AlphaPorfolio for Investment and economically Interpretable AI*

Lin William Cong   Ke Tang   Jingyuan Wang   Yang Zhang

First draft: December 2019; current draft: August 2020.

Abstract

We propose reinforcement-learning-based portfolio management, an alternative to the traditional two-step portfolio-construction paradigm (e.g., Markowitz, 1952), to directly optimize investors' objectives without estimating distributions of asset returns. Specifically, we modify cutting-edge AI tools such as Transformer to allow multi-asset sequence modeling that effectively captures the high-dimensional, non-linear, noisy, interacting, and dynamic nature of economic data. The resulting AlphaPortfolio yields stellar out-of-sample performances even after imposing various economic and trading restrictions. Importantly, we use polynomial-feature-sensitivity and textual-factor analyses to project the model onto linear regression and natural language spaces for greater transparency and interpretation. Such "economic distillations" reveal key market signals, firms' financials, and disclosure topics, including their rotation and non-linearity, that drive investment performance. Overall, we highlight the utility of reinforcement deep learning and provide a general procedure for interpreting AI and big data models in social sciences.

Keywords: Artificial Intelligence, Asset Pricing, Dynamic Programming, Machine Learning, Portfolio Theory, Neural Network, Textual Analysis.

*We are grateful to Agostino Capponi, John Cochrane, Andrew Karolyi, Andreas Neuhierl, Markus Pelger, Marcos de Prado, Michael Weber, Lu Zhang, Guofu Zhou, and conference and seminar reviewers and participants at the American Finance Association Annual Meeting (Chicago), Australasian Banking and Finance conference, Cheung Kong Graduate School of Business (CKGSB), Cornell University SC Johnson College of Business, Center for Research in Economics and Statistics (CREST) and École Polytechnique, Econometric Society World Congress (Milan), Georgetown University Global Virtual Seminar Series on FinTech "Machine Learning Day," Global Digital Economy Summit for Small and Medium Enterprises (DES2020), Global Quantitative and Macro Investment Conference (Wolfe QES), Alibaba Group Luohan Academy, 13th International Risk Management Conference (IRMC), Toulouse School of Economics, University of Virginia McIntire School of Commerce and Darden School of Business, Xi'an Jiaotong University, Yangtze River Delta International Forum "Corporate Finance and Financial Markets," and Zhongnan University of Economics and Law. We also thank Fujie Wang, Hanyao Zhang, and Guanyu Zhou for excellent research assistance. This research was funded in part by the Ewing Marion Kauman Foundation and INQUIRE Europe. The contents of this article are solely the responsibility of the authors. Cong is at Cornell University; Tang is at Tsinghua University; Wang and Zhang are at Beihang University. Contact author: Cong at will.cong@cornell.edu

1 Introduction

Modern portfolio theory as a staple in both academia and the investment industry entails first estimating population moments of asset returns from available samples and then choosing possible combinations of assets.[1] Such an approach has serious drawbacks due to estimation errors in the first step (e.g., DeMiguel, Garlappi, and Uppal, 2007). Furthermore, financial data, or data in social sciences in general, tend to be high-dimensional (high number of explanatory variables relative to observations), noisy, and non-linear, with complicated interaction effects and fast, non-stationary dynamics, rendering traditional econometric tools ineffective. Recently, researchers have started to adopt machine learning models to tackle the challenge. Yet, many applications indiscriminately use statistical packages or models designed for other disciplines with different data-generating processes, failing to maximize the strength cutting-edge AI offers. Importantly, the black-box nature of machine learning has precluded their wide use in social sciences where economic interpretation is integral.

Our first objective is therefore to take an alternative, direct optimization approach to portfolio management, aided by AI tools tailored for handling financial data, that overcomes the aforementioned challenges. Our key insight is that given the complexity of the real world, training a model through dynamic interactions with the environment can be more effective than attempting to estimate all assets' return distributions well regardless of their relevance for the portfolio construction.[2] To this end, we use reinforcement Learning (RL)—a machine learning methodology based on multi-arm bandit problem and dynamic programming that has proven to be effective at such tasks as seen in the wide range of AI applications such as computer vision, interactive games, and self-driving (e.g., Mnih, Kavukcuoglu, Silver, Rusu, Veness, Bellemare, Graves, Riedmiller, Fidjeland, Ostrovski, et al., 2015; Silver, Schrittwieser, Simonyan, Antonoglou, Huang, Guez, Hubert, Baker, Lai, Bolton, et al., 2017). We are among the first to tailor deep RL to a social science application with sound economic motivations and methodology innovations, identifying economic settings suitable for using RL and illustrating its efficacy.

Like many other applications of machine learning and AI models in finance and economics, our deep RL approach may be subject to critiques on the opaque nature of the

[1] Mean-variance optimization based on the investor's preference is one example (Markowitz, 1952). Practitioners and researchers also routinely construct portfolios by sorting based on asset characteristics and simple weight adjustments to manage the portfolio risk.

[2] After all, if the goal is to construct an optimal portfolios, any approach including ML and AI techniques should extract the signals that are the most relevant for portfolio construction (potentially under constraints). Although supervised learning and unsupervised learning have been widely applied in economics and finance, a direct optimization of investors' objectives require reinforcement learning because there is no labeled correct metrics for supervised learning. For example, if Sharpe ratio is the evaluation metric, there is no "correct" Sharpe ratio to speak of because we want it to be as high as possible.

algorithm and the lack of economic interpretation. Meanwhile, in a world divided by discrimination and injustice, it is insufficient to attribute all biases in AI to training data either; understanding models as a starting point for improving algorithmic fairness also constitutes a pressing issue.[3] Our second objective is then to introduce "economic distillation" that lends greater interpretability and transparency to machine learning models by projecting complex models onto linear modeling or natural language spaces. The polynomial-sensitivity and textual-factor analyses we devise not only provide insights into our AI model for portfolio management, but also apply to other AI applications in social sciences where economic interpretation, model impartiality, and mechanism transparency are indispensable.

Specifically, we add to variants of the latest sequence learning models such as the Transformer or Long Short-term Memory (LSTM) our novel cross-asset-attention networks (CAAN) that capture attribute interactions across assets. The use of deep neural networks also flexibly and effectively represents and extracts information from input features such as firms' fundamentals and market signals. Our emphasis is not on any specific functional form of the model, but the RL-based, data-driven approach that takes the joint distribution of asset returns as unknown, observes the outcomes of its interaction with the environment (e.g., realized Sharpe ratio), tests a range of actions in each state (various portfolio weights), and then dynamically explores a high-dimensional parameter space to maximize the investors' objective directly without an intermediate noisy step of modeling asset-return distributions. This "distribution/model free" approach can incorporate variable positions, transaction costs, risk aversion, path dependence etc., and exploits the Bellman principle and stochastic gradient descent (e.g., Friedman, 2002).

Our "direct optimization" improves portfolio performance out-of-sample drastically and the results remain robust after imposing various economic constraints. In an illustrative study of U.S. equities, we use a large number of firm characteristics and market signals as predictor variables, similar to Freyberger, Neuhierl, and Weber (2019). We focus on the out-of-sample Sharpe ratio and excess alphas not spanned by the traditional risk factors. With Sharpe ratio as a performance metric, we train a portfolio model (henceforth referred to as "AlphaPortfolio" or "AP") which generates a Sharpe ratio consistently above 2 on both the full test sample (1990-2016) and the subsamples excluding micro caps (10% or 20% based on market cap). The annualized excess alpha after controlling for various factors (CAPM, Fama-French-Carhart factors, Fama-French-Carhart plus liquidity factors, Fama-French five factors, Fama-French six factors, Stambaugh and Yuan factors, Q4 factors) are also typically above 10%. AP's other performance metrics such as turnover and maximum

[3]Society increasingly demands transparency and interpretability of algorithmic decisions (Goodman and Flaxman, 2017; Barocas, Hardt, and Narayanan, 2017, and articles 13-15 of European Union's General Data Protection Regulation (GDPR)).

drawdown are significantly lower than those of most regular anomaly strategies and known machine-learning strategies.[4]

RL-based AP consistently achieves an out-of-sample Sharpe ratio above 2 (reaching 4.7 in early testing years) compared to below 0.8 for a two-step implementation of Transformer Encoder or for the Fama-French five-factor model. Our findings are not driven by short positions or ad hoc weighing or particular industry sectors, and the outperformance remains significant with alternative turnover definition, after excluding unrated and downgraded firms, as well as restricting testing samples to more recent years or episodes of different market sentiment, volatility, and liquidity. AlphaPortfolio is therefore robust to imposing various economic restrictions that Avramov, Cheng, and Metzker (2019) identify to significantly hamper performances by other machine-learning-based strategies.

Our particular choice of Transformer Encoder (TE), a cutting-edge AI tool typically used for supervised machine translation, complements recent studies such as Nagel (2019) and Chen, Pelger, and Zhu (2020) in demonstrating that machine learning can help handle high-dimensional panel data in financial markets with complex nonlinear and interaction effects.[5] In particular, TE not only solves the vanishing and exploding gradient problems in Recurrent Neural Networks (RNNs), but also allows multi-sequence modeling for multiple assets once combined with our novel CAAN. Moreover, we show that the RL implementation more than doubles the performance of TE-CAAN under supervised learning. Given the robustness of our findings and the flexibility RL offers, our AI-driven model is conveniently deployable by practitioners and robo-advisors for trading and investment advising.

To better understand the AlphaPortfolio model, we use gradient-based methods and Lasso to distill the model into a linear model with a small number of input features, while allowing higher-order terms and feature interactions. This polynomial sensitivity analysis essentially projects a complex model onto a space of linear models. The distilled model informs us of features driving AP's performance. Besides some usual suspects such as Tobin's Q, features such as inventory changes (ivc) and changes in shares outstanding (delta_so) also play dominant roles. In addition, we find higher order terms (e.g, ivc^2) affect AP's behavior but not interaction effects (which could still be important for estimating assets' returns or a pricing kernel). Finally, we observe short-term reversals and identify important features dominant throughout and others rotating in and out. In particular, market trading signals and firms' fundamentals and financials take turns to dominate (correlation of -0.33).

We further apply textual factors analysis, an analytic combining the strengths of neural-network language processing and generative statistical modeling, to interpret the behavior of

[4]For example, the maximum drawdowns of common factors such as Fama-French-Carhart four factors over the same time period lie in the 40% to 60% over our sample period. Yet AP's is 2%-10%.

[5]We also build a LSTM model with CAAN and show the results remain robust.

AP based on texts from firms' filings. To project a complex model onto a natural language space, we follow Cong, Liang, and Zhang (2018) to generate textual factors by (i) representing texts using vector word embedding, (ii) clustering words using locality-sensitive hashing, and (iii) identifying spanning vector clusters through topic modeling. This data-driven approach captures complex linguistic structures while ensuring computational scalability and economic interpretability. We find that AlphaPortfolio buys stocks of firms whose 10-K and 10-Q talk about sales, profitability, loss-cutting, etc., whereas it short-sells stocks of firms that prominently mention real estates, mistakes, and corporate events, among others.

In general, our economic distillation (i) provides a sanity check on coding errors and model fragility, (ii) allows us to understand the economic mechanisms better so that we can avoid pitfalls of AI applications in economics, such as exacerbated discrimination, and (iii) enables practitioners to have a starting point for adjusting models when the market environment or policy changes. By introducing the concept of projecting black-box models onto simpler and more transparent models to make machine learning and AI applications in economics more interpretable, we complement attempts in the computer science field to extend explainable AI towards social sciences more generally.

We organize the remainder of the article as follows. Section 2 provides the background and clarifies our contributions as an interdisciplinary study; Section 3 describes our model and methodology; Section 4 applies the model to U.S. equities and reports the performance; Section 5 uses economic distillation to interpret the model; Section 6 discusses the general utility of reinforcement learning and interpretable AI in social sciences; Section 7 concludes; the appendices contain foundations of reinforcement learning, a description of variable construction, and an implementation of AlphaPortfolio using LSTM.

2 Related Literature and Contributions

As one of the first economic studies to apply AI to portfolio management, our paper makes three main contributions. (i) We develop a reinforcement-learning-based framework to directly optimize investors' objectives without intermediate estimations of assets' return moments, overcoming the challenges in conventional paradigms such as estimation errors and ad hoc weighing after characteristic-based sorting. Our approach particularly suits applications with unlabeled data, dynamic learning and decision-making, and environments so complex that a full specification is difficult. (ii) Our AlphaPortfolio outperforms most existing strategies (traditional or machine-learning-based), especially after imposing reasonable economic constraints and restrictions, demonstrating that cutting-edge AI tools typically used for supervised machine translation can be effective and immediately deployable in practice, once properly tailored to economic and financial applications. (iii) We provide general, expandable, and intuitive procedures for economically interpretable AI in social sciences that complement endeavors from computer science and machine learning fields.

2.1 Portfolio Theory and Investment Advising

Our paper foremost adds to portfolio theory. The trading strategy aids both institutional investors and retail investors (potentially as a second-generation robo-advisor), and more broadly provides insights on portfolio theory and applications of AI in finance.

Conventional paradigms. Following Markowitz (1952), the typical portfolio construction consists of two steps: (i) estimate population moments using available samples and (ii) optimize over possible combinations of assets, or simply sort and assign ad hoc weights.

Estimating returns accurately is extremely difficult due to the lack of long time series of data (e.g., Merton, 1980), while estimates of variance-covariance are rarely well-behaved (e.g., Green and Hollifield, 1992). This leads to unstable and extremely positive and negative weights in the second-step portfolio construction, giving poor out-of-sample performance and implementability in practice. This "error-maximizing" problem of the mean-variance portfolio (e.g., Best and Grauer, 1991) essentially derives from the fact that the second-stage optimization exploits too many small differences in the first-stage estimates without properly considering their estimation errors.

Models explicitly designed to mitigate the estimation errors include the Bayesian approach to estimation errors with diffuse-priors (Barry, 1974; Bawa, Brown, and Klein, 1979), "shrinkage estimators" (e.g., Jobson, 1979; Jorion, 1986), model-based priors (Pástor, 2000; Pástor and Stambaugh, 2000), robust portfolio allocation (Goldfarb and Iyengar, 2003; Garlappi, Uppal, and Wang, 2006), estimation risk, and optimal diversification (e.g., Klein and Bawa, 1976; Kan and Zhou, 2007).[6] But the estimation errors are so problematic that these attempts achieve very moderate success (DeMiguel, Garlappi, and Uppal, 2007).[7] Though

[6]Popular among practitioners are Black-Litterman models combining model-based priors with investors' subjective beliefs (Black and Litterman, 1990, 1992) and the "risk parity" approach (e.g., Jurczenko, 2015), which nevertheless suffers from instability the variance-covariance matrices are not always positive-definite for easy inversion (e.g., Bailey and Lopez de Prado, 2012; de Prado, 2016).

[7]The authors show that various models explicitly developed to deal with the estimation errors fail to beat the naive benchmark (each of the N assets available for investment gets a fraction $1/N$ of the total wealth at rebalancing) in terms of Sharpe ratio, certainty-equivalent return, and turnover, because in practice one has either very short estimation windows or small true Sharpe ratios of the efficient portfolio to start with or small portfolio sizes. Investors typically demand diversified portfolios and even for portfolios with no more than 50 assets, extant models are often estimated using five to ten years of data (instead of the decades of data the authors found necessary for reducing estimation errors sufficiently).

ad hoc sorting strategies based on asset characteristics avoid the estimation errors altogether, they are limited in handling high-dimensional features or capturing non-linear effects.

Instead of estimating the moments and parameters of the model under supervised learning, we directly optimize the portfolio's performance metric. Our findings demonstrate that AI tools can be a potential solution for the estimation error problem — hitherto the Achilles' heel of portfolio construction. Our approach is motivated by the possibility that (i) the relationship between the portfolio weights (as complicated functions of the return distribution) and the predictors could be less noisy than the relationship between the individual moments and the predictors, (ii) intermediate estimation of the return distribution may introduce additional noise and potential mis-specifications, and (iii) given the end goal, a global optimization, albeit imperfect, may work better than two local optimizations in the two steps.

Direct derivation of optimal portfolio weights. Brandt (1999) is among the earliest studies that focus directly on the dependence of the portfolio weights on the predictors rather than model the conditional return distribution. The simplest approach entails parametrized portfolio weights as functions of observables (e.g., Brandt and Santa-Clara, 2006; Brandt, Santa-Clara, and Valkanov, 2009), but suffer from potential mispecification of the portfolio weight function. While nonparametric or locally parametric estimators from sample analogues of the FOCs or Euler equations can guard against mispecification (Brandt, 1999), the curse of dimensionality when using kernel methods or polynomial expansions limits reliable implementation with more than two predictors (Brandt, 2010) unless one imposes linear structures such as variants of index regression (Powell, Stock, and Stoker, 1989; Aït-Sahali and Brandt, 2001).

All these studies still use supervised learning or restricted model structures such as linear regressions or methods of the moments, whereas our RL approach can incorporate more complex and dynamic environment, recognizing that high-dimensional and noisy financial data necessarily limit the estimation of asset return distributions and model specification. For example, our model picks up non-linear effects that Aït-Sahali and Brandt (2001) do not identify. While our model employs dynamic programming also seen in dynamic portfolio theory, we optimize investors' objectives directly without going through the explicit intermediate step of estimating asset returns or SDFs. Chen, Pelger, and Zhu (2020); Bryzgalova, Pelger, and Zhu (2020) focus on pricing kernel recovery and on using tree-based method to create basis assets for testing, which resembles the traditional approach, but they have optimizing a Sharpe ratio criterion in their tests and therefore achieve comparable Sharpe ratios in their portfolio constructions.

Robo-advising 2.0. Despite an outburst of media articles and industry reports discussing trends in robo-advising, current applications are limited in scope and functionality, and do little on artificial emotion, human behavior, active strategies, etc. (Lo, 2016; D'Acunto and Rossi, 2020).

Relative to the first-generation robo-advisors which mostly help clients avoid behavioral biases and manage asset allocation and factor exposure through trading ETFs, index mutual funds, and smart-beta products (Cong, Huang, and Xu, 2020), future robo-advisors likely automate more active strategies and customize the service according to individual investors' preference, tax situation, risk aversion, portfolio constraints, and transaction costs (e.g., Detemple and Murthy, 1997). Though not emphasized in our paper, the RL implementation can accommodate investors' specific preference and liquidity needs while economic interpretability offers greater transparency, meeting the demand of second-generation robo-advisors to aptly adjust models and convey investment principles to clients.[8]

The RL framework we adopt is useful for goals-based wealth management described in Dasa, Ostrova, Radhakrishnanb, and Srivastavb (2018) and followed by Betterment, a leading robo-adivisor with over US $14 billions asset under management. Another recent elegant application of RL to retail robo-advising is Alsabah, Capponi, Ruiz Lacedelli, and Stern (2019). Instead of using RL to develop trading strategies, the authors consider the tradeoff between exploration and exploitation and use RL to learn investors' preferences while providing investment advice simultaneously.

2.2 Machine Learning and Applications in Finance

Our paper contributes to an emerging literature that applies machine learning in economics for forecasting macroeconomic outcomes, asset returns, corporate defaults, risk exposures, etc., and for analyzing unstructured data such as texts.[9] Data in social sciences could differ drastically from data in science and engineering fields. Besides high dimensionality and non-linearity (e.g., Cochrane, 2011; Harvey, Liu, and Zhu, 2016; Karolyi and Van Nieuwerburgh, 2020) that ML packages from science and engineering help address, finan-

[8]Big data analytics and AI are deemed important components of the next-generation Robo-advisors. (EY "The evolution of Robo-advisors and Advisor 2.0 model" 2018, Scott Becchi, Ugur Hamaloglu, Taroon Aggarwal, Samit Panchal.) For example, Numerai, an AI-based hedgefund with native token Numerarie, already plans for an app Daneel that combines robo-advising and personal assistant. Sharpe Capital is another app that use ML. Users typically expect to understand what robo-advisors do at a high level.

[9]Cochrane (2011) specifically called for methods beyond cross-sectional regressions and portfolio sorts, which has contributed to the rise of machine learning applications in asset pricing. Rapach, Strauss, and Zhou (2013) is among the earliest contributions in finance; Gu, Kelly, and Xiu (2018) and Bianchi, Büchner, and Tamoni (2019) compare ML methods for predicting stock and bond returns; De Prado (2018) gives a detailed introduction.

cial data are often characterized by low signal-to-noise ratio, significant interaction effects, and non-stationary/fast dynamics.[10] Moreover, machine learning with the causal inference that economists emphasize is also in its nascence (e.g., Athey and Imbens, 2017; Wager and Athey, 2018).

Existing studies typically adopt a supervised approach with several non-mutually exclusive lines of work. One line discusses dimension reduction (in a linear framework or with transparent models) either using regularization methods such as Lasso, Ridge, or Elastic Net (e.g., Rapach and Zhou, 2019; Feng, Giglio, and Xiu, 2020), or rotation and clustering techniques such as PCAs (e.g., Kelly, Pruitt, and Su, 2019; Kozak, Nagel, and Santosh, 2020; Kim, Korajczyk, and Neuhierl, 2019; Chinco, Neuhierl, and Weber, 2019). A second line aims at capturing interactions and non-linear effects through semi-parametric, distribution-free, or flexible but complex model architectures such as group LASSO, splines, and ensemble learning (e.g., Freyberger, Neuhierl, and Weber, 2019; Light, Maslov, and Rytchkov, 2017; Rossi, 2018; Moritz and Zimmermann, 2016). These tools, together with the aforementioned dimension reduction methodologies, are traditionally categorized as general statistical learning, which was developed in the 1980s in statistics to address the noise, high-dimensionality, non-linearity, and strong interactions in the data.

Modern machine learning mostly concerns neural-network-based deep learning that only gained dominance over the past decade. It is quickly adopted by asset pricers for studying pricing kernels and reducing estimation errors in return distributions (e.g., Feng, Polson, and Xu, 2018; Gu, Kelly, and Xiu, 2019). However, when applied to portfolio management, it is unclear if machine learning's superior performance is driven by microcaps and value weighing, as is the case with traditional anomalies (Hou, Xue, and Zhang, 2020; Avramov, Cheng, and Metzker, 2019). Most finance applications of deep learning focus on simple neural networks without involving sequence modeling, with Feng, He, and Polson (2018); Chen, Pelger, and Zhu (2020); Cong, Tang, Wang, and Zhang (2020) being among the exceptions. Missing from the current literature are direct portfolio construction, use of RL and cutting-edge AI tools, as well as interpretation of complex ML models, which we focus on.[11]

Admittedly, there is a literature in computer science and statistics applying RL or deep learning to investment (e.g. Ding, Liu, Bian, Zhang, and Liu, 2018), including learning investors' risk preference (Alsabah, Capponi, Ruiz Lacedelli, and Stern, 2019). But the

[10]Nagel (2019) discusses the Lucas critique in finance settings. While hotdogs do not change their shape in response to image classification, investors alter their behaviors after others' using ML tools. The low signal-to-noise also necessitates the use of out-of-sample performance instead of in-sample predictability Martin and Nagel (2019).

[11]Among notable exceptions regarding interpreting complex ML models, Sak, Huang, and Chng (2019) uses characteristic sorts and Stambaugh and Yuan (2017)'s mispricing factors to identify monthly dominant characteristics and ascertain the ex-post source of alpha.

conventional value-based approaches built on Q-learning do not capture the complexity of financial markets and the continuum action space of trading (e.g., Neuneier, 1996; Jin and El-Saawy, 2016a); extant policy-based approaches similar to ours overcome the limitation (e.g., Moody, Wu, Liao, and Saffell, 1998; Deng, Bao, Kong, Ren, and Dai, 2016), but often ignore proper economic benchmarks or fail to control for risk factors (several focus on cryptocurrency markets that do not even have such benchmarks and well-known risk models to begin with), not to mention that the models are not adjusted for multi-asset investment problems. Moreover, most studies on trading leave out economic motivations of the models, interpretability and stability of the strategies, or discussions of robustness against economic and trading restrictions, which limit their application in financial markets and research in social sciences (for a survey, see Fischer, 2018).

To our knowledge, our paper is the first reinforcement deep learning study in finance that is motivated economically by the challenges observed in portfolio theory and practice, as well as the need for interpretability.[12] We build on Transformer and LSTM specifically designed to handle sequential information, which excels in extracting complex information in time-series data. More fundamentally, the RL implementation allows us to separate from other machine learning models that estimate distributions of asset returns and construct portfolios through the two-stage process prone to estimation errors. None of the prior RL studies treats assets as multiple sequences (Wang, Zhang, Tang, Wu, and Xiong, 2019). We innovate by devising the CAAN to incorporate linkages among sequence of assets' features and characteristics. Earlier studies also do not focus on economic interpretability, a point Karolyi and Van Nieuwerburgh (2020) emphasize and we investigate.

2.3 Interpretable AI and Data Science

Economists have recently started to discuss the socioeconomic implications of AI and data science, such as discrimination, data privacy, and macroeconomic outcomes (e.g., Bartlett, Morse, Stanton, and Wallace, 2019; Liu, Sockin, and Xiong, 2020; Farboodi, Veldkamp, et al., 2019). Underlying these issues is that data have massively expanded in volume, velocity, and variety, and tools for analyzing them have become too complex to inspect or understand.[13]

[12] Koray Kavukcuoglu, the director of research at Deepmind, is quoted for saying: "Reinforcement Learning is a very general framework for learning sequential decision making tasks. And Deep Learning, on the other hand, is of course the best set of algorithms we have to learn representations. And combinations of these two different models is the best answer so far we have in terms of learning very good state representations of very challenging tasks that are not just for solving toy domains but actually to solve challenging real world problems" (Garychl, 2018).

[13] Lasso, ridge, elastic net do not suffer much from interpretability issues in terms of model transparency. The main challenge comes with modern ML methods such as deep neural networks. That said, even when the models are simple and transparent, the input features may not be economically interpretable. That is

As such, economic interpretability has become critical when applying machine learning or big data anlytics in social sciences. Our study is among the earliest to emphasize interpreting deep reinforcement learning models in finance and economics.

Our paper is related to an emerging literature in computer science and machine learning on model compression or distillation (e.g., Bucilu, Caruana, and Niculescu-Mizil, 2006; Hinton, Vinyals, and Dean, 2015). Unlike their distillations that typically still have a large set of features and are aimed at deployment and computational efficiency, our "economic distillation" advocates using the original complex model for prediction and inference and using the distilled model to guide parameter tuning when, for example, an economy enters a fiscal and monetary policy regime that was never experienced before. Our objective completely differs and focuses on understanding the underlying mechanism from a cumbersome or opaque model to a familiar model with not only simpler structure but also a smaller set of input features so that it is suitable for economic interpretation.

Our approach also falls into the area of explainable AI (XAI, e.g., Guidotti, Monreale, Ruggieri, Turini, Giannotti, and Pedreschi, 2018; Horel and Giesecke, 2019a). From the functional perspective, *local XAI* tries to understand why the model makes a specific decision for a certain input, whereas *global XAI* tries to elucidate the model logic and rules. From the methodological perspective, XAI entails either *surrogate model* or *feature importance extraction*. Surrogate models use decision trees, rule sets, linear or generalized additive models, etc., to proxy the neural network to be interpreted (e.g., Wu, Hughes, Parbhoo, Zazzi, Roth, and Doshi-Velez, 2018; Ribeiro, Singh, and Guestrin, 2016). Feature importance extraction focuses on analyzing contributions of feature inputs to model outcomes, including gradient-based sensitivity analysis (e.g., Sundararajan, Taly, and Yan, 2017; Wang, Zhang, Tang, Wu, and Xiong, 2019), Partial Dependence Plots (PDPs, e.g., Krause, Perer, and Ng, 2016), and significance tests for single-layer neural networks Horel and Giesecke (2019b).

We contribute conceptually by introducing projections of AI models onto relatively transparent and interpretable modeling spaces. Methodology-wise, we are the first to expand feature sensitivity analysis and combine it with surrogate modeling to achieve global interpretability. Our polynomial sensitivity analysis thus improves upon conventional gradient-based methods and captures higher-order and interaction effects in economic data that the above approaches miss.[14] Moreover, we complement our polynomial sensitivity analysis with textual-factor analysis to enhance economic interpretability, which is integral to research in

where textual data help (e.g., Cong, Liang, and Zhang, 2018; Cong, Liang, Yang, and Zhang, 2019).

[14]As two exceptions, Datta and Sen (2018) build on the concept of Shapley value to develop a Quantitative Input Influence method, and demonstrate input influences can be summarized using clustering approaches, and Horel and Giesecke (2019a) develop computationally efficient feature significance test that can not only identify feature interactions of any order but also generate model-free feature importance measures. However, these approaches are not designed for RL-based applications.

social sciences.[15] To our knowledge, we are the first to use texts to improve economic interpretability of big data and AI models.

3 Model and Methodology

We start with a Transformer Encoder (TE), which is a variant of the Transformer model proposed by Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin (2017).[16] For each asset, we use the TE to extract and represent information from its history states/features. To describe interrelationships among the assets, we then use Cross-Asset Attention Network (CAAN), which takes the representations of all assets as inputs, and estimates a winner score for every asset to be used in decisions when constructing a portfolio. We then frame this AlphaPortfolio strategy into a reinforcement learning game to train the model parameters to maximize an evaluation criterion, such as the out-of-sample Sharpe ratio.

The Transformer model constitutes the most cutting-edge approach in sequence modeling and has been recently used in neural machine translation. Like recurrent neural networks (RNNs), Transformers are designed to handle ordered sequences of data and capture history dependence. But unlike many RNNs (including LSTM), Transformers do not require that the sequence be processed in order. Transformer makes long-range dependencies in sequences easier to learn by reducing network path length and allows for more parallelization by reducing the reliance on the prohibitive sequential nature of inputs. We describe the development of deep sequence modeling in Cong, Tang, Wang, and Zhang (2020) and the basics of reinforcement deep learning in Appendix A.[17]

[15] As Athey (2018) aptly puts, "extensions and modifications of prediction methods to account for considerations such as fairness, manipulability, and interpretability to be among the very first changes to emerge concerning how empirical work is conducted." This is exactly where our economic distillation procedures add to both computer science and economics.

[16] As discussed in Cong, Tang, Wang, and Zhang (2020), deep sequence modeling is needed for analyzing time series data. However, methods based on recurrent neural networks (RNNs) preclude parallelization within training samples, which becomes an issue with longer sequences. Transformer instead relies solely on self-attention to compute representations of its inputs and outputs, dispensing with sequence-aligned RNNs or convolution entirely, making long-range dependencies in sequences easier to learn and allows for more parallelization (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin, 2017). It follows the encoder-decoder architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. Each encoder processes its input vectors to generate encodings and passes its set of generated encodings to the next encoder or decoder as inputs. Each decoder does the opposite, using contextual information incorporated in the encodings to generate an output sequence. The attention mechanism then allows weighing the relevance of every input and drawing information accordingly when producing outputs. The encoder-decoder structure is very common. For example, Gu, Kelly, and Xiu (2019) has the namesake but is a completely different model from ours.

[17] We find that a LSTM-CAAN-based model may yield greater performance in terms of Sharpe ratio than

Why one-step portfolio optimization using reinforcement learning? As we described in Section 2, one-step optimization likely works better than the combination of two indirect optimizations. Moreover, reinforcement learning can better handle complex environment and incorporate investors' long-term objective, which allows incorporating budget constraint, long-term goal, etc. In addition, deep learning helps to handle high dimensionality and nonlinear effects with computational efficiency and model stability.

3.1 Asset Representations Extraction using Transformer Encoder

The return distribution of an asset has close relationships with its history states. In AlphaPortfolio, we first use a TE to represent the time series of each asset's features. Figure 1 illustrates the architecture of a plain-vanilla TE. Here the encoder is composed of a stack of several identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, which we modify in AlphaPortfolio, and the second is a simple position-wise fully connected feed-forward network. In addition, residual connection and layer normalization are employed for each sub-layer. We elaborate further in Section 3.5.

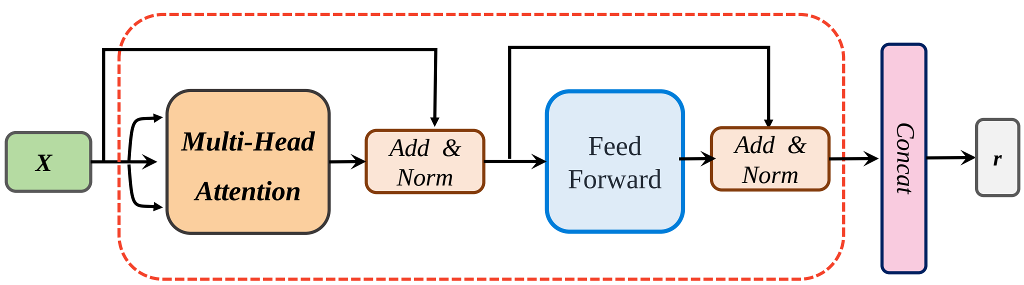

Figure 1: The Architecture of Transformer Encoder.

Because TE takes sequential states/features as inputs, we therefore organize assets' raw features into sequential form. We use vector $\tilde{\boldsymbol{x}}_t^{(i)}$ to denote the history state of an asset i at time t, which consists of asset features/firm characteristics, for example, as given in Section 4.1. We name the last K historical holding periods at time t, *i.e.,* the period from time $t-K$ to time t, as a *look-back window* of t. One example is features from the previous 12 months when we construct a portfolio for the 13th month. The history states of an asset in the look-back window are denoted as a sequence $\boldsymbol{X}^{(i)} = \left\{\boldsymbol{x}_1^{(i)}, \cdots, \boldsymbol{x}_k^{(i)}, \cdots, \boldsymbol{x}_K^{(i)}\right\}$, where

using Transformer. But when we dive into the interpretability, we find the algorithm suffers from exploding gradient problems in the test sample (not the training sample). We report the empirical results in Appendix C and focus on Transformer Encoder (TE) and CAAN as our main model, which achieves a balance between performance and interpretability.

$\boldsymbol{x}_k^{(i)} = \tilde{\boldsymbol{x}}_{t-K+k}^{(i)}$. Our model uses TE to encode $\boldsymbol{X}^{(i)}$ into vector space

$$\boldsymbol{Z}^{(i)} = \mathrm{TE}\left(\boldsymbol{X}^{(i)}\right), \tag{1}$$

where $\boldsymbol{Z}^{(i)} = \left\{\boldsymbol{z}_1^{(i)}, \cdots, \boldsymbol{z}_k^{(i)}, \cdots, \boldsymbol{z}_K^{(i)}\right\}$. The $\boldsymbol{z}_k^{(i)}$ is the hidden state encoded at step k, which takes all other steps into consideration. We concatenate all the steps in $\boldsymbol{Z}^{(i)}$ as a representation of the asset: $\boldsymbol{r}^{(i)} = \mathrm{Concat}\left(\boldsymbol{z}_1^{(i)}, \ldots, \boldsymbol{z}_k^{(i)}, \ldots, \boldsymbol{z}_K^{(i)}\right)$, which contains the global dependence among all elements in $\boldsymbol{X}^{(i)}$. In our model, the representation vector for all assets are extracted by the same TE, which means the parameters are shared by all assets. In this manner, the representations extracted by TE are relatively stable and generally applicable for all assets available rather than for a particular one.

As mentioned earlier, learning long-range dependencies is a key challenge when using recurrent neural networks, *i.e.,* RNN and LSTM. The Transformer architecture connects all positions in the sequence, which can effectively extract both short-term and long-term dependencies.

3.2 Cross-asset Attention Network & Winner Score Estimation

In the traditional RL-based strategy models, the investment portfolio is often directly generated from the asset representations through a softmax normalization (Jin and El-Saawy, 2016b; Deng, Bao, Kong, Ren, and Dai, 2017; Ding, Liu, Bian, Zhang, and Liu, 2018). The drawback of this approach is that it does not fully exploit the interrelationships among assets. In light of this, we propose a CAAN to describe the interrelationships among assets. Note that our design of CAAN model is inspired in part by the self-attention mechanism proposed by Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin (2017), which Transformer models use. But the application is different, as we explain in this section and Section 3.5.

Figure 2 illustrates the architecture of CAAN. Specifically, given the asset representation $\boldsymbol{r}^{(i)}$ (we omit time t without loss of generality), we calculate a query vector $\boldsymbol{q}^{(i)}$, a key vector $\boldsymbol{k}^{(i)}$ and a value vector $\boldsymbol{v}^{(i)}$ for asset i as

$$\boldsymbol{q}^{(i)} = \boldsymbol{W}^{(Q)}\boldsymbol{r}^{(i)}, \quad \boldsymbol{k}^{(i)} = \boldsymbol{W}^{(K)}\boldsymbol{r}^{(i)}, \quad \boldsymbol{v}^{(i)} = \boldsymbol{W}^{(V)}\boldsymbol{r}^{(i)}, \tag{2}$$

where $\boldsymbol{W}^{(Q)}$, $\boldsymbol{W}^{(K)}$ and $\boldsymbol{W}^{(V)}$ are the matrices of parameters to learn. The interrelationship of asset j to asset i is modeled as using the $\boldsymbol{q}^{(i)}$ of the asset i to query the key $\boldsymbol{k}^{(j)}$ of asset j, *i.e.,* the re-scaled inner product similarity between $\boldsymbol{q}^{(i)}$ and $\boldsymbol{k}^{(j)}$:

$$\beta_{ij} = \frac{\boldsymbol{q}^{(i)\top} \cdot \boldsymbol{k}^{(j)}}{\sqrt{d_k}}, \tag{3}$$

where d_k is a re-scale parameter to avoid the dot-product from becoming too large.[18] Then, we use the normalized interrelationships $\{\beta_{ij}\}$ as weights to sum the values $\{\boldsymbol{v}^{(j)}\}$ of other assets into an attenuation score:

$$\boldsymbol{a}^{(i)} = \sum_{j=1}^{I} \mathrm{SATT}\left(\boldsymbol{q}^{(i)}, \boldsymbol{k}^{(j)}\right) \cdot \boldsymbol{v}^{(j)}, \tag{4}$$

where the self-attention function $\mathrm{SATT}(\cdot,\cdot)$ is a softmax normalized interrelationships of β_{ij}, *i.e.*,

$$\mathrm{SATT}\left(\boldsymbol{q}^{(i)}, \boldsymbol{k}^{(j)}\right) = \frac{\exp\left(\beta_{ij}\right)}{\sum_{j'=1}^{I} \exp\left(\beta_{ij'}\right)}. \tag{5}$$

Note that the winner score $s^{(i)}$ is calculated according to the attention of all other assets. This way, CAAN accounts for the interrelationships among all assets.

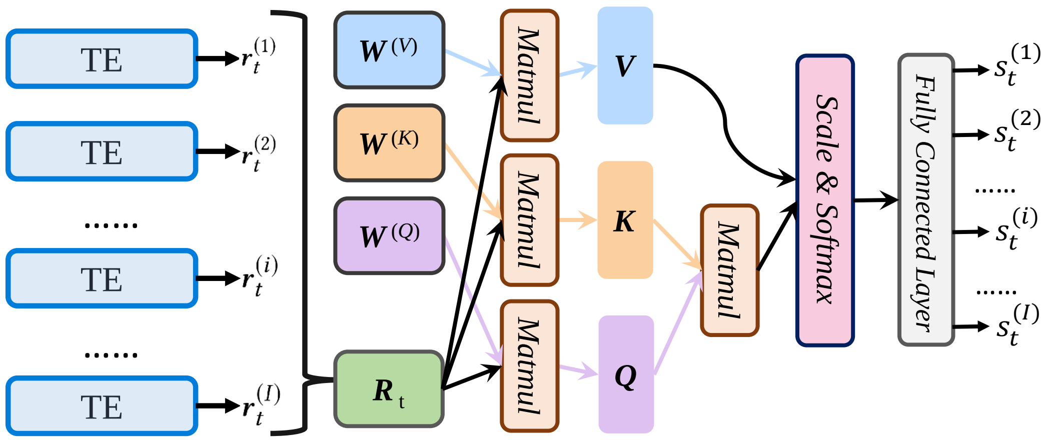

Figure 2: Architecture of Cross-asset Attention Network (CAAN).

We use a fully connected layer to transform the attention vector $\boldsymbol{a}^{(i)}$ into a winner score as $s^{(i)} = \tanh\left(\boldsymbol{w}^{(s)\top} \cdot \boldsymbol{a}^{(i)} + e^{(s)}\right)$, where $\boldsymbol{w}^{(s)}$ and $e^{(s)}$ are the connection weights and the bias to learn. The winner score $s_t^{(i)}$ indicates the likelihood of asset i being selected into long positions in the t-th holding period. So far, the model embeds little economic meaning because an asset with a higher winner score may not necessarily contribute positively to portfolio performance. It is just a flexible structure (with high-dimensional parameters) for generating portfolios, to be trained using RL later.

[18]Assume that the components of q and k are independent random variables with mean 0 and variance 1. Then their dot prodcut, $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$, has mean 0 and variance d_k.

3.3 Portfolio Generator

Given the winner scores $\left\{s^{(1)}, \cdots, s^{(i)}, \cdots, s^{(I)}\right\}$ of a total of I assets, AlphaPortfolio next constructs a hedge portfolio with long positions in assets with high winner scores and short positions in those with low winner scores. Specifically, we first sort the assets in descending order by their winner scores and obtain the sequence number $o^{(i)}$ for each asset i. Let G denote the preset size of the long and short parts of the portfolio $\boldsymbol{b}^+$ and $\boldsymbol{b}^-$. If $o^{(i)} \in [1, G]$, asset i then enters the portfolio $\boldsymbol{b}^{+(i)}$, with the investment proportion given by $b^{+(i)} = \frac{\exp\left(s^{(i)}\right)}{\sum_{o^{(i')} \in [1,G]} \exp\left(s^{(i')}\right)}$; if $o^{(i)} \in (I-G, I]$, $b^{-(i)} = \frac{\exp\left(-s^{(i)}\right)}{\sum_{o^{(i')} \in (I-G,I]} \exp\left(-s^{(i')}\right)}$ is the short proportion of asset i.

The remainder assets do not have clear buy/sell signals and are thus not included in the portfolio. For simplicity, we use one vector to record all the information of the two portfolios. That is, we form the vector $\boldsymbol{b}^c$ of length I, with $b^{c(i)} = b^{+(i)}$ if $o^{(i)} \in [1, G]$, or $b^{c(i)} = b^{-(i)}$ if $o^{(i)} \in (I-G, I]$, or 0 otherwise, $i = 1, \ldots, I$. In what follows, we use $\boldsymbol{b}^c$ and $\{\boldsymbol{b}^+, \boldsymbol{b}^-\}$ interchangeably.

Note that before we fully train the model, because the parameters for TE and CAAN are all randomly initiated, the AlphaPortfolio could perform miserably at the beginning. Before proper training, a high winner score does not mean it is a better asset to invest in. After training, constructing the portfolio based on winner scores can generate portfolios that lead to high performance metrics. We next describe the training process.

3.4 Optimization via Reinforcement Learning

We embed the AlphaPortfolio strategy into an RL game with continuous agent actions to train the model parameters, where an episode (a T-period investment) is modeled as a state-action-reward trajectory π of an RL agent, *i.e.,* $\pi = \{state_1, action_1, reward_1, \ldots, state_t, action_t, reward_t, \ldots, state_T, action_T, reward_T\}$. The $state_t$ is the history market state observed at t, which is expressed as a tensor $\mathcal{X}_t = \{\boldsymbol{X}_t^{(i)}, i = 1, \cdots, I\}$. The $action_t$ is portfolio vector $\boldsymbol{b}_t^c$ given by AP, of which the element $action_t^{(i)}$ indicates the portfolio weight the agent invests asset i at t.[19]

Let $H_\pi = SharpeRatio\left(\{reward_1, \ldots, reward_t\}\right)$ denote the Sharpe ratio of π, then $reward_t$ is the return of $action_t$. For all episodes, the average reward is $J(\theta) = \mathbb{E}\left[H_{\pi_\theta}\right]$. The objective of the RL model optimization is to find the optimal parameters $\theta^* = \arg\max_\theta J(\theta)$. We use the gradient ascent approach to iteratively optimize θ at round τ as $\theta_\tau = \theta_{\tau-1} + \eta \nabla J(\theta)|_{\theta=\theta_{\tau-1}}$, where η is a learning rate. When we empirically train the model, an episode

[19] An episode is one complete play of the agent interacting with the environment in the general RL setting.

is defined as one year of investment which contains 12 trading periods and $\nabla J(\theta)$ is automatically calculated using the deep learning framework we employ.

Note that our case involves a deterministic gradient and implicit in our RL setup is that our action does not alter the environment. In other words, we assume that our investor is a price-taker and small enough not to move prices significantly. This common assumption in many asset pricing models implies a limitation on the scale of AlphaPortfolio, but can be relaxed. Note also that when we want to directly optimize over, say, the Sharpe ratio, we cannot use supervised learning because there is no "correct" Sharpe ratio to start with.

3.5 Multi-head Attention and Implementation Details

To better understand the model and connect that to empirical studies, we should explain how AlphaPortfolio builds on existing Transformer models and details of our implementation. Readers familiar with Transformer models can safely skip this subsection.

First, it is useful to discuss the use of multi-head attention in existing Transformer models, which we inherit. Scaled dot-product attention in plain-vanilla Transformer models (show in Fig.3) constitutes a basic unit of multi-head attention. It replaces recurrence with self-attention. Unlike traditional attention methods, self-attention performs attention on a single sequence. The value of each position is calculated by all the positions in the sequence.

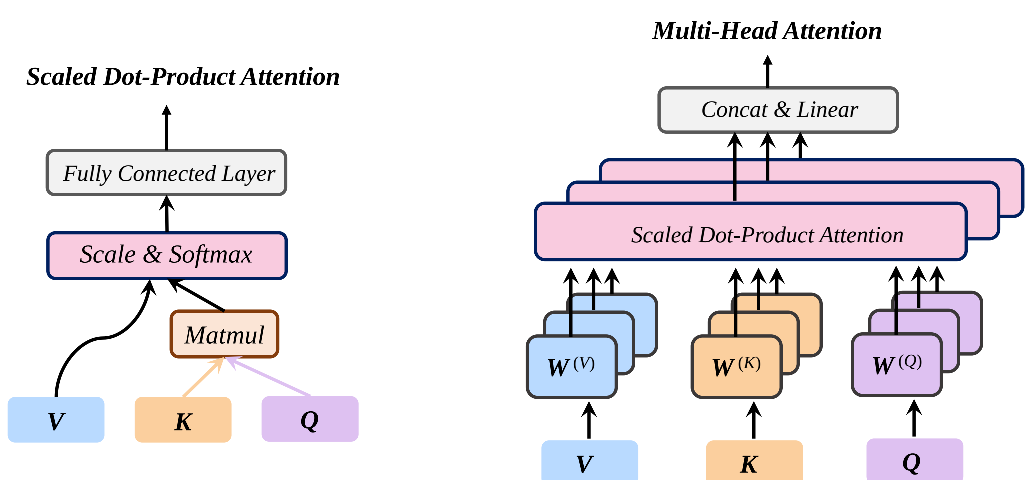

Figure 3: Scaled Dot-Product Attention (left) and Multi-Head Attention (right).

The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. In practice, query, key and value matrices can be respectively packaged into Q, K and V. So we can compute the attention function on a set of queries simultaneously. The scale factor,

$\frac{1}{\sqrt{d_k}}$, is to avoid the dot-product getting too large.[20]

$$Attention(Q, K, V) = softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{6}$$

Multi-head attention (shown in Fig.3) can be regarded as applying scaled dot-product attention in an h different feature space and finally concatenate the results.

$$MultiHead(Q, K, V) = Concat(head_1, \cdots, head_h)W^O \tag{7}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{8}$$

In addition to attention sub-layers, each encoder contains a fully connected feed-forward network which is applied to each position separately and identically. In fact, we can consider this part as convolutions with kernel size 1. It consists of two linear transformations with a ReLU activation in between.

It should be clear that our TE-CAAN retains features of the original design to the extent possible and utilizes residual connection and layer normalization. However, we isolate the encoder and differ in implementation details. Our model is based on PyTorch version 1.0.1 on four NVIDIA 1080Ti. To achieve better performance and take full advantage of computing resources, we adopt PyTorch's advanced api to automate data parallelism for the TE.

Because the amount of data in neural-network translation tasks differs significantly from that in our task, we do not follow the original parameter setting with a stack of six encoder blocks. We instead find that one TE block achieves fast convergence and already produces exceptional results. Also, we reduce the dimension of embedding from 512 to 256 and reduce the dimension of feed forward from 2048 to 1024 for computational efficiency. The number of heads in multi-head attention is set to four. Once again, our innovation lies in the dynamic programming approach for direct optimization, which drive the results, not these fine specifications of the TE model. Importantly, we add CAAN as which in itself is an innovation on top of TE.

4 Empirical Performance: A Study of U.S. Equities

4.1 Data Description

We now apply the AlphaPortfolio model to public equities in the United States. Our sample period is July 1965 to June 2016. There are about 1.8 million observations in the

[20] Assume that the components of q and k are independent random variables with mean 0 and variance 1. Then their dot prodcut, $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$, has mean 0 and variance d_k.

most baseline analysis.

Market data. Monthly stock return data are from the Center for Research in Security Prices (CRSP). We follow the literature standard to focus on common stocks of firms incorporated in the United States and trading on Amex, Nasdaq, or NYSE.

Input feature data. Firms' balance-sheet data come from the Standard and Poor's Compustat database. To mitigate survivorship bias due to backfilling, we also require that a firm appears in the dataset for at least two years for training the model. For out-of-sample test, we only require a firm to be in the dataset for one year.

Similar to Freyberger, Neuhierl, and Weber (2019), we construct firm characteristics as raw input features that fall into six categories: price-based signals such as monthly returns, investment-related characteristics such as the change in inventory over total assets, profitability-related characteristics such as return on operating assets, intangibles such as operating accruals, value-related characteristics such as the book-to-market ratio, and trading frictions such as the average daily bid-ask spread. We consider lagged features up to 12 months prior to the month of portfolio construction. Overall we have 51 times 12 input features at any time. Appendix B describes the construction of input features.[21]

Text data. We obtain text data from Company Filings at SEC Edgar (`https://www.sec.gov/edgar/`). To facilitate the rapid dissemination of financial and business information about companies, the United States Securities and Exchange Commission (the SEC) approved a rule requiring publicly-listed firms to file their securities documents with the SEC via the Electronic Data Gathering, Analysis and Retrieval (EDGAR) system. This has made regulatory filings publicly available since 1993. We illustrate the approach with Management Discussion and Analysis (MD&A) sections of both the quarterly report (10-Q) and the annual report (10-K). Other forms of text data we can utilize are Risk Factor Discussion in 10-K reports and analyst reports.

4.2 Empirical Tests and Results

The objective we specify for AlphaPortfolio is the out-of-sample Sharpe ratio, which is natural and widely used by academics and practitioners (e.g., Nagel, 2019). Alternative objectives can be accommodated easily. To train the model, we use data till the end of

[21] Tables 1 and 2 in Freyberger, Neuhierl, and Weber (2019) provide an overview of their input features and their summary statistics, while their Section A.1 describe the construction of characteristics and related references. We incorporate time-variation of firm characteristics over the past month for variables beyond past return based predictors, and therefore have more input features.

1989. Note that RL differs from supervised learning and relies on interactions with the environment. It therefore does not distinguish between training and validation sets. We use rewards to judge the quality of training and adjust hyper-parameters of AlphaPortfolio.

To start, the parameters are randomly initiated and we randomly draw a month from the training set without replacement and use the subsequent 12 months' performance (e.g., Sharpe ratio) as the reward to update the parameters. We repeat the step with the remaining months in the training set until we exhaust all the months in the training set. We call this multi-step process an epoch. In our implmentation, we use 30 epochs which is sufficient for the parameters converge.[22]

After training, we test AlphaPortfolio on the sample starting from 1990. All our results are obtained out-of-sample rather than relying on in-sample predictability adopted in traditional statistical tests. This is crucial to prevent over-fitting with low signal-to-noise financial data (e.g., Martin and Nagel, 2019). Note that the AlphaPortfolio model is fine-tuned at annual intervals in our test samples (rolling updates). In order words, after seeing one year's performance in the test sample, we use it to update the model parameters. Here we use 6 epochs each containing 12 steps. learning rate is similarly set to 1e-4, then 5e-5 after 2 epochs and 1e-5 after 4 epochs. Even though one could have fine-tuned the model at higher frequencies such as monthly, we use annual frequency to avoid overfitting to monthly variations and high computation costs for updating deep learning models at high frequency — a point also discussed in Gu, Kelly, and Xiu (2018). Updating at a lower frequency, on the other hand, tend to reduce the out-of-sample performance because of stale information.

Table 1 reports the main results. Columns (1)-(3) display the various moments of the AlphaPortfolio (AP) as well as metrics such as turnover. AP achieves an out-of-sample Sharpe ratio of 2.0 in the full test data set, and even higher when we restrict the training and testing to large and liquid stocks (in Columns (2) and (3) we require the stocks to be in the top 90 or 80 percentiles based on market cap). This means that the AP strategy is not driven by microcaps and can be implemented without worrying much about liquidity or transaction costs.[23]

Both high average returns and low volatility contribute to the high Sharpe ratio of our

[22] As is typical in the training of AI models, we gradually decrease the learning rate η as we go through more epochs. For example, we use a learning rate of 1e-4 in the first 5 epochs, then 5e-5 in the next 10 epochs, and 1e-5 after that. Such a tuning prevents the parameters from oscillating around optimum points or settling on local optima. By monitoring the flattening of a loss curve (loss as in the negative of reward), one can decide whether the parameters have converged.

[23] As Hou, Xue, and Zhang (2020) point out, 65 percent of known anomalies cannot clear the single test hurdle of $|t| \geq 1.96$ because the original studies overweight microcaps via equal-weighted returns and often with NYSE-Amex-NASDAQ breakpoints in portfolio sorts and cross-sectional regressions, especially those with ordinary least squares, are highly sensitive to microcap outliers. The authors also point out how liquidility and trading frictions also renders many anomalies in academic studies infeasible for trading.

Table 1: Out-of-sample Performance of AlphaPortfolio

In each month, AlphaPortfolio constructs a hedged portfolio of long/short stocks in highest/lowest decile of winner scores. The detailed investment strategy is described in Section 3.3. Parameters are initially obtained from the training periods, then fine-tuned once a year in the out-of-sample periods (rolling update). q_n symbolizes the n^{th} NYSE size percentile. Columns (1)-(3) display portfolio return information. Columns (4)-(9) further adjust portolio returns by the CAPM, Fama-French-Carhart 4-factor model (FFC), Fama-French-Carhart 4-factor and Pastor-Stambaugh liquidity factor model (FFCPS), Fama-French 5-factor model (FF5), Fama-French 6-factor model (FF6), Stambaugh-Yuan 4-factor model (SY), and Hou-Xue-Zhang 4-factor model (Q4). Again, (4)-(5) present the alphas for the overall sample where as (6)-(9) present alphas for subsamples excluding microcap firms in the smallest decile and quintile, respectively. "*," "**," and "***" denote significance at the 10%, 5% and 1% level, respectively.

	AP Performance				AP Excess Alpha					
	(1)	(2)	(3)		(4)	(5)	(6)	(7)	(8)	(9)
Firms	All	$> q_{10}$	$> q_{20}$	Factor Models	All		$> q_{10}$		$> q_{20}$	
					$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return (%)	17.00	17.10	18.10	CAPM	13.9***	0.005	12.2***	0.088	14.0***	0.102
Std.Dev. (%)	8.50	7.70	8.20	FFC	14.2***	0.052	13.4***	0.381	14.7***	0.465
Sharpe	2.00	2.31	2.21	FFC+PS	13.7***	0.054	12.3***	0.392	13.3***	0.480
Skewness	1.42	1.74	1.91	FF5	15.3***	0.12	13.8***	0.426	14.7***	0.435
Kurtosis	6.33	5.70	5.97	FF6	15.6***	0.128	14.5***	0.459	15.8***	0.516
Turnover	0.26	0.24	0.26	SY	17.4***	0.037	15.8***	0.332	17.0***	0.394
MDD	0.08	0.02	0.02	Q4	16.0***	0.121	15.0***	0.495	16.2***	0.521

algorithm. Moreover, AP uses a much lower frequency of re-balancing (monthly), turnover, and maximum drawdown relative to other (high-frequency) machine learning strategies or traditional, anomaly-based trading.

As is standard in the literature, we also control for benchmark factor models in Columns (4)-(9), which include the CAPM, the Fama-French-Carhart 4-factor model (FFC, Carhart 1997), the Fama-French-Carhart 4-factor model plus the Pastor-Stambaugh liquidity factor model (FFC+PS, Pástor and Stambaugh 2003), the Fama-French 5-factor model (FF5, Fama and French 2015), the Fama-French 6-factor model (FF6, Fama and French 2018), the Stambaugh-Yuan 4-factor model (SY, Stambaugh and Yuan 2017), and the Hou-Xue-Zhang 4-factor model (Q4, Hou, Xue, and Zhang 2015). AP has significant and large annualized α even after controlling for various factors.

Note that AP does not pick small and illiquid stocks as many other models do based on back-testing — a somewhat surprising result. We attribute this to the fact that even though small and illiquid stocks tend to commend high expected returns, they also significantly contribute to the volatility of a portfolio. The direct optimization of the Sharpe ratio rather than characteristic sorting effectively avoids small and illiquid stocks in the construction.

Table 2 further demonstrates the efficacy of RL and AI for investment. Panel A compares AP with the "non-parametric" (NP) model and portfolio strategy in Freyberger, Neuhierl, and Weber (2019). We pick NP as a benchmark because in addition to the fact that we use similar firm characteristics as inputs, NP is likely the best-performing approach among all existing recent machine learning applications in asset pricing. AP outperforms most other machine-learning-based strategies in the literature, but NP achieves a higher Sharpe ratio on its test sample in 1991-2014. Once we exclude illiquid and small stocks, AP outperforms NP significantly, consistent with Avramov, Cheng, and Metzker (2019)'s findings that recent machine learning strategies often derive their performances from microcap and illiquid stocks. The superior performance here does not invalidate other models such as NP, as their focus is on minimizing pricing errors rather than optimizing portfolio performance.

Table 2: Comparison with Alternative Models Using Out-of-sample Performance

Panel A compares AP's performance with one benchmark model proposed in Freyberger, Neuhierl, and Weber (2019) in 1991-2014 (their test sample period). *NP* and *AP* denote the nonparametric model in Freyberger, Neuhierl, and Weber (2019) and the AlphaPortfolio, respectively. Panel B presents the results using the full test-sample period (1990-2016) when we follow the traditional two-step approach to first use transformer encoder to predict stock returns and then form expected-return-sorted portfolios. Again, q_n symbolizes the n^{th} NYSE size percentile.

Firms	All		$> q_{10}$		$> q_{20}$	
	(1)	(2)	(3)	(4)	(5)	(6)
Panel A: Comparison with NP Model						
Model	NP	AP	NP	AP	NP	AP
Return	45.84	15.60	21.12	17.70	15.48	17.90
Std. Dev.	16.66	8.20	13.27	7.60	14.90	8.60
Sharpe	2.75	1.90	1.60	2.33	1.04	2.08
Skewness	3.53	1.20	0.30	1.77	-0.50	1.88
Kurtosis	19.56	6.54	7.80	5.57	13.06	5.46
Turnover	0.69	0.26	0.74	0.24	0.74	0.26
MDD	0.10	0.08	0.27	0.02	0.36	0.08
Panel B: TE-based Return-sorted Portfolio						
Weight	Equal	Value	Equal	Value	Equal	Value
Return	8.80	3.20	19.30	5.90	18.10	6.30
Std.Dev.	9.40	8.80	11.70	8.60	9.80	7.90
Sharpe	0.94	0.36	1.65	0.69	1.85	0.80
Skewness	2.46	-1.78	6.02	1.48	4.07	1.00
Kurtosis	19.84	27.55	67.00	10.89	33.21	15.63
MDD	0.08	0.15	0.3	0.10	0.02	0.07

It is worth mentioning that the winner score is not just another estimator of expected returns. It is an abstract score that helps maximize investor objectives. In that sense, RL takes into consideration both the expected returns and variance-covariance of available assets. Moreover, it allows incorporation of position limits and other constraints widely observed in applications. To see how RL adds to AP's performance, Panel B in Table 2 presents the out-of-sample performance when we follow the traditional two-step approach to first use TE to predict stock returns and then form expected-return-sorted portfolios.[24] We note that out-of-sample Sharpe ratio can reach 0.8 with market-cap-adjusted weights and close to 2 with equal weights. On the one hand, this demonstrates the TE under the traditional two-step portfolio construction still outperforms many other strategies (machine-learning-based or anomaly/sorting-based). On the other hand, the performance is dwarfed by the RL-based AP model, highlighting the utility of RL. For example, the value-weighted portfolio on the full sample, which is more feasible than equal weighting in practice, has an out-of-sample Sharpe ratio of 0.36 as compared to AP's Sharpe ratio of 2. In other words, had we used winner scores as an estimator of expected returns, the portfolio performance would be significantly inferior to AP's, no matter if one uses equal weights or value weights.

4.3 Economic Restrictions and Model Robustness

Performance of traditional anomalies and machine learning strategies is often suspected to be primarily driven by microcap stocks, illiquid stocks, extreme market conditions, equal-weighting of the portfolio, etc. We now dispel such concerns and demonstrate AP's robustness. We note that the results reported below after imposing various economic restrictions are only lower bounds on AP's performance as AP has not been re-trained on the subsamples excluding certain stocks or with specific restrictions. Instead, in the tests we simply set the portfolio weights of the excluded stocks or non-admissible stocks under the specific restrictions to zero.

Microcaps. Given that microcaps have the highest equal-weighted returns and the largest cross-sectional dispersions in returns and in anomaly variables, anomalies in cross-sectional asset returns could be driven by microcap stocks and costly-to-trade stocks (Novy-Marx and Velikov, 2016; Hou, Xue, and Zhang, 2020). Avramov, Cheng, and Metzker (2019) find that machine techniques face similar issues in that return predictability and the anomalous patterns are concentrated in difficult-to-arbitrage stocks and in times of high limits to arbitrage.

[24] Because the links among various assets are typically not explicitly modeled in simple characteristic-based sorting, Table 2 does not involve CAAN when modeling asset returns. The results with CAAN are similar.

Table 1 reveals that AP's performance is not driven by the bottom 10% and 20% of stocks based on market capitalization. If anything, the performance improves after excluding them, which outperforms almost all other machine learning models in terms of Sharpe ratio, maximum drawdown, and turnover. This observation implies that AP is more effective in uncovering patterns in large and liquid stocks than in a mixture of large and small cap stocks.

Turnovers, shorting, and transaction costs. Note that following Koijen, Moskowitz, Pedersen, and Vrugt (2018); Freyberger, Neuhierl, and Weber (2019), we calculate turnover using $Turnover_t = \frac{1}{4}\sum_i \left|w_{t-1}^i(1+r_t^i) - w_t^i\right|$, where the coefficient $\frac{1}{4}$ is to avoid double-counting (a factor of 2) and to adjust for that the long/short strategies have \$2 exposure (another factor of 2).

Gu, Kelly, and Xiu (2018) uses an alternative definition of turnover, $Turnover_{GKX} = \left|w_{i,t+1} - \frac{w_{i,t}(1+r_{i,t+1})}{1+\sum_j w_{j,t}r_{j,t+1}}\right|$.[25] As Gu, Kelly, and Xiu (2018) point out, many strategies have turnovers well above 100% monthly. AP still has very low turnovers relative to other traditional anomalies or machine learning strategies, whether we calculate long/short-leg turnover respectively and sum it up, or we directly calculate turnover of hedge portfolio using this alternative measure.

Because of the low turnovers, AP's performance is robust to incorporating transaction costs. For example, Setting a transaction cost at 0.1%, AP still yields an out-of-sample Sharpe Ratios of 2.01, 2.27, and 2.16 for the full sample and the $size > q_{10}$ and $size > q_{20}$ subsamples respectively. The turnover and maximum drawdown do not differ much from those from the baseline model.

While many existing strategies (whether anomaly-based or machine-learning-based) for long-short portfolio construction heavily depend on the short positions, AP's long and short positions both contribute significantly to the performance. If anything, the long positions play a more dominant role. For example, they generate returns of 37.7%, 40.3%, and 41.1% for the full sample, size$>q10$ sample, and size$>q20$ sample respectively, as compared to the 4.8%, 6.1%, and 5% from short positions.

Performance attenuation over time. Many anomalies have attenuated since the early 2000s (e.g., Chordia, Subrahmanyam, and Tong, 2014; McLean and Pontiff, 2016; Linnainmaa and Roberts, 2018; Han, He, Rapach, and Zhou, 2018) because the U.S. equity market witnessed several structural changes since the 2000s, such as the introduction of decimalization. Could AP be trading on known anomalies that have been traded away in recent years or on seeming anomalies from data snooping? To answer this, we restrict the test to a

[25]Note that to have the proper normalization for long-short portfolios in our setting, we have added 1 in the denominator.

subsample of more recent years (2001-2016) As we report in Table 3 Columns (4)-(6), AP's performance remain robust.

Table 3: Out-of-Sample Performance in Recent Years

This table reports alphas for portfolios of long/short stocks in the highest/lowest decile of winner-scores from 2001 to 2016. Portfolio returns are further adjusted by the CAPM, Fama-French-Carhart 4-factor model (FFC), Fama-French-Carhart 4-factor and Pastor-Stambaugh liquidity factor model (FFCPS), Fama-French 5-factor model (FF5), Fama-French 6-factor model (FF6), Stambaugh-Yuan 4-factor model (SY), and Hou-Xue-Zhang 4-factor model (Q4). The first two columns present the alphas for the overall sample. The remaining four columns present alphas for subsamples excluding microcap firms in the smallest decile and quintile, respectively. q_n symbolizes the n^{th} NYSE size percentile. "*," "**," and "***" denote significance at the 10%, 5% and 1% level, respectively.

	AP Performance				AP Excess Alpha					
	(1)	(2)	(3)		(4)	(5)	(6)	(7)	(8)	(9)
Firms	All	$> q_{10}$	$> q_{20}$	Factor Models	All		$> q_{10}$		$> q_{20}$	
					α(%)	R^2	α(%)	R^2	α(%)	R^2
Return(%)	18.10	16.10	16.60	CAPM	16.5***	0.007	13.6***	0.136	13.8***	0.176
Std.Dev.(%)	9.20	7.90	8.90	FFC	16.3***	0.078	12.9***	0.497	13.3***	0.594
Sharpe	1.97	2.04	1.87	FFC+PS	15.7***	0.080	11.7***	0.506	11.7***	0.606
Skewness	1.67	1.53	1.61	FF5	18.0***	0.151	13.8***	0.426	14.6***	0.432
Kurtosis	5.95	4.23	3.59	FF6	17.8***	0.174	13.3***	0.560	14.0***	0.620
Turnover	0.25	0.23	0.25	SY	18.9***	0.065	15.3***	0.428	16.5***	0.502
MDD	0.05	0.03	0.04	Q4	16.9***	0.121	13.7***	0.532	14.6***	0.551

We also plot AP's Sharpe ratio and excess α relative to seven benchmark factor models over time. We look at non-overlapping 3-year windows and the trends are depicted in Figure 4. Overall, the Sharpe ratio is particularly high at the start of the test sample, but does not exhibit particular trends post 2000s. Excess alphas fluctuate but show no sign of attenuation.

Industry attribution and weights. We perform industry/style attribution tests and report the results in Table 4, After regressing AP's monthly out-of-sample returns (1990-2016) on the 12 industries according to Fama and French, the intercept is both economically and statistically significant.[26] We also do not see significant loadings on most industry portfolios, except for durables (positive), energy (negative), and retails (negative). Overall, AP picks up patterns beyond those associated with industries or styles and does not heavily weigh particular industries.

It is also worth noting that our findings are not driven by the issue of equal weighing versus value weighing. The Pearson coefficient of assets' investment proportion and its market capitalization shows that under 15% of the time, the two factors are significantly

[26] The definition and data are at https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html.

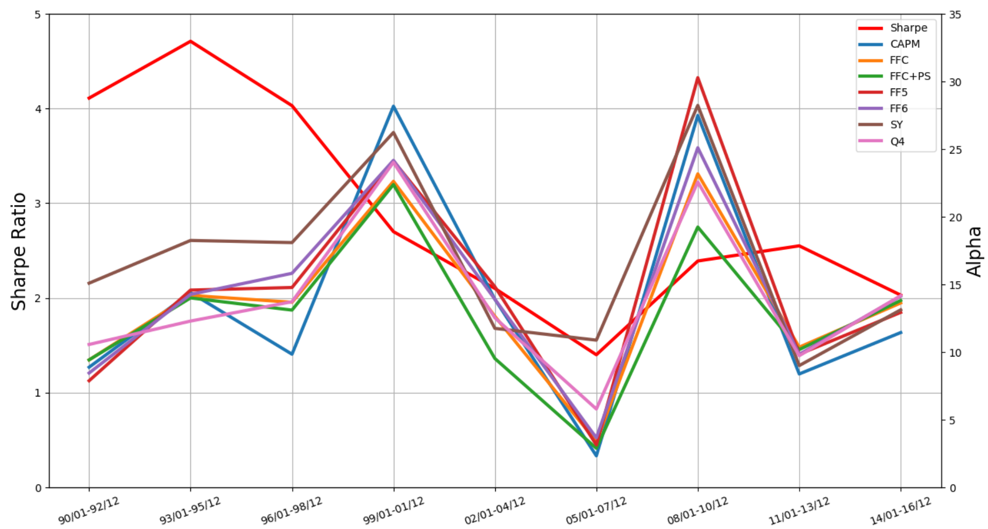

Figure 4: Trends of AlphaPortfolio Performance

Table 4: Industry Attribution Test for AlphaPortfolio

This table presents the results from regressing the AlphaPortfolio return on various industry portfolio returns. The 12 industries are Fama-French industries based on four-digit SIC codes: 1 NoDur (Consumer Nondurables) – Food, Tobacco, Textiles, Apparel, Leather, Toys; 2 Durbl (Consumer Durables) – Cars, TVs, Furniture, Household Appliances; 3 Manuf (Manufacturing) – Machinery, Trucks, Planes, Off Furn, Paper, Com Printing; 4 Enrgy –Oil, Gas, and Coal Extraction and Products; 5 Chems – Chemicals and Allied Products; 6 BusEq (Business Equipment) – Computers, Software, and Electronic Equipment; 7 Telcm – Telephone and Television Transmission; 8 Utils – Utilities; 9 Shops – Wholesale, Retail, and Some Services, Laundries, Repair Shops; 10 Hlth – Healthcare, Medical Equipment, and Drugs; 11 Money – Finance; 12 Other Other – Mines, Constr, BldMt, Trans, Hotels, Bus Serv, Entertainment. q_n symbolizes the n^{th} NYSE size percentile. "*," "**," and "***" denote significance at the 10%, 5% and 1% level, respectively.

		Industry												
	Intercept	NoDur	Durbl	Manuf	Enrgy	Chems	BusEq	Telcm	Utils	Shops	Hlth	Money	Other	R^2
All														
Coefficient	1.438***	0.096	0.035	0.169**	-0.133***	-0.059	0.055	0.019	-0.026	-0.125**	-0.074*	-0.033	-0.004	0.119
Std.Err.	0.141	0.071	0.039	0.084	0.037	0.067	0.034	0.043	0.045	0.061	0.043	0.048	0.078	
$> \mathbf{q_{10}}$														
Coefficient	1.494***	0.058	0.073**	0.027	-0.060*	-0.107*	0.047	0.020	-0.082**	-0.119**	-0.012	0.043	0.048	0.135
Std.Err.	0.125	0.064	0.035	0.075	0.033	0.060	0.030	0.039	0.040	0.054	0.039	0.043	0.070	
$> \mathbf{q_{20}}$														
Coefficient	1.584***	0.078	0.076**	0.041	-0.082**	-0.134**	0.080**	0.006	-0.078*	-0.167***	-0.034	0.049	0.104	0.190
Std.Err.	0.133	0.068	0.037	0.080	0.035	0.063	0.032	0.041	0.042	0.057	0.041	0.046	0.074	

negatively correlated. In other words, our portfolio is close to value-weighted, which is more feasible to create in practice than equal-weighted portfolios that many extant anomalies are derived from.

Unrated or downgraded firms. To rule out the possibility that our results are driven by unrated or downgraded firms that are hard to trade on, we next follow Avramov, Cheng, and Metzker (2019) to test our model on a subsample including only rated firms, i.e., firms with data on S&P's long-term issuer credit rating. Such rated firms tend to be large and liquid, with better disclosure, more analyst coverage, and smaller idiosyncratic volatility, which makes trading them cheaper and more feasible. Within the rated firms, we further exclude firms with credit rating downgrades which are typically associated with greater trading and arbitrage frictions (e.g., Avramov, Chordia, Jostova, and Philipov, 2013). As in Avramov, Cheng, and Metzker (2019), we exclude stock-month observations from 12 months before to 12 months after the downgrade events. The results are reported in Table 5. As seen in Panel A, even though the α values are smaller, they are still significant and greater than most known anomalies. The Sharpe ratios shown in Panel B exhibit similar patterns.

The significant reductions in excess alpha and Sharpe ratio are natural and are mostly an artifact that AP is not re-estimated after imposing the economic restrictions or re-trained on samples excluding unrated and downgraded firms. We simply set the portfolio weights for excluded stocks to zero when performing out-of-sample tests. So the excess returns reported here are all lower bounds on AP's performance under the various economic restrictions.

Market conditions. Prior studies document that traditional anomalies are more salient during high investor sentiment, high market volatility, and low market liquidity (e.g., Stambaugh, Yu, and Yuan, 2012; Nagel, 2012). Many machine learning strategies are also shown to have insignificant α in times of low VIX or low sentiment (Avramov, Cheng, and Metzker, 2019). If the out-of-sample tests of AP only perform well during high investor sentiment and high market volatility, AP may not be implementable in practice due to limits to arbitrage.

To investigate this issue, we examine the AP performance in sub-periods of different states of investor sentiment, market volatility (implied and realized), and liquidity. Investor sentiment ($SENT$) is defined as the monthly Baker and Wurgler (2007) investor sentiment; market volatility (VIX) is defined as the monthly VIX index of implied volatitity of S&P 500 index options; realized market volatility ($MKTVOL$) is defined as the standard deviation of daily CRSP value-weighted index return in a month; and market illiquidity ($MKTILLIQ$) is defined as the value-weighted average of stock-level Amihud illiquidity for all NYSE/AMEX

Table 5: Out-of-Sample Performance of Portfolios Excluding Unrated (and Downgraded) Stocks

This table reports alphas for portfolios of long/short stocks in the highest/lowest decile of winner-scores from 1990 to 2016, where (1)-(3) exclude *unrated* stocks from the portfolio and (4)-(6) exclude both *unrated* and recently *downgraded* stocks. In Panel A, portfolio returns are adjusted by the CAPM, Fama-French-Carhart 4-factor model (FFC), Fama-French-Carhart 4-factor and Pastor-Stambaugh liquidity factor model (FFC+PS), Fama-French 5-factor model (FF5), Fama-French 6-factor model (FF6), Stambaugh-Yuan 4-factor model (SY), and Hou-Xue-Zhang 4-factor model (Q4). Within (1)-(3) and within (4)-(6), the first two columns present the alphas for the whole sample. The remaining four columns present alphas for subsamples excluding microcap firms in the smallest decile and quintile, respectively. Panel B presents other performance metrics in a similar fashion. q_n symbolizes the n^{th} NYSE size percentile. "*," "**," and "***" denote significance at the 10%, 5% and 1% level, respectively.

Panel A: Out-of-Sample Alpha under Various Factor Models

	Excluding Unrated Firms						Excluding Unrated & Downgraded					
	(1)		(2)		(3)		(4)		(5)		(6)	
Firms	All		Size > q_{10}		Size > q_{20}		All		Size > q_{10}		Size > q_{20}	
	α(%)	R^2	α(%)	R^2	α(%)	R^2	α(%)	R^2	α(%)	R^2	α(%)	R^2
CAPM	3.4***	0.000	4.6**	0.051	5.3***	0.067	4.7***	0.000	3.8***	0.070	4.3***	0.076
FFC	3.1***	0.061	5.2***	0.380	5.5***	0.504	4.4***	0.047	4.2***	0.305	4.4***	0.437
FFC+PS	2.4*	0.070	4.6***	0.384	4.7***	0.511	3.4***	0.060	3.6***	0.309	3.4***	0.448
FF5	3.6**	0.146	4.3**	0.251	4.0***	0.375	5.0***	0.100	3.7***	0.219	2.9***	0.338
FF6	3.7***	0.147	5.6***	0.391	5.3***	0.517	5.0***	0.100	4.7***	0.308	4.0***	0.446
SY	5.3***	0.182	7.9***	0.393	7.4***	0.494	6.6***	0.149	6.9***	0.333	6.2***	0.444
Q4	3.5***	0.109	5.8***	0.294	5.6***	0.394	5.0***	0.082	5.1***	0.257	4.5***	0.350

Panel B: Other Portfolio Performance Metrics

	Excluding Unrated Firms			Excluding Unrated & Downgraded		
	(1)	(2)	(3)	(4)	(5)	(6)
Firms	All	Size > q_{10}	Size > q_{20}	All	Size > q_{10}	Size > q_{20}
Return(%)	6.18	8.30	8.99	7.45	7.54	8.06
Std.Dev.(%)	5.93	7.89	7.03	6.42	7.02	7.00
Sharpe	1.04	1.05	1.28	1.16	1.07	1.15
Skewness	-0.24	2.23	1.69	-0.89	1.18	1.18
Kurtosis	4.52	11.83	10.16	7.96	7.92	8.68
MDD	0.06	0.08	0.08	0.05	0.08	0.08

Table 6: Out-of-sample Performance of AP under Various Market Conditions

This table reports alphas (adjusted by Fama-French 6-factor model) and Sharpe ratios of the AlphaPortfolio in high and low sentiment periods (Panel A), low and high VIX periods (Panel B), low and high realized volatility periods (Panel C), and low and high illiquidity periods (Panel D). Sentiment ($SENT$) is measured from the raw version of monthly Baker and Wurgler (2007) sentiment index that excludes the NYSE turnover variable; market volatility (VIX) is defined as the monthly VIX index of implied volatitity of S&P 500 index options; realized market volatility ($MKTVOL$) is defined as the standard deviation of daily CRSP value-weighted index return in a month; market illiquidity ($MKTILLIQ$) is defined as the value-weighted average of stock-level Amihud illiquidity for all NYSE/AMEX stocks in a month. The panels report the results in the full sample as well as subsamples that exclude the bottom 10% and 20% microcaps. q_n symbolizes the n^{th} NYSE size percentile. Newey-West adjusted t-stats are shown in brackets with "*," "**," and "***" denoting 10%, 5%, and 1% significance levels, respectively.

	(1)	(2)	(3)	(4)	(5)	(6)
Sample	All		Size > q_{10}		Size > q_{20}	
Variable	Low	High	Low	High	Low	High
Panel A: AP Performance under Various SENT Periods						
FF-6 α	19.273***	13.351***	14.132***	13.746***	14.756***	16.072***
t-stat	(5.975)	(7.914)	(8.284)	(9.441)	(8.980)	(9.751)
Sharpe	1.734	1.69	2.106	1.796	2.123	1.677
Panel B: AP Performance under Various VIX Periods						
FF-6 α	10.248***	19.776***	10.812***	18.660***	10.272***	21.084***
t-stat	(5.060)	(7.421)	(8.862)	(10.201)	(8.598)	(11.107)
Sharpe	1.719	1.713	2.371	1.951	2.331	1.932
Panel C: AP Performance under Various MKTVOL Periods						
FF-6 α	10.385***	17.167***	10.687***	17.750***	11.038***	19.626***
t-stat	(30636)	(7.577)	(6.686)	(11.088)	(6.765)	(11.896)
Sharpe	1.654	1.668	2.429	1.855	2.461	1.806
Panel D: AP Performance under Various MKTILLIQ Periods						
FF-6 α	11.9635***	19.385***	14.107***	13.048***	16.096***	12.541***
t-stat	(5.616)	(6.971)	(9.084)	(9.029)	(10.209)	(8.038)
Sharpe	1.447	1.874	1.971	1.885	1.880	1.877

stocks in a month.[27] We divide the full sample into two sub-periods using the median breakpoints of $SENT$, VIX, $MKTVOL$, and $MKTILLIQ$ over the whole sample period. We report the analyses in Tables 6. For brevity, we only present baseline samples and

[27]Investor sentiment index is available at Jeffrey Wurgler's website http://people.stern.nyu.edu/jwurgler/ while monthly VIX index is available at CBOE website http://www.cboe.com/products/vix-index-volatility/vix-options-and-futures/vix-index/vix-historical-data. We thank Si Cheng for sharing the market illiquidity data used in Avramov, Cheng, and Metzker (2019).

FF6-adjusted α of the portfolios.

Overall, unlike most other machine learning strategies that yield no significant profits during low volatility/sentiment or high liquidity or when unrated or downgraded firms are excluded (Avramov, Cheng, and Metzker, 2019), AP continues exhibiting a high Sharpe ratio and significant α even after controlling for various factors, regardless of the market conditions. It does not seem that AP is more profitable during high investor sentiment or market volatility or liquidity, and the alpha is both economically and statistically significant under various market conditions. If anything, the Sharpe ratio seems better during low sentiment and low market volatility. Our deep RL signals deliver meaningful risk-adjusted performance over the entire test sample period as well as in various market states.

5 Economic Distillation of AlphaPortfolio

For model interpretation, we primarily use a regression-based model to "represent" or mimic the AP model. We first express the function of history features of a stock to its score in the TE-CAAN system, we then examine the marginal contribution of each feature and inspect its comparative statics when other features change. That allows us to identify the variables (or their higher order terms or interaction terms) that matter the most in the model. Next we use these variables and their higher-order and interaction terms as input variables to estimate a Lasso regression model.

To complement the analysis, we also regress AP's winner scores onto each firms' corresponding textual loadings on various topics discussed in the firms' filings. The natural language helps us enhance our understanding of how AlphaPortfolio behaves.

5.1 Polynomial Sensitivity Analysis for Model Transparency

In this part, we describe the polynomial feature sensitivity analysis. The main idea is to project AP onto a space of linear models using Algorithm 1.

Specifically, we first adopt the gradient-based characteristic importance method to determine which characteristics AP mostly depends on. We use $s = \mathcal{F}(\boldsymbol{X})$ to denote a combined network of TE and CAAN which maps asset's history states $\boldsymbol{X}$ to its winner score s. x_q is used to denote an element of $\boldsymbol{X}$ which is the value of feature q. Given the history state $\boldsymbol{X}$ of an asset, the sensitivity of s to x_q can be calculated as

$$\delta_{x_q}(\boldsymbol{X}) = \lim_{\Delta x_q \to 0} \frac{\mathcal{F}(\boldsymbol{X}) - \mathcal{F}\left(x_q + \Delta x_q, \boldsymbol{X}_{\neg x_q}\right)}{x_q - (x_q + \Delta x_q)} = \frac{\partial \mathcal{F}(\boldsymbol{X})}{\partial x_q}, \tag{9}$$

where $\boldsymbol{X}_{\neg x_q}$ denotes the element $\boldsymbol{X}$ except x_q.

In our implementation of AP using PyTorch, the gradient follows from the auto-grad module in the deep learning package. For all possible stock states in a market, the average influence of the stock state feature x_q to the winner score s is:

$$\mathbb{E}\left[\delta_{x_q}\right] = \int_{D_{\boldsymbol{X}}} \Pr(\boldsymbol{X})\delta_{x_q}(\boldsymbol{X})\ \mathrm{d}_\sigma, \tag{10}$$

where $\Pr(\boldsymbol{X})$ is the probability density function of $\boldsymbol{X}$, and $\int_{D_{\boldsymbol{X}}} \cdot\ \mathrm{d}_\sigma$ is an integral over all possible value of $\boldsymbol{X}$. According to the Law of Large Numbers, given a dataset that contains history states of I stocks in N holding periods, the $E\left[\delta_{x_q}\right]$ is approximated as

$$\mathbb{E}\left[\delta_{x_q}\right] = \frac{1}{I \times N} \sum_{n=1}^{N} \sum_{i=1}^{I} \delta_{x_q}\left(\boldsymbol{X}_n^{(i)} \middle| \mathcal{X}_n^{(\neg i)}\right), \tag{11}$$

where $\boldsymbol{X}_n^{(i)}$ is the history state of the i-th stock at the n-th holding period, and $\mathcal{X}_n^{(\neg i)}$ denotes the history states of other stocks that are concurrent with the history state of i-th stock.

Algorithm 1: pseudo code for economic distillation approach one

Input: Model parameters $\boldsymbol{\theta}$ of AlphaPortfolio trained on $\mathcal{D}_{train}$; Test data set $\mathcal{D}_{test} = \{\mathcal{X}_1, \ldots, \mathcal{X}_N\}$ which consists of N trading period; Hyper parameters $\{K, \alpha, p\}$;

Output: Evaluation metrics of test period;

for $n = 1$ to N **do**
- for each $\boldsymbol{X}^{(i)}$ in $\mathcal{X}_n$, generate winner score $\boldsymbol{s}_n = \left\{s^{(1)}, \ldots, s^{(I)}\right\}$ from AP;
- Calculate gradients of $s^{(i)}$ to each input raw characteristic q as $\delta_{x_q}(\boldsymbol{X})$;
- Select characteristics with top $K\%$ of $\mathbb{E}\left[\left|\delta_{x_q}\right|\right]$;
- Generate p-degree polynomial terms with selected characteristics $\boldsymbol{q}_n^{selected}$;
- Select important terms with Lasso regression(Penalty factor $= \alpha$);
- Regress $\boldsymbol{s}_n$ to selected terms, obtain corresponding coefficient;
- Re-generate winner score $\boldsymbol{s}_n'$ using selected terms and coefficients;
- Using $\boldsymbol{s}_n'$ to calculate rate of return r_n of this trading period;

end for

Calculate evaluation metrics given $R = \{r_1, \ldots, r_N\}$;

We use $\mathbb{E}\left[\left|\delta_{x_q}\right|\right]$ to measure the overall influence of an asset feature x_q on the winner score. We then generate polynomial terms with the most important features. For each month in out-of-sample periods, we can distill the AP model by regressing winner scores to selected terms using LASSO. Results in Table 7 show that even the distilled linear model

Table 7: Out-of-Sample Performance of Distillation for Algorithm One

This table presents the results of distillation for Algorithm 1. In each month of out-of-sample periods, the AlphaPortfolio is distilled into a linear model, and winner scores on the test sample are regenerated. Portfolios are formed according to the distilled winner scores following the same strategy as in Table 1. q_n symbolizes the n^{th} NYSE size percentile.

Algo1_Poly2 Performance				Algo1_Poly2 Excess Alpha						
	(1)	(2)	(3)		(4)	(5)	(6)	(7)	(8)	(9)
Firms	All	$> q_{10}$	$> q_{20}$	Factor Models	All		$> q_{10}$		$> q_{20}$	
					α(%)	R^2	α(%)	R^2	α(%)	R^2
Return(%)	16.80	22.20	19.20	CAPM	14.1***	0.000	17.9***	0.054	15.0***	0.095
Std.Dev.(%)	16.40	12.40	9.40	FFC	15.8***	0.024	18.2***	0.196	15.9***	0.411
Sharpe	1.02	1.79	2.04	FFC+PS	15.4***	0.024	17.5***	0.197	14.2***	0.428
Skewness	1.89	6.21	2.21	FF5	16.8***	0.044	19.5***	0.252	17.0***	0.447
Kurtosis	7.35	65.62	8.87	FF6	17.8***	0.062	20.3***	0.272	18.0***	0.503
Turnover	0.60	0.40	0.40	SY	21.7***	0.053	22.0***	0.188	19.1***	0.361
MDD	0.08	0.03	0.03	Q4	17.2***	0.042	20.6***	0.263	18.2***	0.505

achieves significant performance in out-of-sample tests. Here $poly = 1$ is essentially a linear regression. One can include higher-degree polynomial terms in the distillation exercises, and we stop at degree-2 for parsimony.

We then carry out both panel or Fama-Macbeth-type regressions to combine each monthly distilled linear model. We can do this for polynomials of different degrees. For parsimony, we report in Table 8 the top 50 dominant features as functions of firm characteristics and market signals, together with the corresponding T values. The sign and magnitudes of the T values allow a glimpse into how each selected feature affects the portfolio construction.

To investigate the time-varying effects of dominant characteristics, we plot heatmaps in Figure 5 for the top 15 terms in degree-2 polynomial functions to highlight their rankings in the out-of-sample periods. We also outline basic statistics for both degree-2 and degree-1 polynomials in Table 9. For distillation with degree-2 polynomials, across Ranks 1, 2, and 3, the top contributing variables are ivc (82.4%), ipmˆ2 (50.3%), Q (36.7%), ivcˆ2 (22.2%), delta_so (21.6%), and C (10.5%).[28] For degree-1 polynomials, across Rank 1, 2, and 3, at the top again are ivc (97.8%), Idol_vol (43.2%), and ipm (26.9%). The other five important characteristics (free_cf, Ret_max, ret, delta_so and C) account for 13.9% to 19.1% each.

From Figure 5 and Tables 8 and 9, we find that a small set of stock features determine the performance of our algorithm. For example, the inventory change (ivc) plays a key role in our algorithm with a probability of more than 80% included in the top contributing factors

[28]Percentages in brackets denote the fraction to be top contributing variables across all months.

Table 8: T-statistics of Selected Features in the First Distillation Algorithm: Fama-Macbetch Regressions with Polynomial Degree Two

This table presents the results where Fama Macbetch regressions are employed to interpret Algorithm 1. Polynomial degree is set to two for all terms in the regression. The suffix "_n" denotes the sequence number of features starting from 12 months ago, *i.e.,* pe_7 denotes P/E ratio at the time of five $(12 - 7 = 5)$ months lag. For details of each characteristic, please refer to Appendix B. q_n symbolizes the n^{th} NYSE size percentile. This table presents features t-statistics in the top half in magnitude.

All		size > q_{10}		size > q_{20}	
pe_7^2	19.05	Q_9	-31.01	Q_9	-23.62
s2p_11^2	17.8	Q_9^2	17.03	Idol_vol_1	14.29
Std_turnover_7^2	-9.4	investment_0	-11.74	Idol_vol_0	13.42
Std_turnover_2^2	9.24	Std_volume_1	-10.6	Q_9^2	12.56
Std_volume_11^2	8.93	Idol_vol_0	10.12	Std_volume_1	-12.29
Std_turnover_0^2	8.02	free_cf_6	-9.58	free_cf_6	-10.13
ldp_0^2	7.94	Idol_vol_4^2	8.44	investment_7	-9.96
Std_volume_5^2	7.89	Idol_vol_1	8.4	investment_0	-8.71
roa_0	7.79	Idol_vol_1^2	8.01	delta_so_1	-8.6
pe_1^2	7.78	Std_volume_1^2	7.62	Idol_vol_4	8.16
Std_turnover_4^2	7.71	ret_11^2	7.47	lev_7	-7.91
roa_0^2	7.63	free_cf_9	-7.34	ret_10^2	7.88
Std_volume_0^2	7.41	ret_5^2	7.31	Std_volume_10	-7.6
Std_turnover_6^2	7.4	delta_so_1	-7.29	Idol_vol_2	7.57
Std_turnover_1^2	7.3	Idol_vol_0^2	7.24	Std_volume_1^2	7.52
Std_turnover_3^2	-7.25	ret_10^2	7.01	delta_so_11	-7.41
Std_turnover_5^2	7.16	Ret_max_7	6.96	Idol_vol_4^2	7.35
pe_7	6.29	delta_so_11	-6.92	ret_9^2	7.27
Beta_daily_3^2	6.16	ldp_6	-6.82	nop_8	-7.13
Std_volume_4^2	6.09	Ret_max_10^2	6.81	Idol_vol_1^2	7.08
Std_turnover_9^2	5.86	s2p_4^2	6.58	beme_9	6.99
roa_11^2	5.71	ret_10	-6.58	pe_7	6.86
Beta_daily_8^2	5.63	s2p_0^2	6.52	ret_2^2	6.85
roa_11	5.48	ret_5	-6.49	ret_10	-6.84
o2p_11^2	-5.34	Ret_max_3^2	6.45	delta_so_8^2	6.79
roe_11^2	5.2	ret_9^2	6.37	Ret_max_9^2	6.78
roa_6^2	5.18	C_2	6.28	ret_11^2	6.75
s2p_10^2	5	ret_2^2	6.28	C_2^2	6.65
ret_11^2	4.97	pe_7	6.25	beme_8	6.65
nop_0^2	4.85	investment_7	-6.16	free_cf_9	-6.65
roa_6	4.74	ret_6^2	6.1	ivc_0^2	6.63
s2p_11 ivc_10	4.52	sat_6	6.08	ret_5^2	6.48
roe_5^2	4.51	ret_11	-6.06	free_cf_8	-6.44
ldp_6^2	4.5	ret_3	-5.95	Idol_vol_0^2	6.38
Turnover_11^2	4.49	ivc_0^2	5.91	Ret_max_10^2	6.38
lev_5^2	4.48	ldp_6^2	5.91	Ret_max_3^2	6.25
roa_3	4.4	sat_6^2	5.78	sat_6^2	6.19
std_4^2	4.4	Ret_max_9^2	5.71	ret_3^2	6.18
delta_so_11^2	4.37	Idol_vol_2	5.69	me_2^2	6.1
Std_turnover_10^2	4.29	ret_2	-5.66	ret_6^2	6
Beta_daily_11^2	4.23	ret_3^2	5.43	Idol_vol_11	6
s2p_11 ivc_5	4.18	s2p_4	-5.36	ldp_6	-6
roc_8	4.01	ol_11	5.15	ret_11	-5.89
Std_volume_9^2	3.9	Turnover_9	-5.15	sga2s_9	5.73
s2p_11 noa_11	3.87	Idol_vol_4	5.1	shrout_9^2	5.72
ivc_11^2	3.74	e2p_10	5.03	s2p_0	5.67
noa_11 roa_6	3.54	free_cf_1	-5.02	rna_2^2	5.58
delta_shrout_11^2	3.5	nop_8	-4.95	s2p_4^2	5.55
Std_turnover_11^2	3.45	ivc_11^2	4.94	delta_so_5	-5.54

in both degree-1 and 2 polynomials. Thomas and Zhang (2002) first document that the ivc factor can predict stocks' future returns, which is consistent with earning management of firms. Given that ivc still plays an important role post 2002, the anomaly has not been traded away. Short-term previous return (ret_11 and ret_10) are strongly negatively significant especially for portfolios with large stocks, implying a short-term reversal, which is consistent with Avramov, Cheng, and Metzker (2019). Note that the signs of certain firm characteristics are different for different lags, which potentially reflect the path-dependent nature of AP.[29]

Other factors including Tobin's Q, pre-tax profit margin (ipm), Ratio of cash and short-term investment to total asset (C), idiosyncratic volatility (Idol_vol) etc. are also prominent. Among them, idiosyncratic volatility (Idol_vol), max per daily return in a month (Ret_Max), etc., are arbitrage constraints and market signals related to trading; growth in external financing (fcf), operating income before depreciation and tax (ipm), etc., are financial signals related to firms' fundamentals. Trading signals affect stock returns through mispricing channel while financial signals do so likely through risk channel (Livdan, Sapriza, and Zhang, 2009). The patterns not only imply that future studies could focus on time-varying relevance of a small set of economic mechanisms and variables, but also tell researchers which of the features' non-linear effects to consider.

To further analyze the rotation patterns of dominant features, we compute the Pearson correlation coefficients both pair-wise and between trading (Idol_vol, Idol_vol^2, Ret_max, ret) and financial (Q, C, C^2, delta_so, ivc, ipm^2, ivc^2, investment, free_cf) signals. Table 10 reveals that inventory changes (ivc) and pre-tax profit margin to the second power (ipm^2) take turns to play important roles in AlphaPortfolio construction, and so do corporate liquidity (C, C^2) and changes in shares outstanding (delta_so) among others. Moreover, features related to trading and those related to financials also take turns to dominate. The rotation patterns are highly significant based on the P-values.

Algorithm 2 shown next describes an alternative distillation exercise. While Algo.1 mimics AlphaPortfolio on out-of-sample tests, Algo.2 distills what AlphaPortfolio has learned from the training set. The former gives information on how a model behaves on a test set while the latter describes what the model learns from a training set. We find similar results using Algo.1 and omit the details. Both algorithms can be extended to capture persistent latent variables in future work.

Overall, economic distillation provides us a basis to better understand and interpret our machine learning model. It informs us of the key input features and the way they matter (interaction or higher-order, etc) so that we can adjust the model accordingly when the

[29]Although not reported here, we also perform the analysis in Table 8 focusing on the top 10 features with 12 month lags. While some variables still exhibit sign changes that indicate path dependence, the signs on ivc, idol_vol, Q, ret, Ret_max consistently follow what theory would prescribe.

Figure 5: The following three heatmaps illustrate how the ranking of dominant features change over time. The most dominant features are inventory change (ivc), idiosyncratic volatility (Idol_vol), change in shares outstanding (delta_so), Tobin's Q (Q), cash and short-term investments to total assets (C), maximum daily return in the month (Ret_max), Pre-tax profit margin to the second power(ipmˆ2), ivc to the second power (ivcˆ2), investment, cash flow to book value of equity (free_cf), sale-to-price (s2p), C to the second power (Cˆ2), standard deviation of daily turnover (Std_volume), Idol_vol to the second power (Idol_volˆ2) and monthly return (ret). Appendix B provide detailed description of the features.

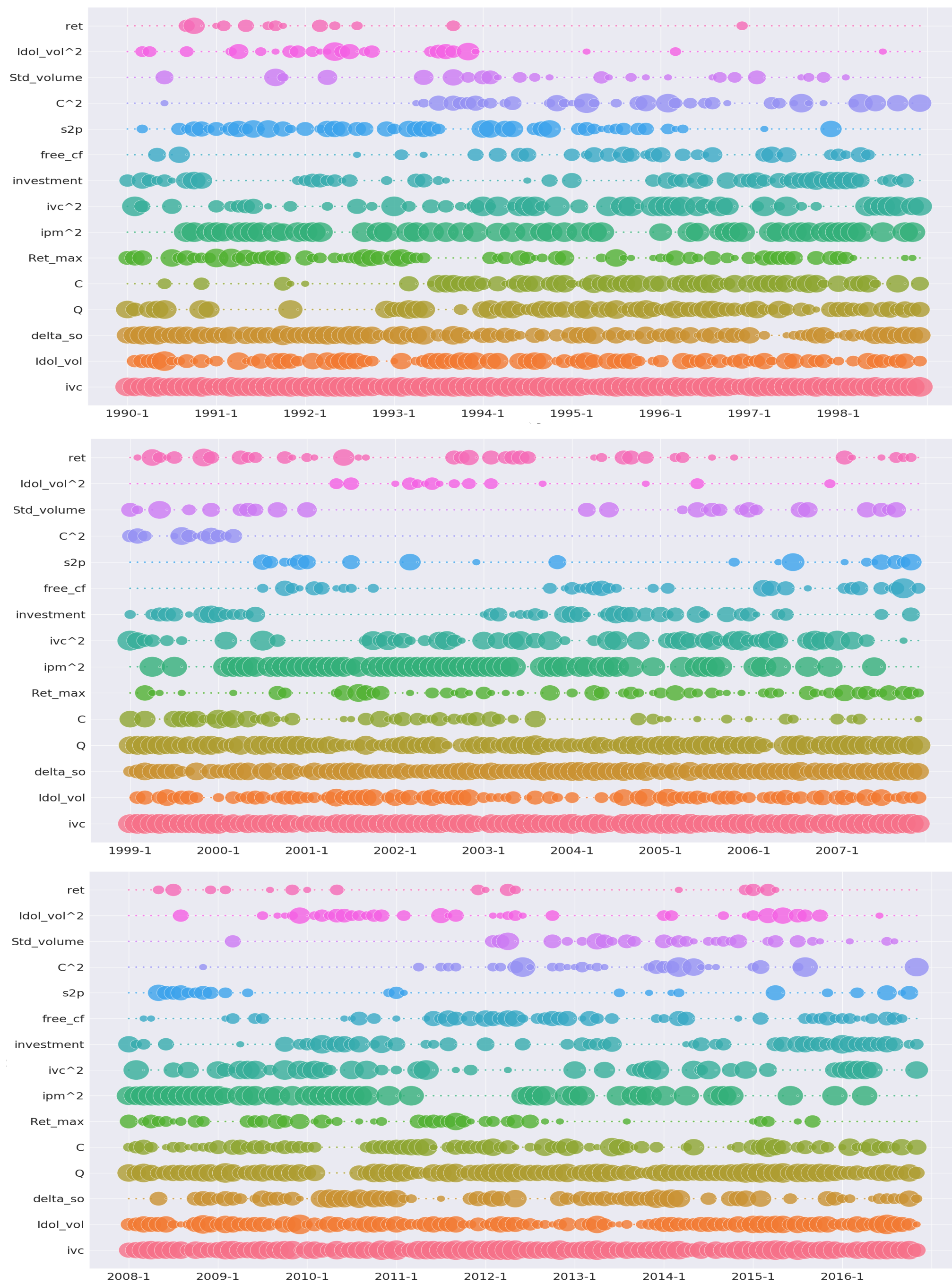

Table 9: Dominant Features after Economic Distillation

This table presents the probability of features ranked in top three across different months in the distillation Algorithm 1. In each month of out-of-sample periods, the AlphaPortfolio is distilled into a second degree polynomial function (Panel A) and a one degree polynomial function (Panel B). The importance of features are calculated by summing up the absolute value of historical 12 months' regression coefficients. Features are hence ranked by their importance in each month.

Panel A: Polynomial Degree Two					
Rank 1		Rank 2		Rank 3	
ipmˆ2	45.7%	ivc	36.7%	ivc	24.7 %
ivc	21.0 %	Q	16.7 %	Q	17.9 %
ivcˆ2	9.9 %	ivcˆ2	8.3 %	delta_so	14.8%
sga2sˆ2	5.2%	delta_so	6.5 %	C	8.3%
investmentˆ2	3.7%	pmˆ2	6.2%	Idol_vol	5.6%
Q	2.2%	investmentˆ2	3.7%	ivcˆ2	4.0%
Panel B: Polynomial Degree One					
Rank 1		Rank 2		Rank 3	
ivc	63.3%	ivc	25.3%	Idol_vol	18.8%
ipm	21.9%	Idol_vol	19.8%	delta_so	13.3%
Idol_vol	4.6%	free_cf	8.3 %	C	13.3%
investment	1.9%	ret	7.7 %	ret	9.3%
free_cf	1.5%	Ret_max	5.2 %	ivc	9.3%
sga2s	1.2%	C	5.2 %	Ret_max	9.0%

Algorithm 2: pseudo code for economic distillation approach two

Input: Model parameters $\boldsymbol{\theta}$ of AlphaPortfolio trained on $\mathcal{D}_{train}$; Training set $\mathcal{D}_{train}$; Test data set $\mathcal{D}_{test} = \{\mathcal{X}_1, \ldots, \mathcal{X}_N\}$; Hyper parameters $\{K, \alpha, p\}$;

Output: Evaluation metrics of test period;

Generate winner score $\boldsymbol{S}$ of all trading periods in $\mathcal{D}_{train}$ using AP;
Calculate gradient based sentivity for each raw feature q as $\mathbb{E}\left[\delta_{x_q}\right]$;
Select characteristics with top $K\%$ of $\mathbb{E}\left[\left|\delta_{x_q}\right|\right]$;
Generate p-degree polynomial terms with selected characteristics $\boldsymbol{q}^{selected}$;
Select important terms with Lasso regression(Penalty factor $= \alpha$);
Regress $\boldsymbol{S}$ to selected terms, obtain corresponding coefficient;
for $n = 1$ to N **do**
    Use $\mathcal{X}_n$ to construct selected terms;
    Re-generate winner score $\boldsymbol{s}'_n$ using selected terms and coefficients;
    Using $\boldsymbol{s}'_n$ to calculate rate of return r_n of this trading period;
end for
Calculate evaluation metrics given $R = \{r_1, \ldots, r_N\}$;

Table 10: Pearson Correlation Coefficients and Feature Rotations

This table presents the Pearson correlations among prominent features constructed from firm characteristics and market signals. Panel A contains the pair of dominant features with the most negative Pearson correlations. Panel B contains the correlation between two subsets of the most dominant features: trading and financial. Trading subset contains {Idol_vol, Idol_vol^2, Ret_max, ret} and financial subset contains {Q, C, C^2, delta_so, ivc, ipm^2, ivc^2, investment, free_cf}.

Term 1	Term 2	Correlation	P-value
Panel A: Pairwise Correlation of Dominant Features			
ivc	ipm^2	-0.38	1.55×10^{-12}
delta_so	C^2	-0.33	1.40×10^{-12}
C	delta_so	-0.31	2.09×10^{-8}
C^2	ret	-0.26	1.47×10^{-6}
investment	s2p	-0.25	4.20×10^{-6}
Idol_vol	ipm^2	-0.24	1.22×10^{-5}
ipm^2	C^2	-0.23	3.79×10^{-5}
Ret_max	C^2	-0.22	8.43×10^{-5}
C	ret	-0.21	1.80×10^{-4}
delta_so	Idol_vol^2	-0.2	2.90×10^{-4}
Panel B: Trading Signals VS. Financial Signals			
Trading_related	Financial_related	-0.33	8.00×10^{-10}

economic environment changes.[30] It also serves as a sanity check of the complex machine learning model in the sense that if something in the distilled model appears strange, such as a stale price playing a dominant role when it should have been subsumed in other features, it is likely that the complex model contains a mis-specification or coding error.[31]

5.2 Textual Factor Analysis for Economic Interpretation

The concept of projecting a complex model onto simpler spaces can be extended to natural languages too. The main idea behind a textual factor analysis is that texts are written in natural languages and if we find correlations of AlphaPortfolio holdings with the topics discussed in text documents, we may be able to develop a narrative or a better understanding of the model.

[30]Interaction terms do not contribute to AP's performance significantly. But this is not at odds with studies that emphasize interactions effects because those studies focus on estimating SDF or minimizing estimation errors in assets' return moments whereas we focus on portfolio performance.

[31]Appendix C provides a case in point of how economic distillation can help assess model stability and functionality, especially on test sets, which computer science studies typically leave out.

To do this, we use the general framework Cong, Liang, and Zhang (2018) introduce for analyzing large-scale text-based data. The methodology combines the strengths of neural network language models and generative statistical modeling. It generates textual factors by (i) representing texts using vector word embedding, (ii) clustering words using locality-sensitive hashing, and (iii) identifying spanning vector clusters through topic modeling. Arguably, one can use other text analytics but the data-driven approach in Cong, Liang, and Zhang (2018) captures complex linguistic structures while ensuring computational scalability and economic interpretability.[32]

Specifically, we use Google's word2vec for the word embedding step and follow Cong, Liang, and Zhang (2018) to generate textual factors from firms' MD&A documents. They are essentially topics and themes with relative frequency on a set of words and phrases that span the textual space. We then regress each firm's MD&A document on the textual factors to obtain the loadings on each topic.

Mathematically, let K denote the number of textual factors, where K is endogenously specified and can potentially depend on the data (we use 200 for simplicity). Denote the set of textual factors by the triplet $(S_i, F_i \in \mathbb{R}^{|S_i|}, d_i \in \mathbb{R}_{\geq 0})$, where S_i denotes the support of word-cluster $i = 1, \cdots, K$, F_i is a real-valued vector representing the textual factor indicating the relative frequencies of words of factor, and the factor importance d_i. Given the textual factors and a firm's document document D (represented by a document-term vector $N^{(D)} \in \mathbb{R}^V$, where V is the size of the vocabulary in the texts), the loading of the textual factor i is simply the projection

$$x_i^{(D)} := \frac{\langle N_{S_i}^{(D)}, F_i \rangle}{\langle F_i, F_i \rangle}, \tag{12}$$

and the document D can be represented quantitatively as $(x_1^{(D)}, \dots x_K^{(D)}) \in \mathbb{R}^K$.

To understand the meaning of these loadings (textual βs), think about how a company continuously discusses and discloses information on profitability, social responsibility, innovativeness, etc., through MD&A. The $x_k^{(D)}$ we obtain allows us to assign a number that measures how much the company loads on that topic—a metric we can use in simple sparse regression framework.

The final step is to regress each stock's winner score in each month from AP onto the

[32]The difficulties in analyzing textual data are three-fold: first, language structures are intricate and complex, and representing or summarizing them using simple frequency/count-based approaches is highly reductive and may lose important informational content; second, textual data are high-dimensional and processing a large corpus of documents is computationally demanding; third, there lacks a framework relating textual data to sparse regression analysis traditionally used in social sciences while maintaining interpretability. In the paper, the authors also discuss applications of textual factors in (i) prediction and inference, (ii) interpreting existing models and variables, and (iii) constructing new metrics and explanatory variables, with illustrations using topics in economics such as macroeconomic forecasting and factor asset pricing.

Table 11: Winner Scores' Loadings on Textual Factors

This table contains examples of the most loaded topics/textual factors when firms' winner scores are regressed on over 30 contemporaneous textual factor loadings selected based on factor importance and domain expertise. The regression coefficients are reported under the textual topics, where a positive (negative) number indicates a long (short) position on stocks with the topic prominently showing up in the firm's filings. The remainder columns list words within each textual factor. We do not stem words because we use word vectors trained directly by Google word2vec in the word embedding step.

Topics	**Most Frequent Words in the Textual Factor**				
Loss-cutting (0.1500)	deferral shutdown decrease recycled	curtailment abolishment reclassification afford	divestiture retrenchment amortization salable	diminution imposition resell	cutback reductions resale
Profitability (0.1358)	profit profits profitable yoy	ebit pretax writedown gross_margin	quarterly revenue business net_income	earnings revenues unprofitable efficiency	profitably viable net_loss operational
Sales (0.0428)	sales profit shipments	purchases income sale	growth profits fiscal	retail comps revenue	earnings resales
Cash/Invest (0.0381)	subsidizing invest	unaffordable paying	reimburse afford	reinvestment cash_trapped	subsidy tariffs
Actions (0.0339)	secures acquire completion	closing introduces donates	unveils signs_agreement acquires	receives deploys announces	exploring stopped delayed
Inventory (-0.0209)	hoarding accumulating rationing transfers	replenishing distributing buying seasonal	stockpiling producing storing restocked	overcharging restock restocking shipping	supplying stockpile inventory stocked
Mistakes (-0.0536)	forgiveness contrition apologize clarifications	confess forgiven punish sorry	forgives wrongs repay forgave	admit excuses mistake repent	forgiving atonement amends redress
Uncertainty (-0.1411)	volatility speculative turbulance	speculator risky changing	irrationals turmoil evolving	traders instability unpredictable	fluctuation uncertainty hedge
Corporate Events (-0.1444)	recapitalization buyout acquisitions takeovers	divestitures acquire acquired synergies	mergers transaction divestiture expansions	divestiture acquisitive merger takeover	unbundling restructure amalgamated takeover
Real Estate (-0.2362)	bungalows residences homes condominiums apartment rented	dwellings cottages buildings townhome farmhouse residence	acres barns condos real_estate cottage duplex	houses cabins lofts foreclosure bedroom ranch	carport outbuilding mansion backyard villa motorhome

contemporaneous textual βs. We iterate the process a few times to reduce the number of textual factors based on their interpretability, importance in the MD&A data, and significance in the AP construction. Specifically, after each iteration, we discard word clusters that are incoherent or are infrequently mentioned in the firms' filing or have insignificant correlations with the winner score. Table 11 contains examples of the most loaded topics/textual factors when constructing AP. A positive coefficient indicates when discussions on the topic dominates the firm's text data, the firm's stock more likely receives a long position; a negative coefficient indicates the opposite. The word lists are the corresponding words within each textual factor. The stocks AP buys typically mention issues such as loss-cutting, sales, and actions, as well as profitability, cash, and investment that are related to (C, Cˆ2, investment, ipm, and ipmˆ2 from the polynomial sensitivity analysis; the stocks AP short-sells are the ones heavily discussing real estates, corporate events, and acknowledging mistakes, as well as uncertainty and inventory, which relate to Cˆ2, delta_so, ivc, ivcˆ2 and Idol_vol and Idol_volˆ2. Further analysis can be conducted. For example, one can relate the negative loading on corporate events textual factor to theories explaining why stock returns may be negative on average for firms going through certain corporate events.

This textual factor analysis only constitutes an initial step towards developing a narrative for how AP behaves. Our simple choice of text data and the application of textual factors are not meant to be optimal and definitive but serve as an illustration of interpreting AI models with texts. More comprehensive analysis in combination with economic theory constitutes future research. For parsimony, we have only included representative textual factors. More implementation details can be found in Cong, Liang, and Zhang (2018).

6 Reinforcement Learning and Interpretable AI

Beyond investment and portfolio management, our model and empirical findings have two broader implications on (i) the utility of reinforcement learning in social sciences and (ii) the importance of and an initial step towards economically interpretable AI.

6.1 Utility of Reinforcement Learning in Social Sciences

Reinforcement learning is a general framework for goal-directed learning in an unknown environment, requiring a learner to explore the environment sufficiently to "discover" where to find the maximum reward and to learn adequately about the dynamics of the environment to devise a policy which delivers the maximum reward in expectation. Supervised learning, in contrast, involves extra knowledge of the environment by way of examples of desirable behaviour. RL is routinely used in and has seen great commercial success with applications for speech recognition, natural language processing, computer vision, interactive games, etc., with Amazon Alexa, Apple Siri, AlphaGo, and Google Android as leading examples.

Our study demonstrates that RL's dynamic programming approach can be useful in handling business or economic environments that are often more complex and dynamic than those in science and engineering. In contrast, models trained on past market dynamics under supervised learning may not be generalizable to the future market if there are substantial changes or distributional shifts. Moreover, most models such as regressions, SVM, and neural networks have RL-based implementations.

Portfolio management is a fitting application for RL because it is a data-driven stochastic optimal control problem involving sequential decision-making in allocating and reallocating funds into assets based on the latest information in an uncertain environment.[33]

More generally, RL can be applied to business and social science problems when the decision problem is complex (e.g., entailing large state spaces and dynamic optimization), with a well-specified objective, but limited pre-existing knowledge or data to work with to provide full solutions.[34]

6.2 Interpreting AI and Big Data in Social Sciences

Use of big data and AI often presents prejudice against groups of individuals, especially those who have been disadvantaged historically.[35] While people often attribute these issues to training data, algorithms and models may also have embedded stories and ideology due to designers' negligence or cultural insensitivity. Algorithms thus may (unintentionally) perpetuate initial random errors and induce undesirable behaviors by catering to users' addiction and bigotry. In a sense, it is equally important to understand a black-box model as it is to carefully process and sample data.

The perils of AI and big data are particularly worrisome in social sciences because of their fast and non-stationary dynamics. We have to recognize that unlike genetics and physical

[33]Note that our model can be extended to incorporate specific investor's preference, or elements such as borrowing constraints, risk tolerance, tax benefits etc., by modifying the performance (reward) function. For example, we can certainly replace the Sharpe ratio to mean-variance or penalize a over-borrowing strategy in the performance function. This is also a key advantage of RL algorithms relative to other algorithms based on return prediction. As such, our RL algorithm has a potential to be used widely in robo-advising, which for parsimony we leave to future exploration.

[34]One still needs good simulation data or historical data in practice, because teaching machines through trial and error is too socially controversial or costly (potentially incurring trading losses).

[35]For example, COMPAS that guides criminal sentencing in the United States could introduce racial biases, according to a report by ProPublica; Algorithm PredPol designed to predict crime locations leads police to unfairly target certain neighborhoods; Joy Buolamwini points out that gender-recognition AIs from IBM Microsoft and Chinese company Megvii increas the risk of false identification of women and minorities; even Google searches could propagate biases against women CEOs and job-seekers. Such phenomena have consequential socio-economic impacts on labor market dynamics, wealth inequality, etc.

laws, business environments and financial markets are evolving constantly. Policies and consumer preferences change all the time. We cannot always take machine learning packages and big data analytics off the shelf and apply them blindly to problems in economics and finance. Without knowing what a model does, superior out-of-sample tests amount to little because one could be dealing with a new environment that technically has no "historical data" that are the life blood of machine learning models. Most current applications of machine learning and big data analytics provide little guidance on how we should understand, trust, and subsequently adjust (when the environment changes) the models. Service providers and entrepreneurs using them are also having a difficult time building trust among consumers, investors, and regulators.

A necessary step to address all aforementioned concerns is to understand complex AI and machine learning models. Polynomial sensitivity analysis builds on current practices in computer science yet allows a flexible framework. For example, one can put in third-order and fourth-order terms of a feature if one deems them important. Textual-factor analysis derives from topic modeling and word embedding, and constitutes one of the many possibilities of using natural languages to better explain model behaviors. Both procedures are, intuitively, projections of complex models into transparent and interpretable spaces. Even though we use specific implementations in this paper, our general approach allows other variants and new candidates for economic distillation.

7 Conclusion

We propose a reinforcement learning-based portfolio management model, an alternative that improves upon the traditional two-step portfolio-construction paradigm to directly optimize investors' objectives. We invent a multi-sequence learning model building on a latest AI development in order to effectively capture the high-dimensional, non-linear, noisy, interacting, and dynamic nature of economic data and market environments. The resulting AlphaPortfolio yields stellar out-of-sample performances even with various economic and trading restrictions, making the framework deployable by practitioners for trading and investment advising.

While supervised and unsupervised learnings have been widely applied in economics and finance, reinforcement learning has not received much attention. We provide economic motivations for its application by identifying suitable applications and illustrating its efficacy, which have implications beyond portfolio theory and more broadly in social sciences. Moreover, using polynomial-feature-sensitivity and textual-factor analyses, we project the AlphaPortfolio model onto linear regression and natural language spaces for greater transparency and better interpretation. Our "economic distillation" not only reveals key firm characteristics (including their rotation and non-linearity) that drive AlphaPortfolio's performance, but also provides a concrete backbone for and an incremental step towards economic interpretations of machine learning and AI applications in finance and business fields.

References

Aït-Sahali, Yacine, and Michael W Brandt, 2001, Variable selection for portfolio choice, *The Journal of Finance* 56, 1297–1351.

Alsabah, Humoud, Agostino Capponi, Octavio Ruiz Lacedelli, and Matt Stern, 2019, Robo-advising: Learning investor's risk preferences via portfolio choices, *Available at SSRN 3228685.*

Athey, Susan, 2018, The impact of machine learning on economics, in *The economics of artificial intelligence: An agenda* (University of Chicago Press).

——— , and Guido Imbens, 2017, The state of applied econometrics: Causality and policy evaluation., *The Journal of Economic Perspectives* 31, 3–328.

Avramov, Doron, Si Cheng, and Lior Metzker, 2019, Machine learning versus economic restrictions: Evidence from stock return predictability, *Available at SSRN 3450322.*

Avramov, Doron, Tarun Chordia, Gergana Jostova, and Alexander Philipov, 2013, Anomalies and financial distress, *Journal of Financial Economics* 108, 139–159.

Bailey, David H, and Marcos Lopez de Prado, 2012, *Journal of Investment Strategies (Risk Journals)*1.

Baker, Malcolm, and Jeffrey Wurgler, 2007, Investor sentiment in the stock market, *Journal of Economic Perspectives* 21, 129–152.

Barocas, Solon, Moritz Hardt, and Arvind Narayanan, 2017, Fairness in machine learning, *NIPS Tutorial.*

Barry, Christopher B, 1974, Portfolio analysis under uncertain means, variances, and covariances, *The Journal of Finance* 29, 515–522.

Bartlett, Robert, Adair Morse, Richard Stanton, and Nancy Wallace, 2019, Consumer-lending discrimination in the fintech era, Discussion paper, National Bureau of Economic Research.

Bawa, Vijay S, Stephen J Brown, and Roger W Klein, 1979, Estimation risk and optimal portfolio choice, *NORTH-HOLLAND PUBL. CO., N. Y., 190 pp.*

Best, Michael J, and Robert R Grauer, 1991, On the sensitivity of mean-variance-efficient portfolios to changes in asset means: some analytical and computational results, *The Review of Financial Studies* 4, 315–342.

Bianchi, Daniele, Matthias Büchner, and Andrea Tamoni, 2019, Bond risk premia with machine learning, *USC-INET Research Paper.*

Black, Fischer, and Robert Litterman, 1990, Asset allocation: combining investor views with market equilibrium, Discussion paper, Discussion paper, Goldman, Sachs & Co.

——— , 1992, Global portfolio optimization, *Financial Analysts Journal* 48, 28–43.

Brandt, Michael W, 1999, Estimating portfolio and consumption choice: A conditional euler equations approach, *The Journal of Finance* 54, 1609–1645.

——— , 2010, Portfolio choice problems, in *Handbook of financial econometrics: Tools and techniques* . pp. 269–336 (Elsevier).

——— , and Pedro Santa-Clara, 2006, Dynamic portfolio selection by augmenting the asset space, *The Journal of Finance* 61, 2187–2217.

——— , and Rossen Valkanov, 2009, Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns, *The Review of Financial Studies* 22, 3411–3447.

Bryzgalova, Svetlana, Markus Pelger, and Jason Zhu, 2020, Forest through the trees: Building cross-sections of stock returns, *Working Paper*.

Bucilu, Cristian, Rich Caruana, and Alexandru Niculescu-Mizil, 2006, Model compression, in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 535–541. ACM.

Carhart, Mark M, 1997, On persistence in mutual fund performance, *The Journal of Finance* 52, 57–82.

Chen, Luyang, Markus Pelger, and Jason Zhu, 2020, Deep learning in asset pricing, Discussion paper, Working paper.

Chinco, Alexander M, Andreas Neuhierl, and Michael Weber, 2019, Estimating the anomaly base rate, Discussion paper, National Bureau of Economic Research.

Chordia, Tarun, Avanidhar Subrahmanyam, and Qing Tong, 2014, Have capital market anomalies attenuated in the recent era of high liquidity and trading activity?, *Journal of Accounting and Economics* 58, 41–58.

Cochrane, John H, 2011, Presidential address: Discount rates, *The Journal of Finance* 66, 1047–1108.

Cong, Lin William, Shiyang Huang, and Douglas Xu, 2020, Rise of factor investing: security design and asset pricing implications, .

Cong, Lin William, Tengyuan Liang, Baozhong Yang, and Xiao Zhang, 2019, Analyzing textual information at scale, *Economic Information to Facilitate Decision Making* Forthcoming, edited by Kashi Balachandran.

Cong, Lin William, Tengyuan Liang, and Xiao Zhang, 2018, Textual factors: A scalable, interpretable, and data-driven approach to analyzing unstructured information, Discussion paper, resubmission requested.

Cong, Lin William, Ke Tang, Jingyuan Wang, and Yang Zhang, 2020, Deep sequence modeling: Development and applications in asset pricing, *Journal of Financial Data Science* Forthcoming.

D'Acunto, Francesco, and Alberto G Rossi, 2020, Robo-advising, .

Dasa, Sanjiv R, Daniel Ostrova, Anand Radhakrishnanb, and Deep Srivastavb, 2018, A new approach to goals-based wealth management, *Journal Of Investment Management* 16, 1–27.

Datta, Anupam, and Shayak Sen, 2018, global cluster explanations for machine learning models, .

de Prado, Marcos López, 2016, Building diversified portfolios that outperform out of sample, *The Journal of Portfolio Management* 42, 59–69.

De Prado, Marcos Lopez, 2018, *Advances in financial machine learning* (John Wiley & Sons).

DeMiguel, Victor, Lorenzo Garlappi, and Raman Uppal, 2007, Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy?, *The Review of Financial Studies* 22, 1915–1953.

Deng, Yue, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai, 2016, Deep direct reinforcement learning for financial signal representation and trading, *IEEE Transactions on Neural Networks and Learning Systems* 28, 653–664.

———, 2017, Deep direct reinforcement learning for financial signal representation and trading, *IEEE TNNLS* 28, 653–664.

Detemple, Jerome, and Shashidhar Murthy, 1997, Equilibrium asset prices and no-arbitrage with portfolio constraints, *The Review of Financial Studies* 10, 1133–1174.

Ding, Yi, Weiqing Liu, Jiang Bian, Daoqiang Zhang, and Tie-Yan Liu, 2018, Investor-imitator: A framework for trading knowledge extraction, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* pp. 1310–1319. ACM.

Fama, Eugene F, and Kenneth R French, 2015, A five-factor asset pricing model, *Journal of Financial Economics* 116, 1–22.

———, 2018, Choosing factors, *Journal of Financial Economics* 128, 234–252.

Farboodi, Maryam, Laura Veldkamp, et al., 2019, Long run growth of financial data technology, Discussion paper, .

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu, 2020, Taming the factor zoo: A test of new factors, *The Journal of Finance* 75, 1327–1370.

Feng, Guanhao, Jingyu He, and Nicholas G Polson, 2018, Deep learning for predicting asset returns, *arXiv preprint arXiv:1804.09314*.

Feng, Guanhao, Nick Polson, and Jianeng Xu, 2018, Deep learning factor alpha, *Available at SSRN 3243683*.

Fischer, Thomas G, 2018, Reinforcement learning in financial markets-a survey, Discussion paper, FAU Discussion Papers in Economics.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber, 2019, Dissecting characteristics nonparametrically, *Working paper*.

Friedman, Jerome H., 2002, Stochastic gradient boosting, *Computational Statistics & Data Analysis* 38, 367378.

Garlappi, Lorenzo, Raman Uppal, and Tan Wang, 2006, Portfolio selection with parameter and model uncertainty: A multi-prior approach, *The Review of Financial Studies* 20, 41–81.

Garychl, 2018, Applications of reinforcement learning in real world, *Medium* Aug 1, 2018.

Goldfarb, Donald, and Garud Iyengar, 2003, Robust portfolio selection problems, *Mathematics of Operations Research* 28, 1–38.

Goodman, Bryce, and Seth Flaxman, 2017, European union regulations on algorithmic decision-making and a right to explanation, *AI Magazine* 38, 50–57.

Green, Richard C, and Burton Hollifield, 1992, When will mean-variance efficient portfolios be well diversified?, *The Journal of Finance* 47, 1785–1809.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2018, Empirical asset pricing via machine learning, Discussion paper, National Bureau of Economic Research.

Gu, Shihao, Bryan T Kelly, and Dacheng Xiu, 2019, Autoencoder asset pricing models, *Available at SSRN.*

Guidotti, Riccardo, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi, 2018, A survey of methods for explaining black box models, *ACM computing surveys (CSUR)* 51, 1–42.

Han, Yufeng, Ai He, David Rapach, and Guofu Zhou, 2018, What firm characteristics drive us stock returns?, *Available at SSRN 3185335.*

Harvey, Campbell R, Yan Liu, and Heqing Zhu, 2016, ...and the cross-section of expected returns, *The Review of Financial Studies* 29, 5–68.

Heaven, Douglas, 2019, Why deep-learning ais are so easy to fool, *Nature (News Feature)* October 9.

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean, 2015, Distilling the knowledge in a neural network, *arXiv preprint arXiv:1503.02531.*

Horel, Enguerrand, and Kay Giesecke, 2019a, Computationally efficient feature significance and importance for machine learning models, *arXiv preprint arXiv:1905.09849.*

——— , 2019b, Towards explainable ai: Significance tests for neural networks, *arXiv preprint arXiv:1902.06021.*

Hou, Kewei, Chen Xue, and Lu Zhang, 2015, Digesting anomalies: An investment approach, *The Review of Financial Studies* 28, 650–705.

——— , 2020, Replicating anomalies, *The Review of Financial Studies* 33, 20192133.

Jin, Olivier, and Hamza El-Saawy, 2016a, Portfolio management using reinforcement learning, Discussion paper, Working paper, Stanford University.

——— , 2016b, Portfolio management using reinforcement learning, Discussion paper, Stanford University.

Jobson, JD, 1979, Improved estimation for markowitz portfolios using james-stein type estimators, in *Proceedings of the American Statistical Association, Business and Economics Statistics Section* vol. 71 pp. 279–284.

Jorion, Philippe, 1986, Bayes-stein estimation for portfolio analysis, *Journal of Financial and Quantitative Analysis* 21, 279–292.

Jurczenko, Emmanuel, 2015, *Risk-based and factor investing* (Elsevier).

Kan, Raymond, and Guofu Zhou, 2007, Optimal portfolio choice with parameter uncertainty, *Journal of Financial and Quantitative Analysis* 42, 621–656.

Karolyi, Andrew, and Stijn Van Nieuwerburgh, 2020, New methods for the cross-section of returns, *The Review of Financial Studies* Forthcoming.

Kelly, Bryan T, Seth Pruitt, and Yinan Su, 2019, Characteristics are covariances: A unified model of risk and return, *Journal of Financial Economics.*

Kim, Soohun, Robert A Korajczyk, and Andreas Neuhierl, 2019, Arbitrage portfolios, *Available at SSRN.*

Klein, Roger W, and Vijay S Bawa, 1976, The effect of estimation risk on optimal portfolio choice, *Journal of Financial Economics* 3, 215–231.

Koijen, Ralph SJ, Tobias J Moskowitz, Lasse Heje Pedersen, and Evert B Vrugt, 2018, Carry, *Journal of Financial Economics* 127, 197–225.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, 2020, Shrinking the cross-section, *Journal of Financial Economics* 135, 271–292.

Krause, Josua, Adam Perer, and Kenney Ng, 2016, Interacting with predictions: Visual inspection of black-box machine learning models, in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* pp. 5686–5697.

Light, Nathaniel, Denys Maslov, and Oleg Rytchkov, 2017, Aggregation of information about the cross section of stock returns: A latent variable approach, *The Review of Financial Studies* 30, 1339–1381.

Linnainmaa, Juhani T, and Michael R Roberts, 2018, The history of the cross-section of stock returns, *The Review of Financial Studies* 31, 2606–2649.

Liu, John Zhuang, Michael Sockin, and Wei Xiong, 2020, Data privacy and temptation, *Working Paper*.

Livdan, Dmitry, Horacio Sapriza, and Lu Zhang, 2009, Financially constrained stock returns, *The Journal of Finance* 64, 1827–1862.

Lo, Andrew W., 2016, Imagine if robo advisers could do emotions, *Wall Street Journal* June 6.

Markowitz, Harry, 1952, Portfolio selection, *The Journal of Finance* 7, 77–91.

Martin, Ian, and Stefan Nagel, 2019, Market efficiency in the age of big data, *Working Paper*.

McLean, R David, and Jeffrey Pontiff, 2016, Does academic research destroy stock return predictability?, *The Journal of Finance* 71, 5–32.

Merton, Robert C, 1980, On estimating the expected return on the market: An exploratory investigation, *Journal of Financial Economics* 8, 323–361.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., 2015, Human-level control through deep reinforcement learning, *Nature* 518, 529–533.

Moody, John, Lizhong Wu, Yuansong Liao, and Matthew Saffell, 1998, Performance functions and reinforcement learning for trading systems and portfolios, *Journal of Forecasting* 17, 441–470.

Moritz, Benjamin, and Tom Zimmermann, 2016, Tree-based conditional portfolio sorts: The relation between past and future stock returns, *Available at SSRN 2740751*.

Nagel, Stefan, 2012, Evaporating liquidity, *The Review of Financial Studies* 25, 2005–2039.

——— , 2019, Machine learning and asset pricing, *Princeton Lectures in Finance*.

Neuneier, Ralph, 1996, Optimal asset allocation using adaptive dynamic programming, in *Advances in Neural Information Processing Systems* pp. 952–958.

Novy-Marx, Robert, and Mihail Velikov, 2016, A taxonomy of anomalies and their trading costs, *The Review of Financial Studies* 29, 104–147.

Pástor, L'uboš, 2000, Portfolio selection and asset pricing models, *The Journal of Finance* 55, 179–223.

——— , and Robert F Stambaugh, 2000, Comparing asset pricing models: an investment perspective, *Journal of Financial Economics* 56, 335–381.

——— , 2003, Liquidity risk and expected stock returns, *Journal of Political Economy* 111, 642–685.

Powell, James L, James H Stock, and Thomas M Stoker, 1989, Semiparametric estimation of index coefficients, *Econometrica: Journal of the Econometric Society* pp. 1403–1430.

Rapach, David, and Guofu Zhou, 2019, Time-series and cross-sectional stock return forecasting: New machine learning methods, *Available at SSRN 3428095.*

Rapach, David E, Jack K Strauss, and Guofu Zhou, 2013, International stock return predictability: what is the role of the united states?, *The Journal of Finance* 68, 1633–1662.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin, 2016, Why should i trust you?: Explaining the predictions of any classifier, in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* pp. 1135–1144. ACM.

Rossi, Alberto G, 2018, Predicting stock market returns with machine learning, Discussion paper, Working paper.

Sak, Halis, Tao Huang, and Michael T Chng, 2019, Exploring the factor zoo with a machine-learning portfolio, *Working Paper.*

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al., 2017, Mastering the game of go without human knowledge, *Nature* 550, 354–359.

Stambaugh, Robert F, Jianfeng Yu, and Yu Yuan, 2012, The short of it: Investor sentiment and anomalies, *Journal of Financial Economics* 104, 288–302.

Stambaugh, Robert F, and Yu Yuan, 2017, Mispricing factors, *The Review of Financial Studies* 30, 1270–1315.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan, 2017, Axiomatic attribution for deep networks, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* pp. 3319–3328. JMLR. org.

Thomas, Jacob K, and Huai Zhang, 2002, Inventory changes and future returns, *Review of Accounting Studies* 7, 163–187.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, 2017, Attention is all you need, in Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, ed.: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA* pp. 5998–6008.

Wager, Stefan, and Susan Athey, 2018, Estimation and inference of heterogeneous treatment effects using random forests, *Journal of the American Statistical Association* 113, 1228–1242.

Wang, Jingyuan, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong, 2019, Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* pp. 1900–1908.

Wu, Mike, Michael C Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez, 2018, Beyond sparsity: Tree regularization of deep models for interpretability, in *Thirty-Second AAAI Conference on Artificial Intelligence.*

Appendix A. Basics of Reinforcement Learning

Articles and books on the basics of neural networks and their applications are widely available. In this appendix, we briefly introduce the basics of reinforcement learning (as opposed to supervised or unsupervised learning).

Reinforcement learning (RL) is "learning what to do — how to map situations to actions— so as to maximize a numerical reward signal" (Sutton and Barto, 2008). RL is one of the three basic machine learning paradigms, alongside supervised learning and unsupervised learning. RL typically solves a reward maximization problem in a Markov-decision-process (MDP) setting in which an agent makes the best decision given its information set under a stochastically-evolving environment.

In RL, an agent must be able to learn about the state of its environment and take actions that potentially affect the state going forward. Below we denote the set of possible states to be S, and the set of possible actions to be A. Beyond the agent and the environment, the other four main elements of a reinforcement learning system are: a *policy*, a *reward signal*, a *value function*, and, optionally, a *model* of the environment.

1. *Policy*: A policy defines the agent's way of behaving at a given time. It is similar to the (behavioral) *strategy* concept in sequential games and determines behavior. In general, a policy is a mapping from perceived states of the environment to (possibly stochastic) actions to be taken when in those states.

2. *Reward Signal*: A reward signal $R \in \mathbb{R}$ defines the goal of a reinforcement learning problem. In each time step, the environment sends the agent a reward (usually a number) based on the current and subsequent states, and the actions taken.[36] The agent's sole objective is to maximize the total discounted reward it receives over the life time. The reward essentially drives the policy; if an action selected by the policy results in low expected reward, then the policy is modified to select some other action in that situation in future.

3. *Value*: Whereas the reward signal indicates the immediate payoff, a value function specifies the maximal total rewards an agent expects to accumulate in the long run, starting from the current state.[37] Whereas rewards determine the immediate, intrinsic desirability of environmental states, values reveal the long-term desirability of states after taking into account the states that are likely to follow and the rewards available in those states. For example, a state might always yield a low immediate reward but

[36] In a sense, this is similar to the state-dependent utility function in economics.

[37] This is similar to the multi-period utility function used in intertemporal choice models in economics, where future utility is subject to a discount compared with contemporaneous one.

still have a high value because it is regularly followed by other states that yield high rewards.

4. *Model*: A model approximates an agent's dynamic interaction with the environment. Specifically, a model (discrete-time, just for illustration) specifies $P\left((s_{i+1}, R_{i+i})|s_i, a_i\right)$, where subscripts denote time.

Formally, a policy can be described as $\pi(a|s)$, which represents the probability of taking action $a \in A$ given state $s \in S$. The value function of a state s under a policy π, denoted $v_\pi(s)$, is the expected payoff when starting in s and following π thereafter. Denote γ as the discount rate for rewards, A_t for the action in period t, and $R_t \equiv R_t(s_{t-1}, s_t)$ for the reward from period $t-1$ to period t. For MDPs, we can write the value of state s under policy π, $v_\pi(s)$, by:

$$v_\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^t R_{t+k+1}|S_t = s] = \mathbb{E}_\pi[G_t|S_t = s], \text{ for all } s \in S. \tag{13}$$

Upper case S_t corresponds to a random variable, and lower case s_t indicates a realized value of S_t. Here $G_t = \sum_{k=0}^{\infty} \gamma^t R_{t+k+1}$ is the total discounted reward after period t. The agent's problem is $\max_\pi v_\pi(s)$

By iterated law of expectations:

$$\begin{aligned}
v_\pi(s_t) =& \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s_t] \\
=& \mathbb{E}_\pi[R_{t+1} + \mathbb{E}_\pi[\sum_{k=1}^{\infty} \gamma^k R_{t+k+1}|S_t = s_t, S_{t+1} = s_{t+1}]|S_t = s_t] \\
=& \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s_t].
\end{aligned}$$

Note S_{t+1} is a random variable given $S_t = s_t$ when the model is stochastic. Therefore, the agent in RL simply solves a Bellman equation. More specifically, if the game is finite, i.e., there are some final states whose values are determined by the model, such as winning a Go game or successfully getting out of a maze, we can use these final states in a "backward induction" to get the value function in other states. In this sense, we have reduced the solution of a more complex problem (the value function of an "earlier" state which is far away from the end) to those of simpler problems (the value function of a "later" state which is closer to the end) with similar structure, which is exactly the thought behind finite horizon dynamic programming.

Similarly, we can define the value of choosing action a in state s under policy π, denoted by $q_\pi(s,a)$, by:

$$q_\pi(s,a) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^t R_{t+k+1} | S_t = s, A_t = a] = \mathbb{E}_\pi[G_t | S_t = s, A_t = a], \text{ for all } s \in S, a \in A(s), \tag{14}$$

where $A(s) \in A$ denotes all the possible actions under state s. It is obvious that $v(s) = \max_a q_\pi(s,a)$. And again,

$$\begin{aligned} q_\pi(s_t,a_t) =& \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s_t, A_t = a_t] \\ =& \mathbb{E}_\pi[R_{t+1} + \gamma \max_{a_{t+1}} q_\pi(S_{t+1}, a_{t+1}) | S_t = s_t, A_t = a_t], \end{aligned}$$

which is also in the form of Bellman equation. Solving this Bellman equation for $q(s,a)$ is the process called value-based learning, or Q-learning where we learn a value function that maps each state-action pair to a value. Q-learning works well when you have a finite (and small enough for computation) set of actions.

Another type of RL algorithm is policy-based learning, where we can deal with optimization over a continuum of possible policies. For instance, with a self-driving car, at each state you can have an infinite number of potential actions (turning the wheel at 15, 17.2, 19,4 degree).[38] Outputting a Q-value for each possible action for example under Q-learning would be infeasible. In policy-based learning, we directly optimize the parameters in a policy function.

Specifically, we define our policy that has a parameter vector $\theta \in \mathbb{R}^d$, i.e.,

$$\pi_\theta(a|s) = \mathbb{P}\left[A_t = a | S_t = s, \theta\right].$$

Now our policy becomes parameterized. Similar to loss functions in machine learning, we can define a scalar performance measure $J(\theta)$ of the policy with respect to the policy parameter θ. These methods seek to *maximize* performance, so their updates approximate gradient *ascent* in J:

$$\theta_{t+1} = \theta_t + \alpha \nabla \widehat{J(\theta_t)}.$$

Now the question becomes how we estimate the gradient of J over θ? First, we need to introduce the concept of *trajectory*: $\tau = (s_1, a_1, s_2, a_2, \ldots, s_H, a_H)$, in which the agent starts at state s_1, chooses action a_1, and gets to state a_2, and so on. The total discounted reward,

[38] Example adopted from https://www.freecodecamp.org/news/an-introduction-to-policy-gradients-with-cartpole-and-doom-495b5ef2207f/

which is the natural target function of RL problem, is then

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H} R(s_t, a_t)\right] = \sum_{\tau} \mathbb{P}_{\pi_\theta}(\tau) R(\tau). \tag{15}$$

Here $R(s_t, a_t)$ indicates that the reward in a period is a function of its current state s_t and action a_t. We then can randomly sample a number of trajectories based on current parameters θ, get an estimation of gradient using the samples and implement gradient ascent algorithms to maximize J.[39]

In our particular application of RL in portfolio construction, even though the high dimensionality can be handled by deep Q-learning, we have continuous action space and need to take the policy-based approach.

It should be clear that reinforcement learning is similar in spirit to multi-arm bandit, optimal experimentation, and dynamic programming. We next provide a simple application for illustration.

An application to cart-pole balancing. In this task, an inverted pendulum is mounted on a pivot point on a cart. The cart itself is restricted to linear movement, achieved by applying horizontal forces. Due to the system's inherent instability, continuous cart movement is needed to keep the pendulum upright. The observation consists of the cart position x, pole angle ω, the cart velocity $\dot{x}$, and the pole velocity $\dot{\omega}$. Therefore, the state is the four-element tuple, $s = (x, \omega, \dot{x}, \dot{\omega})$. The 1D action space consists of the horizontal force applied to the cart body. The reward function is given by

$$r(s, a) = 10 - (1 - \cos(\omega)) - 10^{-5}\|a\|_2^2,$$

which states, the angle deviated from upright should be close to 90° in order to get high reward. In addition, the force a should not be too large in order to maintain the stability of the system. The game terminates when $|x| > 2.4$ or $|\omega| > 0.2$, i.e., the pole irreversibly falls down. (This comes from the survey study of Duan et al., 2016, `https://arxiv.org/pdf/1604.06778.pdf`, for complete physical/environmental parameters, see `https://github.com/rll/rllab`).

We can use (artificial) neural networks (NN) to generate the i-th action a_i given state

[39] For the exact way to compute policy gradient, please refer to Sutton and Barto, Ch 13. Or for a simpler version, see this excellent blog post.

variables, where the parameters θ are the weights in neural network.

$$\pi_\theta(a_i|s) = \begin{cases} 1 & \text{if } a_i = f_\theta(s), \\ 0 & \text{else,} \end{cases}$$

where $f_\theta(\cdot)$ is the artificial neural network with weights θ. In Figure A.1, we have input $\boldsymbol{s} = (s_1, s_2, s_3, s_4)$. To transform the input $\boldsymbol{x}$ to a, the first step involves linear transformation:

$$u_1 = \theta_{10} + \theta_{11}s_1 + \theta_{12}s_2 + \theta_{13}s_3 + \theta_{14}s_4,$$
$$u_2 = \theta_{20} + \theta_{21}s_1 + \theta_{22}s_2 + \theta_{23}s_3 + \theta_{24}s_4,$$

with all θ_{ij} as model parameters just like β in linear models. After that, we apply a non-linear function called activation function $g(\cdot)$ on (u_1, u_2).[40] We get:

$$y_1 = g(u_1), \qquad y_2 = g(u_2).$$

And we get the output $\boldsymbol{y} = (y_1, y_2)$. Then, the final action of $f_\theta(s)$ could be given as

$$a = g\left(\theta_{y0} + \theta_{y1}y_1 + \theta_{y2}y_2\right).$$

Note that the parameters θ_{ij} are the ones we use policy derivative algorithms to optimize in order to generate the optimal horizontal force and get the max reward.

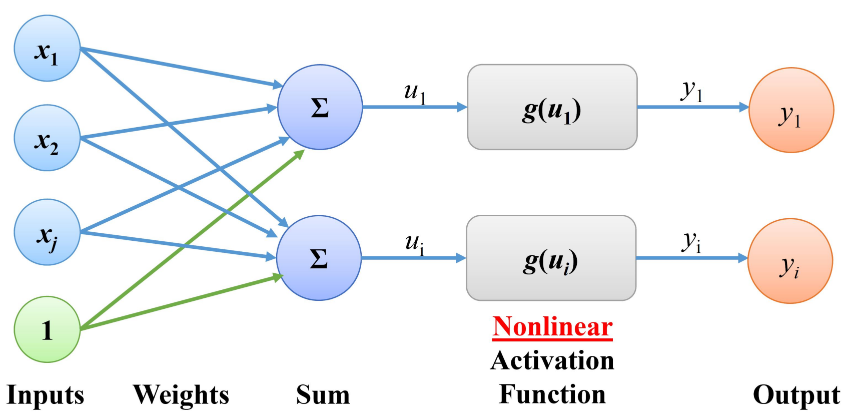

Figure A.1: A Single-Layer Artificial Neural Network

In general, after specifying the environment, reward, policy, and parameters, we can use policy derivative to approximate the optimal policy. The agent seeks θ that maximizes the reward function using gradient ascent on sampled state-action trajectories.

[40]Common activation functions include sigmoid, rectified linear unit(ReLU), hyperbolic tangent, etc.

Appendix B. Input Feature Construction

This section details the construction of the 51 variables we use as input features. We obtain the raw data from three WRDS databases: CRSP, CRSP Compustat Merged, and Financial Ratio Firm Level. Characteristics with highlights can be obtained from Financial Ratio Firm Level database.

A2ME: We define assets-to-market cap as total assets over market capitalization.

$$A2ME = \frac{AT}{(SHROUT * PRC)} \tag{16}$$

OA: We define operating accruals as change in non-cash working capital minus depreciation scaled by lagged total asset.

$$OA = \frac{\Delta\left((ACT + CHE - LCT - DLC - TXP) - DP\right)}{AT_{t-1}} \tag{17}$$

AOA: We define AOA as absolute value of operation accruals

AT: Total asset

BEME: Ratio of book value of equity to market equity.

Beta_daily: Beta_daily is the sum of the regression coefficients of daily excess returns on the market excess return and one lag of the market excess return.

C: Ratio of cash and short-term investments to total assets.

$$C = CHE/AT \tag{18}$$

C2D: Cash flow to price is the ratio of income and extraordinary items and depreciation and amortization to total liabilities.

$$C2D = (IB + DP)/LT \tag{19}$$

CTO: We define capital turnover as ratio of net sales to lagged total assets.

$$CTO = SALE/AT_{t-1} \tag{20}$$

Dept2P: Debt to price is the ratio of long-term debt and debt in current liability to the market capitalization.

$$Dept2P = \frac{(DLTT + DLC)}{(SHROUT * PRC)} \tag{21}$$

Δceq: The percentage change in the book value of equity.

$$\Delta ceq = (CEQ_t - CEQ_{t-1})/CEQ_{t-1} \tag{22}$$

Δ(ΔGm - ΔSales:) The difference in the percentage change in gross margin and the percentage change in sales.

$$\Delta(\Delta Gm - \Delta Sales) = \frac{SALE_t - COGS_t}{SALE_{t-1} - COGS_{t-1}} - \frac{SALE_t}{SALE_{t-1}} \tag{23}$$

ΔSo: Log change in the split adjusted shares outstanding.

$$\Delta So = \log\left(CSHO_t * AJEX_t\right) - \log\left(CSHO_{t-1} * AJEX_{t-1}\right) \tag{24}$$

Δshrout: Percentage change in shares outstanding.

$$\Delta shrout = \left(SHROUT_t - SHROUT_{t-1}\right)/SHROUT_{t-1} \tag{25}$$

ΔPI2A: The change in property, plants, and equipment over lagged total assets.

$$\Delta P12A = \frac{\Delta\left(PPENT + INVT\right)}{AT_{t-1}} \tag{26}$$

E2P: We define earnings to price as the ratio of income before extraordinary items to the market capitalization.

$$E2P = IB/\left(SHROUT * PRC\right) \tag{27}$$

EPS: We define earnings per share as the ratio of income before extraordinary items to shares outstanding.

$$E2P = IB/SHROUT \tag{28}$$

Free CF: Cash flow to book value of equity.

$$FreeCF = \frac{NI + DP + WCAPCH + CAPX}{BE} \tag{29}$$

Idol vol: Idiosyncratic volatility is the standard deviation of the residuals from a regress of excess returns on the Fama and French three-factor model.

Investment: We define investment as the percentage year-on-year growth rate in total assets.

$$Investment = \left(AT_t - AT_{t-1}\right)/AT_{t-1} \tag{30}$$

IPM: Pre-tax profit margin, $EBT/Revenue$.

IVC: We define IVC as change in inventories over the average total assets of t and $t-1$.

$$IVC = \frac{2 * \left(INVT_t - INVT_{t-1}\right)}{AT_t + AT_{t-1}} \tag{31}$$

Lev: Leverage is the ratio of long-term debt and debt in current liabilities to the sum of long-term debt, debt in current liabilities, and stockholder's equity.

$$Lev = \left(DLTT + DLC\right)/SEQ \tag{32}$$

LDP: We define the dividend-price ratio as annual dividends over price.

$$LDP = \frac{\sum\left(RET - RETX\right)}{PRC} \tag{33}$$

ME: Size is the market capitalization.

Turnover: Turnover is volume over shares outstanding.

$$Turnover = VOL/SHROUT \tag{34}$$

NOA: Net operating assets are the difference between operating assets minus operating liabilities scaled by lagged total assets.

$$NOA = \frac{left\left(AT - CHE - IVAO\right) - \left(AT - DLC - DLTT - MIB - PSTK - CEQ\right)}{AT_{t-1}} \tag{35}$$

NOP: Net payout ratio is common dividends plus purchase of common and preferred stock minus the sale of common and preferred stock over the market capitalization.

$$NOP = \frac{(DVC + PRSTKC - SSTK)}{ME} \tag{36}$$

O2P: Payout ratio is common dividends plus purchase of common and preferred stock minus the change in value of the net number of preferred stocks outstanding over the market capitalization.

$$O2P = \frac{DVC + PRSTKC - (PSTKRV_t - PSTKRV_{t-1})}{ME} \tag{37}$$

OL: Operating leverage is the sum of cost of goods sold and selling, general, and administrative expenses over total assets.

$$OL = (COGS + XSGA)/AT \tag{38}$$

PCM: The price-to-cost margin is the difference between net sales and costs of goods sold divided by net sales.

$$PCM = (SALE - COGS)/SALE \tag{39}$$

PM: The profit margin (operating income/sales)

Prof: We define profitability as gross profitability divided by the book value of equity.

$$Porf = GP/BE \tag{40}$$

Q: Tobin's Q is total assets, the market value of equity minus cash and short-term investments, minus deferred taxes scaled by total assets.

$$Q = \frac{(AT + ME/1000 - CEQ - TXDB)}{AT} \tag{41}$$

Ret: Return in the month.

Ret_max: Maximum daily return in the month.

RNA: The return on net operating assets.

ROA: Return-on-assets.

ROC: ROC is the ratio of market value of equity plus long-term debt minus total assets to cash and short-term investments.

$$ROC = \frac{(DLTT + ME/1000 - AT)}{CHE} \tag{42}$$

ROE: Return-on-equity.

ROIC: Return on invested capital.

S2C: Sales-to-cash is the ratio of net sales to cash and short-term investments.

$$S2C = SALE/CHE \tag{43}$$

Sale_g: Sales growth is the percentage growth annual rate in annual sales.

$$Sale_g = SALE_t/SALE_{t-1} - 1 \tag{44}$$

SAT: We define asset turnover as the ratio of sales to total assets.

$$SAT = SALE/AT \tag{45}$$

S2P: Sale-to-price is the ratio of net sales to the market capitalization.

SGA2S: SG&A to sales is the ratio of selling, general and administrative expenses to net sales.

$$SAT = XGSA/SALE \tag{46}$$

Spread: The bid-ask spread is the average daily bid-ask spread in the month.

Std_turnover: Standard deviation of daily turnover in the month.

Std_vol: Standard deviation of daily trading volume in the month.

Tan: Tangibility.

$$Tan = \frac{0.715 * RECT + 0.547 * INVT + 0.535 * PPENT + CHE}{AT} \tag{47}$$

Total_vol: Standard deviation of daily return in the month.

Appendix C. AlphaPortfolio — An LSTM Edition

For an LSTM-implementation of our AlphaPortfolio, we replace TE with Bi-directional LSTM with attention mechanism. Table C.1 reports the results. In terms of out-of-sample metrics such as the Sharpe ratio, the Bi-LSTM-CAAN model outperforms our original AP.

Table C.1: Out-of-Sample Performance of Bi-LSTM-CAAN-based AP

This table presents the out-of-sample performance for Bi-LSTM-CAAN-based AP. For each month in out-of-sample periods (1990-2016), AP constructs a hedge portfolio which goes long the 10% of stocks with the highest winner scores and shorts 10% of stocks with the lowest winner scores. The investment proportions are calculated according to Section 3.3

	(1)	(2)	(3)
Firms	All	size > q_{10}	size > q_{20}
Return	16.90	15.50	15.10
Std	7.70	5.10	5.10
Sharpe	2.20	3.04	2.96
Skewness	1.63	0.86	1.06
Kurtosis	6.85	2.43	4.88
MDD	3.40	1.40	1.70

However, our economic distillation reveals that LSTM does not have stable utilization of input features or economic interpretability using textual factors. This is consistent with that LSTM deals with vanishing and exploding gradients only in the training sample, and with that AI may face adversarial attacks and instability of performance (Heaven, 2019).

Specifically, from Table C.2, Bi-LSTM-CAAN-based AP tend to select characteristics of the first/last position ("_0" and "_11") in the input sequence, a pattern robust when we change the number of months to generate lagged inputs. This is indicative of exploding gradient issues, which means the trained model go to some extremes or gradient-based interpretation methods are not that suitable for such RNN like models in our case.

While gradient exploding problem can be solved by gradient clip during training (back-propagation), when we have a well-trained model and use it to test out-of-sample data, researchers typically do not know whether in the test sample there is a problem of gradient explosion. In other words, the detection of gradient explosions in computer science which focuses on the training stage and would not flag such technical issues of a model from test samples. Our economic distillation therefore helps to detect modeling issues out-of-sample that traditional CS approach neglects.

Table C.2: Fama-Macbetch T-test Values of Selected Terms (Bi-LSTM based AP)

This table presents the results of using Fama-Macbetch method to interpret Algorithm 1. Polynomial degree is set as one and for all terms with suffixed like ”_*no*”, *no* indicates the sequence number of input features. *i.e.,* *pe*_7 denotes P/E ratio at the time of five $(12-7=5)$months ags. For details of each characteristic, please refer to Appendix B. q indicates the size percentile of *NYSE* firms. In this table, we present the top 50 significant terms.

All		size $> q_{10}$		size $> q_{20}$	
pe_0	85.04	tan_11	-83.87	tan_11	-84.56
ivc_11	-82.17	ivc_11	-78.49	ivc_11	-81.81
C_11	78.34	pe_0	68.77	pe_0	68.10
Q_11	-70.57	C_11	68.14	C_11	62.64
tan_11	-65.63	Q_11	-52.07	Q_11	-50.44
ivc_0	-58.49	Idol_vol_0	50.99	e2p_11	50.42
Idol_vol_0	51.74	ivc_0	-50.08	Idol_vol_0	48.96
Turnover_0	-44.84	Turnover_0	-48.06	ret_11	-46.00
delta_so_11	-43.99	e2p_11	45.04	Turnover_0	-45.15
Idol_vol_11	38.00	ret_11	-44.48	ivc_0	-44.60
Turnover_11	-34.92	delta_so_11	-39.49	delta_so_11	-38.08
Ret_max_11	-33.85	Beta_daily_0	-37.94	Turnover_11	-37.19
s2p_11	33.25	Turnover_11	-36.8	Beta_daily_0	-35.24
Beta_daily_0	-31.15	beme_11	31.72	beme_11	28.06
delta_shrout_11	26.15	Idol_vol_11	27.07	s2p_11	26.95
s2p_0	23.97	s2p_11	25.91	investment_11	-26.54
ret_11	-23.82	cto_0	-24.41	Idol_vol_11	24.57
beme_11	23.29	Std_volume_0	-20.67	oa_11	24.23
oa_11	23.03	investment_11	-20.65	cto_0	-22.75
roa_11	-22.62	Beta_daily_11	20.31	Beta_daily_11	22.07
roa_0	-22.36	sat_11	20.29	Std_volume_0	-20.98
investment_11	-19.87	oa_11	20.17	sat_11	20.78
Std_volume_0	-19.86	s2p_0	20.03	s2p_0	20.68
pe_11	-18.91	delta_shrout_11	19.97	delta_shrout_11	19.08
sat_11	18.49	roa_11	-19.52	noa_11	19.05
at_11	17.02	pe_11	-17.76	shrout_11	17.93
cto_0	-16.31	shrout_11	17.58	roic_0	17.45
c2d_11	16.28	Ret_max_11	-16.09	me_11	16.33
shrout_11	15.09	sat_0	16.05	nop_0	-15.85
beme_0	13.93	me_11	15.49	roa_11	-14.59
investment_0	-13.84	nop_0	-15.46	pe_11	-14.16
sga2s_11	13.61	c2d_11	14.39	c2d_11	13.81
roic_11	13.42	roic_0	13.9	sat_0	13.39
me_11	13.20	noa_11	12.89	Ret_max_11	-13.31
aoa_11	-12.21	delta_pi2a_11	12.86	delta_so_0	-12.92
delta_pi2a_0	12.09	sga2s_11	12.45	sga2s_11	12.38
Beta_daily_11	12.07	delta_so_0	-11.76	vol_11	-11.13
roic_0	12.07	sale_g_0	11.62	a2me_11	-10.69
shrout_0	11.07	roic_11	11.23	sale_g_0	10.66
delta_pi2a_11	10.89	at_11	11.12	roic_11	10.62
sat_0	10.52	a2me_11	-10.35	investment_0	-10.38
e2p_11	10.21	beme_0	10.29	at_11	9.91
delta_so_0	-9.56	C_0	10.14	nop_11	9.41
C_0	9.03	nop_11	10.02	shrout_0	9.29
sale_g_0	8.93	vol_11	-9.83	aoa_11	-9.03
sale_g_11	8.53	shrout_0	8.82	beme_0	8.65
std_0	-8.23	sale_g_11	8.09	C_0	8.40
a2me_11	-8.07	investment_0	-7.53	std_11	-6.95
Spread_11	-6.79	std_11	-6.9	free_cf_11	6.84