning can meaningfully alter both return and risk, but explicit regime information is not necessary for the offline model to outperform naive diversification.

The comparison with the online results is particularly informative. GDP performs well under both architectures, but other combinations have substantially different effects. Oil/GDP produces the highest out-of-sample Sharpe ratio for online RL, at 1.217, but only 0.954 for the Decision Transformer. Conversely, Oil/Unemployment produces an out-of-sample Sharpe ratio of 1.190 for the Decision Transformer but only 0.932 for online RL. The economic value of a given information set therefore depends not only on the information itself but also on the architecture through which that information is processed.

The specification containing all three regime-conditioning variables again performs poorly. For the Decision Transformer, annualized return falls to 10.23%, annualized volatility rises to 17.78%, and the out-of-sample Sharpe ratio declines to 0.668, approximately 0.505 below the vanilla Decision Transformer. The same qualitative deterioration appears for online RL. This common pattern suggests that the weakest performance associated with the largest information set is not specific to one learning architecture, but instead reflects the broader information–complexity tradeoff examined in the paper.

7.3 When Does More Information Hurt?

The combined evidence reveals a central result of the paper: the relationship between regime information and out-of-sample portfolio performance is non-monotonic. Increasing the size of the information set does not necessarily improve portfolio decisions. Selected signals are valuable, but combining increasingly many regime-conditioning variables can materially reduce both returns and risk-adjusted performance.

For online RL, the out-of-sample Sharpe ratio declines from 0.979 in the vanilla specification to 0.569 when Oil, GDP, and Unemployment are included jointly. For the Decision Transformer, the corresponding decline is from 1.172 to 0.661. Thus, the specification containing the largest

macroeconomic information set performs substantially worse than the baseline model under both learning architectures.

This pattern is consistent with the information–complexity tradeoff developed in Section 3.3. Adding a regime-conditioning variable expands the information set available to the learner but simultaneously increases the dimensionality of the state representation. The marginal effect on portfolio performance is therefore positive only when the additional variable contributes sufficiently useful information to compensate for the increase in learning complexity.

Several mechanisms may contribute to this result. First, GDP, unemployment, and oil prices may contain overlapping information about economic conditions, so enlarging the state does not necessarily produce a proportional increase in independent information. Second, the variables differ substantially in persistence and frequency, with GDP and unemployment changing much more slowly than daily financial-market variables. Third, the deterioration in the fully conditioned Decision Transformer reflects both lower returns and higher volatility, indicating that the larger state does not merely induce a more conservative portfolio but instead produces less effective allocation decisions.

[Insert Table 7 Here]

Table 7 provides the clearest cross-architecture summary. It reports the out-of-sample Sharpe ratio for every information set under online RL and the Decision Transformer, together with the common $1/N$ benchmark.

For online RL, Oil/GDP delivers the highest out-of-sample Sharpe ratio (1.217), followed by GDP (1.167). For the Decision Transformer, the Oil/Unemployment model performs best (1.190), followed closely by the vanilla model (1.172). The ranking of information sets therefore varies materially across architectures, implying that regime information has no fixed value independent of the learning mechanism used to process it.

The central implication is that more information does not necessarily produce better portfolio performance. Successful portfolio learning requires selective rather than indiscriminate expansion of the state space. Although sophisticated architectures can process high-dimensional information, additional capacity does not eliminate the costs of redundant or weak signals. In our setting, selected macroeconomic variables improve out-of-sample performance, while richer combinations can reduce it. The value of regime information therefore depends jointly on its incremental economic content and on the architecture used to translate that information into portfolio decisions.

8 Robustness Tests

This section provides additional evidence on the robustness and economic interpretation of the main results. We examine whether performance survives higher transaction costs and turnover, and whether differences in learning architecture and regime information translate into distinct portfolio holdings. These tests show that the strongest specifications remain competitive after implementation costs and reflect economically meaningful allocation decisions.

8.1 Trading Frictions and Portfolio Turnover

Trading frictions are central to the comparison between reinforcement learning and naive diversification because dynamic strategies may require more portfolio adjustment than $1/N$. The baseline environment already incorporates proportional transaction costs, so the results in Sections 6 and 7 are net of the baseline implementation cost. We therefore examine whether the relative ranking of the principal strategies persists when transaction costs vary from 0 to 50 basis points, while also reporting portfolio turnover.

[Insert Table 8 Here]

Table 8 reinforces the main findings. Within the online RL family, the Oil+GDP specification remains the strongest performer and maintains an out-of-sample Sharpe ratio above 1.21 even at 50 basis points. Likewise, the Oil+Unemployment Decision Transformer remains competitive as implementation costs rise. These results show that the gains from selected regime-conditioning variables are not an artifact of negligible trading costs. At the same time, only selected regime combinations remain strong, consistent with the information–complexity tradeoff: additional information is valuable when it improves the state representation, but richer conditioning does not uniformly improve performance.

Turnover provides further evidence on the source of these gains. The best regime-conditioned online RL model generates approximately 80% more daily turnover than the vanilla online RL specification, yet its risk-adjusted performance remains strong as transaction costs increase. Because the online RL reward is defined net of the baseline 5-basis-point transaction cost, unnecessary reallocation directly reduces the agent’s reward. The persistence of the regime-conditioned model’s advantage therefore suggests that its stronger performance reflects economically meaningful portfolio adjustments rather than excessive trading.

The Decision Transformer provides complementary evidence on the role of learning architecture.

Turnover remains low for both the vanilla and regime-conditioned specifications, at 0.0078 and 0.0071, respectively, and adding regime information does not mechanically increase trading activity. Combined with the modest deterioration in Sharpe ratios as transaction costs rise, this result supports the broader finding that offline sequence learning can translate information into effective portfolio decisions without relying on frequent rebalancing. Overall, the transaction-cost evidence strengthens three central conclusions: reinforcement learning does not automatically outperform $1/N$, the value of regime information is selective rather than monotonic, and its economic usefulness depends critically on the architecture used to process it.

8.2 Portfolio Holdings and Learned Allocation Patterns

Portfolio holdings provide a direct view of how the learning models translate state information into investment decisions. Even when online RL and Decision Transformer specifications deliver similar aggregate performance, their underlying allocations can differ substantially. Table 9 therefore reports the ten largest holdings for the vanilla and best regime-conditioned Online RL and Decision Transformer models as of April 30, 2024, together with portfolio weights, shares, prices, and industry classifications.

[Insert Table 9 Here]

The portfolios depart meaningfully from $1/N$ and differ across both architectures and information sets. The vanilla online RL portfolio is highly concentrated in AutoZone (AZO), while the best regime-conditioned online RL model places its largest weight in Booking Holdings (BKNG) and reallocates toward AZO, TransDigm (TDG), Lam Research (LRCX), and Autodesk (ADSK). The Decision Transformer portfolios follow different patterns: the vanilla model is dominated by BKNG, whereas the best regime-conditioned specification places its largest weights in AZO, TDG, BKNG, MSCI, and Grainger (GWW). Although AZO, TDG, and BKNG recur across models, the remaining allocations differ materially, showing that learning architecture and regime information reshape the portfolio policy rather than simply rescale a common set of positions.

The sector composition provides additional economic interpretation. Shops, Manufacturing, High Technology, and Other are prominent across the four portfolios, whereas Energy is largely absent from the largest positions. Thus, conditioning on WTI does not mechanically increase exposure to energy producers. Instead, regime information appears to alter the relative attractiveness of firms across sectors. AZO is illustrative: although classified in Shops rather than Energy, its business environment

can be affected indirectly by fuel prices, transportation costs, and vehicle-maintenance demand. Its repeated appearance among the largest holdings suggests that macroeconomic information can influence allocations through firms outside the sector most directly associated with a given signal.

Overall, the holdings evidence reinforces the paper’s architecture and information–complexity results. The best regime-conditioned models select different securities and redistribute capital across industries relative to their vanilla counterparts, consistent with the finding that selected regime signals improve performance by changing how investment opportunities are ranked. At the same time, online RL and the Decision Transformer converge to distinct portfolio structures despite facing the same investment universe. The results therefore provide a portfolio-level mechanism for the main findings: reinforcement learning can outperform naive diversification when the learning architecture and state representation jointly generate more effective security selection and capital allocation.

9 Conclusion

This paper examines when reinforcement learning can outperform naive diversification in portfolio choice. We compare conventional online actor–critic reinforcement learning with an offline Decision Transformer and the equally weighted $1/N$ benchmark within a common portfolio environment. The analysis focuses on three determinants of out-of-sample performance: learning architecture, regime information, and trading frictions. The results show that reinforcement learning does not uniformly dominate $1/N$; its economic value depends on whether the learning architecture can convert financial information into persistent portfolio improvements that survive model complexity and implementation costs.

The baseline evidence reveals a clear distinction between online and offline learning. The vanilla online RL model performs only marginally better than the equally weighted benchmark, whereas the vanilla Decision Transformer delivers materially stronger out-of-sample return and risk-adjusted performance. Because all strategies use the same asset universe, portfolio constraints, and transaction-cost assumptions, these differences underscore the importance of learning architecture. In our setting, offline conditional sequence modeling provides a more effective mapping from historical financial states to portfolio decisions than the baseline online actor–critic approach.

The regime-information results show that expanding the information set does not monotonically improve performance. Selected macroeconomic signals improve risk-adjusted outcomes, but their value

depends on the architecture processing them. Certain combinations materially improve online RL, while the Decision Transformer remains strong even without explicit macroeconomic conditioning. In contrast, broader combinations of regime variables weaken performance under both architectures. The evidence therefore supports an information–complexity tradeoff in which additional state variables are valuable only when their incremental economic content outweighs the added dimensionality and redundancy they introduce.

The robustness evidence strengthens these conclusions. The strongest regime-conditioned strategies remain competitive as transaction costs increase, and turnover patterns indicate that their performance is not driven solely by aggressive rebalancing. Portfolio holdings further show that online RL and Decision Transformer models construct economically distinct portfolios and that regime conditioning changes both security selection and industry allocation. These results suggest that the observed performance differences reflect genuinely different learned policies rather than simple scaling of a common portfolio.

The broader implication is that the value of artificial intelligence in portfolio management depends less on how much information a model can process than on how effectively it can identify and translate useful information into portfolio actions. Future research could examine realized market regimes, alternative macroeconomic and market-state variables, richer offline-learning architectures, market-impact specifications, and rolling out-of-sample designs. An especially promising direction is to allow learning algorithms to endogenously select, compress, or dynamically weight state information, providing a more direct solution to the information–complexity tradeoff documented in this paper.

References

- Avramov, Doron, Si Cheng, and Lior Metzker. 2023. “Machine learning vs. economic restrictions: Evidence from stock return predictability.” *Management Science* 69 (5): 2587–2619.
- Black, Fischer. 1972. “Capital Market Equilibrium with Restricted Borrowing.” *The Journal of Business* 45 (3): 444–455.
- Chan, Nicholas Tung, and Christian Shelton. 2001. “An electronic market-maker.”
- Chen, Lili, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. “Decision Transformer: Reinforcement Learning via Sequence Modeling.” In *Advances in Neural Information Processing Systems (NeurIPS 2021)*.
- Chen, Luyang, Markus Pelger, and Jason Zhu. 2024. “Deep learning in asset pricing.” *Management Science* 70 (2): 714–750.
- Cong, Lin William, Kai Tang, Jingyi Wang, and Yifan Zhang. 2021. “AlphaPortfolio: Direct Construction Through Deep Reinforcement Learning and Interpretable AI.” SSRN.
- Cong, Lin William, Ke Tang, Jingyuan Wang, and Yang Zhang. 2020. “Deep Sequence Modeling: Development and Applications in Asset Pricing.” *The Journal of Financial Data Science* 3, no. 1 (December): 28–42.
- DeMiguel, Victor, Lorenzo Garlappi, and Raman Uppal. 2009. “Optimal Versus Naive Diversification: How Inefficient Is the 1/N Portfolio Strategy?” *The Review of Financial Studies* 22 (5): 1915–1953.
- Feng, Guanhao, Stefano Giglio, and Dacheng Xiu. 2020. “Taming the Factor Zoo: A Test of New Factors.” *The Journal of Finance* 75 (3): 1327–1370.
- Giglio, Stefano, Bryan Kelly, and Dacheng Xiu. 2022. “Factor Models, Machine Learning, and Asset Pricing.” *Annual Review of Financial Economics* 14:337–368.
- Gu, Shihao, Bryan Kelly, and Dacheng Xiu. 2020. “Empirical asset pricing via machine learning.” *The review of financial studies* 33 (5): 2223–2273.
- . 2021. “Autoencoder Asset Pricing Models.” *Journal of Econometrics* 222 (1): 429–450.

- Huang, Weichen, Arjun Sahm, and Andrew Y. Ng. 2024. “Q-Learning with Log-Barrier Regularization: Provable Sample Efficiency and Empirical Performance.” *arXiv* abs/2402.05272.
- Kelly, Bryan T., Seth Pruitt, and Yinan Su. 2019. “Characteristics Are Covariances: A Unified Model of Risk and Return.” *Journal of Financial Economics* 134 (3): 501–524.
- Li, Jiahao, Yong Zhang, Xingyu Yang, and Liangwei Chen. 2023. “Online portfolio management via deep reinforcement learning with high-frequency data.” *Information Processing & Management* 60 (3): 103247.
- Lintner, John. 1969. “The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets: A reply.” *The review of economics and statistics* 51 (2): 222–224.
- Liu, Yong, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. “Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting.” In *Advances in Neural Information Processing Systems*, 35:9881–9893.
- Markowitz, Harry. 1952. “Portfolio Selection.” *The Journal of Finance* 7 (1): 77–91.
- Merton, Robert C. 1973. “An Intertemporal Capital Asset Pricing Model.” *Econometrica* 41 (5): 867–887.
- Moody, John, and Matthew Saffell. 1998. “Reinforcement Learning for Trading.” In *Advances in Neural Information Processing Systems*, edited by M. Kearns, S. Solla, and D. Cohn, vol. 11. MIT Press.
- Mossin, Jan. 1966. “Equilibrium in a Capital Asset Market.” *Econometrica* 34 (4): 768–783.
- Neuneier, Ralph. 1997. “Enhancing Q-Learning for Optimal Asset Allocation.” In *Advances in Neural Information Processing Systems*, edited by M. Jordan, M. Kearns, and S. Solla, vol. 10. MIT Press.
- Pelger, Markus, and Ruoxuan Xiong. 2022. “State-Varying Factor Models of Large Dimensions.” *Journal of Business & Economic Statistics* 40 (3): 1315–1333.
- Pippas, Nikolaos, Elliot A. Ludvig, and Cagatay Turkay. 2025. “The Evolution of Reinforcement Learning in Quantitative Finance: A Survey.” *ACM Comput. Surv.* 57, no. 11 (June).
- Sharpe, William F. 1964. “Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk.” *The Journal of Finance* 19 (3): 425–442.

- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” In *Advances in Neural Information Processing Systems*, 30:5998–6008. Curran Associates, Inc.
- Yan, Xinyu. 2024. “Optimal Portfolio Allocation with Regime-Switching and Jump-Diffusion Dynamics: A Deep Reinforcement Learning Approach.” SSRN Electronic Journal, January 31, 2025.

Table 1: Summary Statistics

Table 1 reports summary statistics for the principal industry-level equities used in the reinforcement learning environment. Each industry reflects the Fama-French 10-industry classification, based on the four-digit Standard Industrial Classification (SIC), and is split into training and trading sets. The training set covers 3,396 time steps, from January 2nd, 2009 to June 30th, 2022. The trading set covers 459 time steps from July 1st, 2022 to April 30th, 2024. Daily returns and volatility show short-horizon performance and risk for a single trading day. Annualized return and volatility provide comparable measures of portfolio performance and risk.

Training Set Statistics										
	Hlth	Enrgy	HiTec	Shops	Manuf	Durbl	NoDur	Telcm	Other	Utils
Daily Ret. (%)	0.0006	0.0005	0.0009	0.0007	0.0006	0.0006	0.0006	0.0007	0.0007	0.0005
Ann. Ret. (%)	0.1890	0.1479	0.2595	0.2223	0.1911	0.1831	0.1655	0.2111	0.2056	0.1527
Daily Vol. (%)	0.011	0.021	0.013	0.011	0.013	0.017	0.010	0.012	0.015	0.011
Ann. Vol. (%)	0.1800	0.3418	0.2184	0.1892	0.2197	0.2856	0.1647	0.1959	0.2495	0.1812
Skewness	−0.421	−0.535	−0.254	0.345	−0.338	0.402	0.352	−0.169	0.003	−0.427
Min (%)	−0.102	−0.302	−0.120	−0.138	−0.117	−0.128	−0.101	−0.109	−0.142	−0.127
Max (%)	0.083	0.191	0.101	0.121	0.114	0.199	0.102	0.090	0.128	0.108
Count	33	11	54	36	43	6	25	8	128	29

Trading Set Statistics										
	Hlth	Enrgy	HiTec	Shops	Manuf	Durbl	NoDur	Telcm	Other	Utils
Daily Ret. (%)	0.0003	0.0007	0.0007	0.0006	0.0006	0.0003	0.0002	0.0000	0.0006	0.0001
Ann. Ret. (%)	0.1045	0.2227	0.2130	0.1817	0.1720	0.0988	0.0364	0.0094	0.1702	0.0281
Daily Vol. (%)	0.0104	0.0184	0.0126	0.0099	0.0118	0.0147	0.0097	0.0105	0.0119	0.0119
Ann. Vol. (%)	0.1657	0.2922	0.2000	0.1569	0.1874	0.2328	0.1534	0.1657	0.1881	0.1884
Skewness	0.1797	−0.1216	0.3217	0.0169	0.3012	0.1984	−0.0835	0.0516	0.3308	0.0049
Min (%)	−0.128	−0.101	−0.109	−0.142	−0.127	−0.044	−0.036	−0.036	−0.038	−0.042
Max (%)	0.045	0.061	0.070	0.040	0.058	0.058	0.027	0.041	0.063	0.047
Count	31	11	53	39	53	8	25	8	136	28

Table 2: Correlations Among Regime-Conditioning Variables

Table 2 reports pairwise correlations among the macroeconomic regime-conditioning variables used in the portfolio-learning models. The variables are WTI oil prices, U.S. GDP, and the U.S. unemployment rate.

	Oil	GDP	Unemployment
Oil	1.000	−0.295	0.243
GDP	−0.295	1.000	−0.716
Unemployment	0.243	−0.716	1.000

Table 3: Baseline Out-of-Sample Performance

Table 3 reports baseline out-of-sample performance for the equally weighted $1/N$ portfolio, the vanilla online reinforcement-learning model, and the vanilla Decision Transformer. The baseline learning specifications exclude the macroeconomic regime-conditioning variables introduced later in the analysis, so the comparison isolates differences associated with the learning architecture under a common information set. Ann. Ret. denotes annualized return, Cum. Ret. denotes cumulative return over the out-of-sample evaluation period, Ann. Vol. denotes annualized portfolio volatility, and OOS Sharpe denotes the out-of-sample Sharpe ratio relative to the risk-free rate. The equally weighted benchmark allocates $1/N$ of portfolio wealth to each of the 393 assets. Vanilla Online RL denotes the baseline online actor-critic specification using portfolio, market, and technical information, while Vanilla Decision Transformer denotes the corresponding offline sequence-learning specification. All strategies begin with the same initial portfolio value, are evaluated over the same out-of-sample period, and are subject to identical long-only, full-investment, position-limit, and transaction-cost assumptions.

Strategy	Ann. Ret. (%)	Cum. Ret. (%)	Ann. Vol. (%)	OOS Sharpe
Equally Weighted ($1/N$)	14.9354	28.8583	15.5898	0.9708
Vanilla Online RL	15.3022	29.6082	15.8244	0.9789
Vanilla Decision Transformer	19.2857	37.8797	16.1593	1.1722

Table 4: Performance Differences Relative to Naive Diversification

Table 4 reports differences in baseline out-of-sample performance between each reinforcement-learning strategy and the equally weighted $1/N$ benchmark. Δ Ann. Ret. denotes the difference in annualized return, while Δ Ann. Vol. denotes the difference in annualized portfolio volatility; both are reported in percentage points (pp). Δ OOS Sharpe is the difference between each strategy’s out-of-sample Sharpe ratio and that of $1/N$. Positive values indicate that the reinforcement-learning strategy produces a higher value than the naive benchmark. The vanilla specifications exclude macroeconomic regime-conditioning variables and use the common baseline portfolio, market, and technical information set. Vanilla Online RL denotes the baseline online actor-critic specification, while Vanilla Decision Transformer denotes the baseline offline sequence-learning specification. All strategies are evaluated over the same out-of-sample period and under identical long-only, full-investment, position-limit, and transaction-cost assumptions, so the reported differences isolate the incremental performance associated with each learning architecture.

Strategy	Δ Ann. Ret. (pp)	Δ Ann. Vol. (pp)	Δ OOS Sharpe
Vanilla Online RL	0.3667	0.2346	0.0081
Vanilla Decision Transformer	4.5007	0.5860	0.2082

Table 5: Performance of Regime-Conditioned Online Reinforcement Learning Strategies

Table 5 reports out-of-sample performance for the online actor-critic reinforcement-learning specifications under alternative regime-conditioning information sets. Ann. Ret. denotes annualized portfolio return, Cum. Ret. denotes cumulative return over the out-of-sample evaluation period, Ann. Vol. denotes annualized portfolio volatility, and OOS Sharpe denotes the out-of-sample Sharpe ratio relative to the risk-free rate. The vanilla specification contains portfolio, market, and technical information but no macroeconomic regime-conditioning variables. The remaining specifications augment the state representation with WTI oil prices, U.S. GDP, the U.S. unemployment rate, or combinations of these signals. The equally weighted portfolio provides the common $1/N$ benchmark. All strategies use the same asset universe, initial portfolio value, out-of-sample period, long-only and full-investment constraints, position limits, and transaction-cost assumptions. Differences across rows therefore isolate the effect of changing the macroeconomic information supplied to the online learning architecture.

Strategy	Ann. Ret. (%)	Cum. Ret. (%)	Ann. Vol. (%)	OOS Sharpe
<i>Regime-Conditioned:</i>				
Oil	8.3722	15.7712	15.4747	0.5967
Oil/GDP	21.5252	42.6309	17.2353	1.2174
Oil/Unemployment	14.0460	27.0478	15.5368	0.9317
GDP	20.8416	41.1728	17.5451	1.1668
GDP/Unemployment	11.5730	22.0746	15.8683	0.7693
Unemployment	9.5010	17.9770	15.5651	0.6608
GDP/Unemployment/Oil	8.8113	16.6270	17.5182	0.5693
<i>Without Regime Information:</i>				
Vanilla Online RL	15.3022	29.6082	15.8244	0.9789
<i>Naïve Benchmark:</i>				
Equally Weighted ($1/N$)	14.9354	28.8583	15.5898	0.9708

Table 6: Performance of Regime-Conditioned Decision Transformer Strategies

Table 6 reports out-of-sample performance for the Decision Transformer under alternative regime-conditioning information sets. Ann. Ret. denotes annualized return, Cum. Ret. denotes cumulative return over the out-of-sample evaluation period, Ann. Vol. denotes annualized portfolio volatility, and OOS Sharpe denotes the out-of-sample Sharpe ratio relative to the risk-free rate. The vanilla Decision Transformer contains portfolio, market, and technical information but no macroeconomic regime-conditioning variables. The remaining specifications augment the state representation with WTI oil prices, U.S. GDP, the U.S. unemployment rate, or combinations of these variables. The equally weighted $1/N$ portfolio provides the common benchmark. All strategies use the same asset universe, initial portfolio value, out-of-sample period, long-only and full-investment constraints, position limits, and transaction-cost assumptions. Differences across rows therefore isolate the effect of changing the macroeconomic information supplied to the offline sequence-learning architecture.

Strategy	Ann. Ret. (%)	Cum. Ret. (%)	Ann. Vol. (%)	OOS Sharpe
<i>Regime-Conditioned:</i>				
Oil	12.0177	22.9623	16.1272	0.7842
Oil/GDP	14.2583	27.4790	15.0306	0.9619
Oil/Unemployment	18.8039	36.8670	15.4880	1.1900
GDP	19.6347	38.6154	17.5754	1.1079
GDP/Unemployment	17.7639	34.6926	16.3591	1.0813
Unemployment	14.6536	28.2834	16.3354	0.9186
GDP/Unemployment/Oil	10.2298	19.4111	16.6617	0.6676
<i>Without Regime Information:</i>				
Vanilla Decision Transformer	19.2857	37.8797	16.1593	1.1722
<i>Naive Benchmark:</i>				
Equally Weighted ($1/N$)	14.9354	28.8583	15.5898	0.9708

Table 7: Regime Information & Out-of-Sample Sharpe Ratios Across Learning Architectures

Table 7 compares out-of-sample Sharpe ratios across the online actor-critic reinforcement-learning architecture and the offline Decision Transformer under alternative regime-conditioning information sets. The Vanilla row reports the common baseline specification containing portfolio, market, and technical information but no macroeconomic conditioning variables. Oil, GDP, and Unemployment denote specifications in which the corresponding macroeconomic signal is added to the state, while the remaining rows report pairwise and three-variable combinations. The equally weighted $1/N$ portfolio provides the common benchmark and therefore has the same out-of-sample Sharpe ratio in both columns.

Information Set	Online RL	Decision Transformer
Vanilla	0.9789	1.1722
Oil	0.5967	0.7842
GDP	1.1668	1.1079
Unemployment	0.6608	0.9186
Oil/GDP	1.2174	0.9619
Oil/Unemployment	0.9317	1.1900
GDP/Unemployment	0.7693	1.0813
Oil/GDP/Unemployment	0.5693	0.6676
Equally Weighted ($1/N$)	0.9708	0.9708

Table 8: Transaction-Cost Sensitivity and Portfolio Turnover

Table 8 reports portfolio turnover and out-of-sample Sharpe ratios under alternative proportional transaction-cost assumptions. Turnover measures the extent of portfolio reallocation generated by each strategy over the out-of-sample evaluation period. The columns labeled 0 bp, 5 bp, 10 bp, 25 bp, and 50 bp report OOS Sharpe ratios after imposing the corresponding transaction cost per unit of turnover. Vanilla Online RL and Vanilla Decision Transformer denote the baseline learning specifications without macroeconomic regime-conditioning variables. Best Regime Online RL and Best Regime Decision Transformer denote the strongest regime-conditioned specifications within each learning architecture based on the baseline out-of-sample analysis. The equally weighted $1/N$ portfolio is included as the common benchmark.

Strategy	Turnover	0 bp	5 bp	10 bp	25 bp	50 bp
Equally Weighted ($1/N$)	0.0000	0.9707	-	-	-	-
Vanilla Online RL	0.0044	0.9876	0.9789	0.9466	0.9401	0.9291
Best Regime Online RL	0.0079	1.2478	1.2174	1.2394	1.2316	1.2176
Vanilla Decision Transformer	0.0078	1.1871	1.1722	1.1665	1.1620	1.1544
Best Regime Decision Transformer	0.0071	1.2046	1.1900	1.2020	1.1968	1.1895

Table 9: Top Portfolio Holdings as of April 30, 2024

Table 9 shows the top portfolio holdings for the Vanilla Online RL model, the Best Regime Online RL model, the Vanilla Decision Transformer, and the Best Regime Decision Transformer. The holdings include details on the portfolio weight and number of shares held for each stock on the final date of our OOS performance. For perspective on the portfolio value, we also include the price per share on the final date of OOS trading. Expanding on the portfolio holdings potentially provides insight into how each RL agent chooses to trade in their respective environments.

Vanilla Online RL					Best Regime Online RL				
Ticker	Industry	Shares	Price	Weight	Ticker	Industry	Shares	Price	Weight
AZO	Shops	100	2956.40	22.81%	BKNG	Other	100	3405.15	23.88%
TDG	Manuf	100	1106.02	8.85%	AZO	Shops	47	2956.40	9.74%
ADBE	HiTec	124	462.83	4.43%	TDG	Manuf	101	1106.02	7.83%
TMO	HiTec	100	565.98	4.37%	LRCX	Manuf	954	87.93	5.88%
ROP	HiTec	100	506.94	3.91%	ADSK	HiTec	385	212.85	5.75%
AMP	Other	100	402.70	3.11%	DHR	HiTec	147	244.63	2.52%
SNPS	HiTec	71	530.59	2.91%	RCL	Other	259	137.69	2.50%
SYK	Hlth	100	331.95	2.56%	HUM	Other	100	296.36	2.08%
CAT	Manuf	100	327.08	2.52%	IDXX	Hlth	56	492.76	1.94%
SHW	Shops	100	295.16	2.23%	BLK	Other	35	726.17	1.78%

Vanilla Decision Transformer					Best Regime Decision Transformer				
Ticker	Industry	Shares	Price	Weight	Ticker	Industry	Shares	Price	Weight
BKNG	Other	15	3405.15	21.73%	AZO	Shops	44	2956.40	9.50%
TDG	Manuf	55	1106.02	4.49%	TDG	Manuf	116	1106.02	9.37%
GWV	Shops	64	907.74	4.21%	BKNG	Other	24	3405.15	6.00%
KLAC	HiTec	72	678.93	3.55%	MSCI	Other	120	455.94	4.00%
LMT	Manuf	100	443.33	3.22%	GWV	Shops	48	907.74	3.18%
VRTX	Hlth	88	392.81	2.51%	SPGI	Other	92	410.44	2.76%
V	Other	116	265.19	2.23%	MCK	Shops	64	533.08	2.49%
MA	Other	52	447.37	1.69%	MSI	HiTec	88	334.20	2.15%
SNA	Shops	84	255.81	1.56%	COST	Shops	40	717.17	2.10%
BLK	Other	27	726.72	1.42%	IDXX	Hlth	56	492.76	2.02%

Table 10: Top Portfolio Holdings by Industry

Table 10 reports industry-level portfolio holdings for the Vanilla Online RL model, the Best Regime Online RL model, the Vanilla Decision Transformer, and the Best Regime Decision Transformer as of April 30, 2024, the final date of the out-of-sample trading period. Industries follow the Fama–French 10-industry classification based on four-digit Standard Industrial Classification (SIC) codes. For each industry, we report its portfolio weight, annualized return and volatility, and the number of equities held. These industry-level allocations provide additional evidence on portfolio concentration and how each reinforcement-learning strategy distributes capital across sectors.

Vanilla Online RL					Best Regime Online RL				
Ind.	Weight	Ann. Ret.	Ann. Vol.	Count	Ind.	Weight	Ann. Ret.	Ann. Vol.	Count
Durbl	0.61%	26.29%	28.15%	1	Durbl	0.35%	5.80%	38.16%	2
Enrgy	0.00%	-	-	0	Enrgy	0.19%	-11.16%	29.96%	2
HiTec	27.46%	17.53%	28.19%	15	HiTec	12.41%	25.79%	34.67%	14
Hlth	5.09%	12.11%	21.69%	4	Hlth	4.56%	6.72%	31.88%	8
Manuf	19.29%	18.68%	27.94%	10	Manuf	19.02%	22.48%	27.84%	19
NoDur	2.47%	2.71%	21.71%	3	NoDur	2.53%	1.46%	30.00%	5
Other	15.07%	9.50%	31.41%	19	Other	46.01%	15.74%	28.81%	51
Shops	25.47%	24.52%	26.34%	6	Shops	13.35%	9.01%	29.60%	12
Telcm	2.57%	-7.49%	24.81%	2	Telcm	0.73%	3.05%	24.10%	5
Utils	1.96%	-2.03%	21.38%	3	Utils	0.85%	7.28%	25.33%	9

Vanilla Decision Transformer					Best Regime Decision Transformer				
Ind.	Weight	Ann. Ret.	Ann. Vol.	Count	Ind.	Weight	Ann. Ret.	Ann. Vol.	Count
Durbl	0.21%	-6.92%	33.63%	3	Durbl	0.58%	11.69%	26.39%	2
Enrgy	0.66%	15.84%	32.20%	5	Enrgy	1.58%	31.74%	33.37%	5
HiTec	17.5%	20.33%	30.92%	30	HiTec	8.87%	24.83%	29.36%	19
Hlth	7.74%	6.97%	26.36%	19	Hlth	7.41%	0.33%	32.57%	14
Manuf	14.45%	18.30%	29.14%	27	Manuf	16.83%	19.20%	29.36%	19
NoDur	1.71%	-5.80%	31.45%	10	NoDur	2.52%	2.49%	22.85%	8
Other	43.02%	9.33%	29.12%	66	Other	37.62%	14.30%	29.73%	68
Shops	11.25%	15.92%	26.50%	20	Shops	21.53%	19.62%	27.26%	15
Telcm	1.14%	-5.50%	25.02%	5	Telcm	1.13%	-4.96%	23.70%	5
Utils	2.32%	3.71%	23.81%	14	Utils	1.93%	0.65%	22.87%	12

Figure 1: Regime-Conditioning Variables and State-Space Complexity

Figure 1 illustrates how adding regime-conditioning variables expands the state representation used by the reinforcement-learning models. The left, middle, and right panels correspond to specifications containing one, two, and three macroeconomic regime-conditioning variables, respectively. The portfolio action space remains fixed at 393 assets across all specifications, while the number of state elements increases from 4,324 with one regime variable to 4,717 with two and 5,110 with all three.

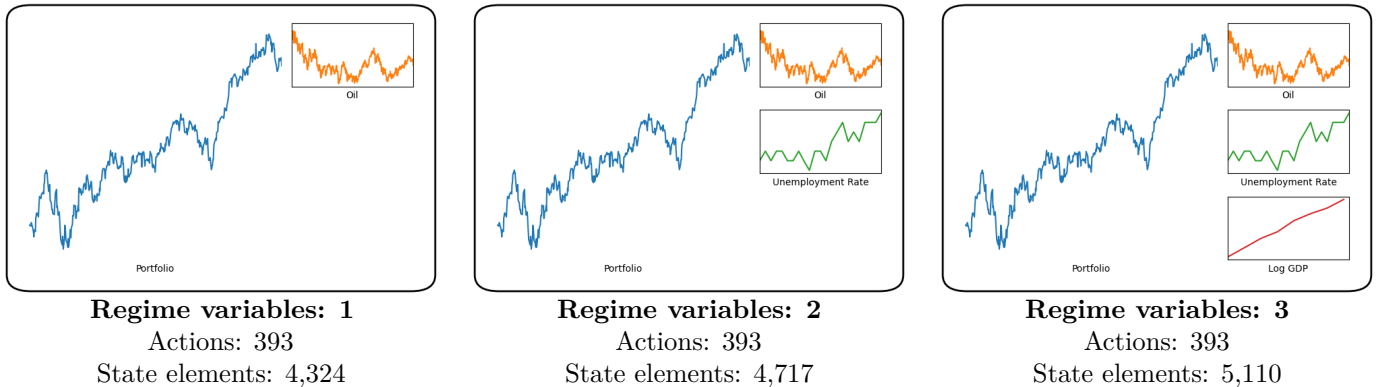


Figure 2: Architecture of the Decision Transformer

Figure 2 illustrates the Decision Transformer architecture used in the portfolio-learning framework. At each timestep, the return-to-go, financial state, and portfolio action are embedded into a common hidden representation and combined with timestep information before entering the causally masked Transformer blocks. The model uses a context length of 20 timesteps and a 128-dimensional hidden representation, allowing the self-attention mechanism to capture dependencies across historical states, actions, and target returns. The resulting output is mapped into admissible portfolio weights that satisfy the long-only and full-investment constraints imposed throughout the analysis.

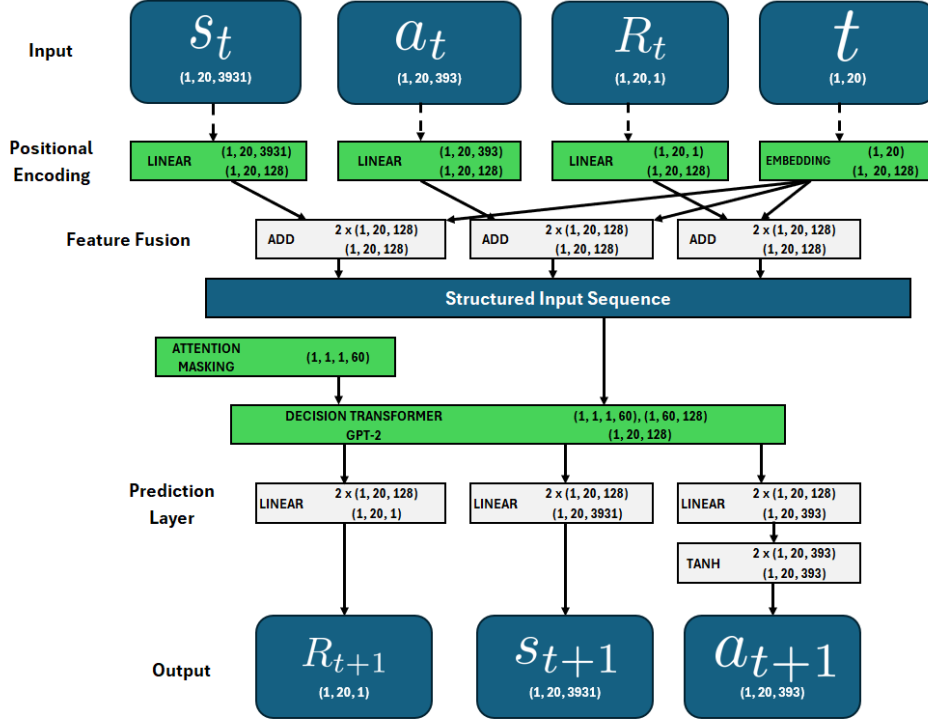


Figure 3: Correlation Matrix of the Reinforcement Learning Environment

Figure 3 presents the correlation matrix for the sector-level return series represented in the reinforcement-learning environment. Correlations are positive across all sector pairs, but their magnitudes vary substantially. Several economically related sectors exhibit relatively strong comovement, including Manufacturing, Other, Shops, and High Technology, while Utilities and Energy generally display lower correlations with the remaining sectors.

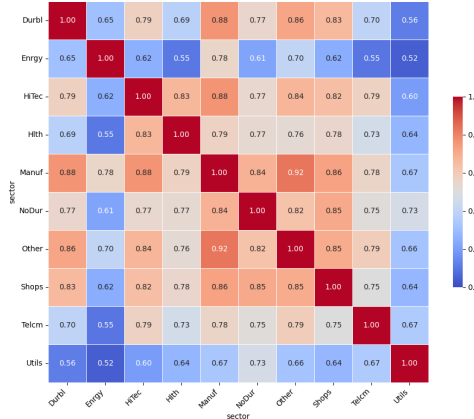


Figure 4: Out-of-Sample Performance of Online Reinforcement Learning Models versus the $1/N$ Strategy

Figure 4 compares the out-of-sample cumulative performance of the online reinforcement-learning specifications with the equally weighted $1/N$ benchmark. The online models include the vanilla specification and alternative versions conditioned on WTI oil prices, U.S. GDP, the U.S. unemployment rate, and their combinations. All strategies are evaluated over the same out-of-sample period and under identical long-only, full-investment, position-limit, and transaction-cost assumptions.

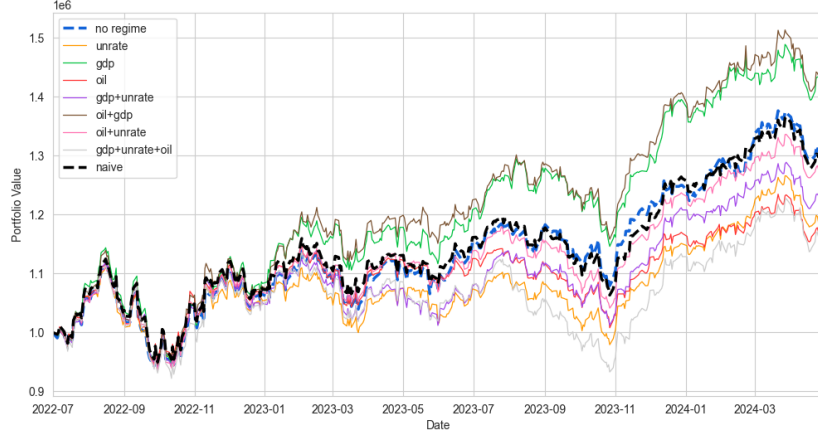
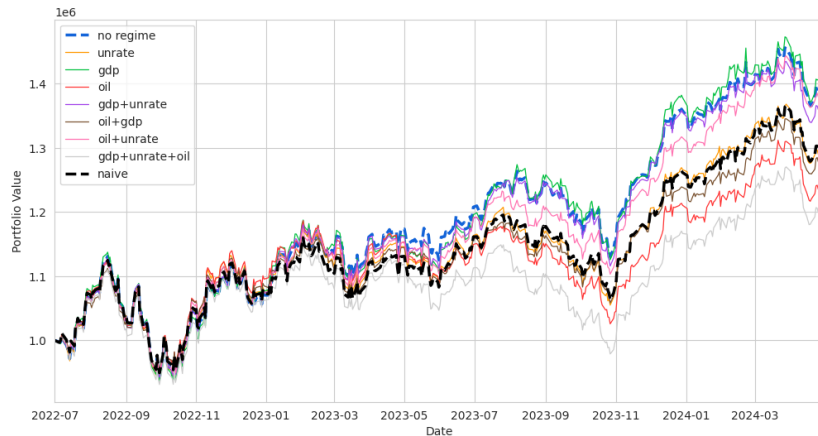


Figure 5: Out-of-Sample Performance of Decision Transformer Models versus the $1/N$ Strategy

Figure 5 compares the out-of-sample cumulative performance of the Decision Transformer specifications with the equally weighted $1/N$ benchmark. The models include the vanilla Decision Transformer and alternative specifications conditioned on WTI oil prices, U.S. GDP, the U.S. unemployment rate, and their combinations. All strategies are evaluated over the same out-of-sample period and under identical long-only, full-investment, position-limit, and transaction-cost assumptions. The figure shows that the Decision Transformer generally delivers stronger out-of-sample performance than the naive benchmark, although the magnitude of the improvement varies with the regime-conditioning information included in the state representation.



Online Appendix

When Does Reinforcement Learning Beat Naive Diversification? Learning Architecture, Regime Information, and Trading Frictions

A Additional Details on the Conceptual Framework

A.1 Trading Frictions and Portfolio Adjustment

The main text summarizes trading frictions through the net-performance relation in equation (21). This appendix provides additional details on the implementation of turnover and portfolio adjustment. Let $w_{i,t}$ denote the portfolio weight in asset i immediately before rebalancing and let $\pi_{i,t}$ denote the target weight selected by the learning agent. Portfolio turnover is defined as

$$TC_t = \sum_{i=1}^N |\pi_{i,t} - w_{i,t}|. \quad (46)$$

Given proportional transaction cost τ , portfolio wealth evolves according to

$$V_{t+1} = V_t (1 + r_{p,t+1}) (1 - \tau TC_t), \quad (47)$$

where portfolio return is

$$r_{p,t+1} = \sum_{i=1}^N \pi_{i,t} r_{i,t+1}. \quad (48)$$

The transaction-cost term penalizes policies that generate frequent or large reallocations. Thus, an RL strategy can outperform the $1/N$ benchmark on a net basis only when the improvement in gross portfolio performance is sufficiently large to offset the incremental costs generated by active rebalancing.

We additionally restrict changes in individual positions to prevent the learning algorithms from generating implausibly large trades. If $H_{\max}$ denotes the maximum number of shares that may be traded in an asset during one decision period, portfolio adjustment satisfies

$$|\pi_{i,t} - w_{i,t}| V_t \leq H_{\max} P_{i,t}, \quad i = 1, \dots, N. \quad (49)$$

This restriction bounds security-level turnover and links the learned allocation rule to a feasible

trading process.

The interaction between information, estimation complexity, and trading costs can therefore be summarized as

$$\mathcal{P}^{\text{net}} = f(\mathcal{A}, \mathcal{I}, \mathcal{C}, \mathcal{F}), \quad (50)$$

where $\mathcal{A}$ denotes learning architecture, $\mathcal{I}$ useful information, $\mathcal{C}$ state complexity, and $\mathcal{F}$ trading frictions. The framework predicts that reinforcement learning is most likely to outperform naive diversification when the additional information extracted by the learning architecture is sufficiently valuable to compensate for both estimation complexity and implementation costs.

B Additional Data and State Construction

B.1 Technical Indicators

The baseline financial state includes asset-level technical indicators intended to summarize momentum, volatility, and price trends. For N assets and M technical features, the indicator matrix at time t is

$$\mathbf{I}_t = \begin{bmatrix} T_{11,t} & T_{12,t} & \cdots & T_{1M,t} \\ T_{21,t} & T_{22,t} & \cdots & T_{2M,t} \\ \vdots & \vdots & \ddots & \vdots \\ T_{N1,t} & T_{N2,t} & \cdots & T_{NM,t} \end{bmatrix}, \quad (51)$$

where $T_{ij,t}$ denotes technical indicator j for asset i at time t . The baseline implementation contains eight technical features. These variables are supplied to the learning algorithms as state variables rather than interpreted as stand-alone trading rules.

The technical-information set includes the following measures:

- **Bollinger Bands:** lower, middle, and upper bands designed to summarize the position of current prices relative to a moving average and recent price variability.
- **Relative Strength Index (RSI):** a momentum measure comparing recent gains and losses over a rolling window.
- **Commodity Channel Index (CCI):** a measure of the deviation of current prices from their recent historical average.

- **Directional Index:** a measure of trend strength based on directional price movements.
- **Simple Moving Averages:** short- and longer-horizon averages of closing prices used to capture persistent price trends.

These indicators allow the learning algorithms to condition portfolio decisions on price dynamics beyond contemporaneous asset prices alone.

B.2 State Normalization

Because portfolio holdings, asset prices, technical indicators, and macroeconomic variables differ substantially in scale and persistence, continuous state variables are normalized before model estimation. For feature x , the standardized representation is

$$\tilde{x}_t = \frac{x_t - \mu_x}{\sigma_x}, \quad (52)$$

where μ_x and σ_x are estimated using the training sample. The same training-sample parameters are subsequently applied to validation and out-of-sample observations, preventing future information from entering the feature transformation.

For a collection of observations $\{x^{(i)}\}_{i=1}^M$, the sample mean and variance are

$$\mu_x = \frac{1}{M} \sum_{i=1}^M x^{(i)} \quad (53)$$

and

$$\sigma_x^2 = \frac{1}{M} \sum_{i=1}^M (x^{(i)} - \mu_x)^2. \quad (54)$$

For pairs of state variables, covariance and correlation are

$$\text{Cov}(x, y) = \frac{1}{M} \sum_{i=1}^M (x^{(i)} - \mu_x) (y^{(i)} - \mu_y) \quad (55)$$

and

$$\rho_{xy} = \frac{\text{Cov}(x, y)}{\sigma_x \sigma_y}. \quad (56)$$

Normalization addresses differences in scale but does not remove the information–complexity tradeoff. A standardized variable can still be redundant or economically uninformative, so the empirical

analysis separately varies the size and composition of the regime-conditioning information set.

B.3 State-Space Expansion

The baseline financial state contains portfolio information, asset prices, and technical indicators. Adding regime-conditioning variables increases the dimensionality of the state representation. In our implementation,

$$\dim(s_t) = \begin{cases} 3,931, & K = 0, \\ 4,324, & K = 1, \\ 4,717, & K = 2, \\ 5,110, & K = 3, \end{cases} \quad (57)$$

where K denotes the number of macroeconomic regime-conditioning variables. This progression provides the basis for the information–complexity analysis in the main text: additional variables may contain useful economic information, but they simultaneously increase the complexity of the state-action mapping that must be learned.

C Online Reinforcement-Learning Implementation

C.1 Actor–Critic Framework

The online reinforcement-learning models use actor–critic architectures. The actor determines the portfolio action,

$$a_t = \pi_\theta(s_t), \quad (58)$$

while the critic evaluates the expected value of the policy. The state-value function is

$$V^\pi(s_t) = \mathbb{E}_\pi \left[\sum_{k=0}^{T-t} \gamma^k r_{t+k+1} \mid s_t \right], \quad (59)$$

where $\gamma \in [0, 1]$ is the discount factor. The corresponding action-value function is

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi \left[\sum_{k=0}^{T-t} \gamma^k r_{t+k+1} \mid s_t, a_t \right]. \quad (60)$$

The advantage function is

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t). \quad (61)$$

A positive advantage indicates that the selected action performs better than the expected action under the current policy from the same state.

C.2 Advantage Actor–Critic

Advantage Actor–Critic updates the actor using an estimate of the advantage function. The policy-gradient update can be written as

$$\nabla_\theta J(\theta) = \mathbb{E} [\nabla_\theta \log \pi_\theta(a_t \mid s_t) A^\pi(s_t, a_t)]. \quad (62)$$

The critic reduces the variance of the policy-gradient update by evaluating whether realized rewards associated with observed portfolio decisions are better or worse than expected. In the portfolio setting, the actor therefore learns how the allocation vector should change as the financial state evolves.

C.3 Proximal Policy Optimization

Proximal Policy Optimization adds a constraint indirectly through its clipped objective:

$$L^{\text{CLIP}}(\theta) = \mathbb{E} \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \quad (63)$$

where $r_t(\theta)$ is denoted by

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \quad (64)$$

The clipping mechanism prevents the updated policy from moving too far away from the policy that generated the initial policy rollout.

C.4 Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient is designed for continuous action spaces and is therefore well suited to portfolio weights. The actor implements a deterministic policy,

$$a_t = \mu_\theta(s_t), \quad (65)$$

while the critic estimates

$$Q_\phi(s_t, a_t), \quad (66)$$

where ϕ denotes the critic parameters. The actor is updated in the direction that increases the critic’s estimated portfolio value:

$$\nabla_\theta J(\theta) \approx \mathbb{E} \left[\nabla_a Q_\phi(s, a) \Big|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s) \right]. \quad (67)$$

Both A2C and DDPG operate sequentially within the training environment, updating their policies as additional state-action-reward transitions become available.

D Decision Transformer Implementation

D.1 Self-Attention Mechanism

The main text presents the Decision Transformer as a conditional sequence model. This appendix provides additional details on the self-attention mechanism underlying the architecture. Given input token representations $x_1, \dots, x_n$, each token is mapped into query, key, and value vectors. The attention representation for token i is

$$z_i = \sum_{j=1}^n \alpha_{ij} v_j, \quad (68)$$

where

$$\alpha_{ij} = \frac{\exp(q_i^\top k_j / \sqrt{d_k})}{\sum_{\ell=1}^n \exp(q_i^\top k_\ell / \sqrt{d_k})}. \quad (69)$$

The attention weights allow the Transformer to assign different importance to observations within the historical context when determining the current portfolio action.

D.2 Token Construction and Sequence Representation

The Decision Transformer receives return-to-go, state, and action tokens. A trajectory is organized as

$$(R_1, s_1, a_1, R_2, s_2, a_2, \dots, R_T, s_T, a_T). \quad (70)$$

Each token type is projected into a common hidden space and combined with a timestep embedding that preserves temporal ordering. A causal attention mask ensures that predictions made at time t

cannot use information from subsequent states, actions, or rewards.

Our implementation uses a context length of 20 timesteps. Because each timestep contributes three token types, the maximum sequence contains 60 tokens. Return-to-go, state, and action embeddings are mapped into a 128-dimensional hidden representation before entering the Transformer blocks. The common representation allows self-attention to model interactions among desired portfolio outcomes, observed market states, and previous allocation decisions.

D.3 Portfolio-Action Output

The Transformer output is mapped into the portfolio-action space. The economically relevant prediction is the target allocation vector

$$a_t = \boldsymbol{\pi}_t = (\pi_{1,t}, \dots, \pi_{N,t})^\top. \quad (71)$$

The resulting portfolio weights satisfy

$$\pi_{i,t} \geq 0, \quad \sum_{i=1}^N \pi_{i,t} = 1, \quad (72)$$

so that the Decision Transformer operates under the same long-only and full-investment restrictions as the online RL models and the broader portfolio environment.

The vanilla Decision Transformer serves as the baseline offline-learning specification. Regime-conditioned variants preserve the same architecture while augmenting the financial state with oil prices, GDP, unemployment, or their combinations. Thus, performance differences across variants arise from changes in the information set rather than changes in the underlying Transformer structure.

E Model Variants and Experimental Structure

For both online and offline reinforcement learning, we consider the following information variants:

$$\mathcal{V} = \begin{cases} \text{Vanilla, Oil, GDP, Unemployment,} \\ \text{Oil + GDP, Oil + Unemployment,} \\ \text{GDP + Unemployment, Oil + GDP + Unemployment.} \end{cases} \quad (73)$$

The vanilla specification contains portfolio, price, and technical information without macroeconomic

regime-conditioning variables. Each subsequent specification expands the state representation while keeping the learning architecture fixed. The empirical design therefore crosses

$$\{\text{Learning Architecture}\} \times \{\text{Information Set}\}, \quad (74)$$

allowing the analysis to distinguish the effect of the learning method from the effect of the information supplied to that method.