NBER WORKING PAPER SERIES

ALPHAPORTFOLIO: GOAL-ORIENTED INVESTMENT MANAGEMENT THROUGH DEEP REINFORCEMENT LEARNING

Lin William Cong
Ke Tang
Jingyuan Wang

Working Paper 35195
http://www.nber.org/papers/w35195

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
May 2026

We are grateful to Dimitris Papanikolaou (the Editor), two anonymous referees, David Avramov, Agostino Capponi, Bing Han, Serhiy Kozak, Andreas Neuhierl, Tom Sargent, and Guofu Zhou for detailed feedback and to Si Cheng for kindly sharing the data on market illiquidity. We also thank Ludwig Chincarini, John Cochrane, Gianluca De Nard, Marcos de Prado, Gavin Feng, Itay Goldstein, Jillian Grennan, Cam Harvey, Andrew Karolyi, Bryan Kelly, Sophia Zhengzi Li, Alejandro Lopez-Lira, Stefan Nagel, Markus Pelger, Amin Shams, Jinfei Sheng, Zhan Shi, Stathis Tompaidis, Neng Wang, Michael Weber, Dacheng Xiu, Mao Ye, Lu Zhang, Luofeng Zhou, and conference and seminar reviewers and participants at the 38th ABFC International Forum, Annual Conference in Digital Economics, Cambridge Algorithmic Trading Society Quant Conference, 2024 CCER Summer Institute, 2nd Conference on FinTech, Innovation and Development (Hangzhou), 2021 Academic Research Colloquium, 2022 AFA Annual Meeting, Australasian Banking and Finance conference, 5th Big Data Econometrics Theory and Applications Conference, Blackrock FMG Webinar, Bloomberg CTO Data Science Speaker Series, Boston Fed, BU, 10th Annual Workshop of Business Financing and Banking Research Group(USyd), Baylor University, CKGSB, 2021 CICF (Shanghai), 15th China International Conference on Insurance and Risk Management, 3rd China International Forum on Finance and Policy, The Chinese Economist Society Annual Deans' Forum, Cornell SC Johnson, Cornell Tech, Center for Research in Economics and Statistics and Ecole Polytechnique, 3rd Joint Conference of Statistics and Data Science in China, CUHK, CUHK (Shenzhen), Econometric Society World Congress (Milan), Duke Fuqua, 2024 EFMA Annual Meeting, Financial Markets and Corporate

Governance Conference (FMCG), 2025 FMA Annual Meeting, Fudan, Georgetown University Global Virtual Seminar Series on FinTech "Machine Learning Day," Global Digital Economy Summit for Small and Medium Enterprises, Global Quantitative and Macro Investment Conference (Wolfe QES), Goethe University Frankfurt, Harvest Fund 25th Anniversary Ceremony, HKU-SCF FinTech Academy Distinguished Lecture, HK Baptist, Lingnan (HK), HKMA & Academy of Finance, IIF International Research Conference & Award Summit, 2021 I NFORMS Annual Meeting, INQUIRE UK/Europe Webinar, KAIST Digital Finance Conference, London Quant Group (LQG) webinar, Luohan Academy Webinar, 13th International Risk Management Conference (IRMC), Machine Lawyering Conference: Human Sovereignty and Machine Efficiency in the Law, 1st Annual MARC Conference at the University of Toronto, 2023 Mid-South DATA Conference (Memphis), Midwest Finance Association Annual Meeting, NUS DAO-ISEM-IORA, National Excellent Students Summer Camp at Fudan, NTU (Dean's Distinguished Speaker Series), 1st Annual MARC Conference at UoT, NUS, 2021 NBER Conference on Big Data and Securities Markets, 2024 NBER Summer Institute (Asset Pricing), P HBS, PKU-NSD, Peking University School of Mathematical Sciences, PolyU Digital Finance Symposium, QJF 2024 Forum, RSiFEB, SAIF, 6th Shanghai Financial Forefront Symposium, Shenzhen Loop Area Institute, Singapore Initiative on Digital Economics Annual Workshop, Southwestern Finance Association Annual Meeting, Annual Meeting of the Swiss Society for Financial Market Research (SGF), SFS Cavalcade (Seoul), Stanford MS&E, Stanford SITE "Macro Finance and Computation," Toulouse School of Economics, University of Turin Collegio Carlo Alberto, UF Research Conference on Machine Learning in Finance, University of Virginia McIntire School of Commerce and Darden School of Business, World Finance Conference, 4th Workshop on Big Data Econometric Theory and Application, Xi'an Jiaotong University, 2023 XJTL U AI and Big Data in Accounting and Finance Conference, XueShuo/DEFT Summer Institute in Digital Finance, Yangtze River Delta International Forum "Corporate Finance and Financial Markets," and Zhongnan University of Economics and Law for their comments. Jiahao Wen and Yang Zhang provided exceptional and integral technical contributions assisting the project; Pietro Bini, Fujie Wang, Hanyao Zhang, and Guanyu Zhou also provided excellent research assistance. This research was funded in part by the Ewing Marion Kauffman Foundation, INQ UIRE UK, ICPM Research Award, and Cornell Center for Social Sciences. The contents of this article are solely the responsibility of the authors. The paper was previously titled "Goal-Orient ed Portfolio Management Through Transformer-Based Reinforcement Learning" and includes partial results from the earlier working paper under the title "Direct Construction Through Deep Reinforcement Learning and Interpretable AI" (jointly with Yang Zhang). Send correspondence to Cong. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

NBER working papers are circulated for discussion and comment purposes. They have not been peer-reviewed or been subject to the review by the NBER Board of Directors that accompanies official NBER publications.

AlphaPortfolio: Goal-Oriented Investment Management Through Deep Reinforcement Learning
Lin William Cong, Ke Tang, and Jingyuan Wang
NBER Working Paper No. 35195
May 2026
JEL No. C14, C58, G11, G12

ABSTRACT

We adapt attention-based neural networks and reinforcement learning to direct portfolio construction, allowing broader portfolio-management objectives (including non-time-additively separable ones) and in a data-driven way, searching over a much richer policy/strategy space than low-dimensional parametric rules or human-specified strategies. As arguably the first non-text-based, "large" GenAI model in Finance, AlphaPortfolio accommodates long- and short-range path dependence in firm and market states (e.g., using Transformer encoder), cross-asset information, flexible (path-dependent) objectives (incl. Sharpe ratio, which is non-additively separable across periods) for end-to-end (rather than step-by-step) optimizations. In U.S. equities, AlphaPortfolio yields superior out-of-sample performance (e.g., Sharpe ratio above two and risk-adjusted alpha over 13% with monthly rebalancing) robust under various market conditions and economic restrictions (e.g., exclusion of small/illiquid stocks) and over time. The gains come from the direct construction, effective sequence modeling, and cross-asset attention network. We further demonstrate AlphaPortfolio's flexibility to incorporate transaction costs, state interactions, and alternative objectives, before developing a polynomial-feature-sensitivity analysis to uncover key drivers of performance, including their rotation and nonlinearity.

Lin William Cong
Nanyang Technological University
and NBER
will.cong@ntu.edu.sg

Jingyuan Wang
Beihang University
jywang@buaa.edu.cn

Ke Tang
Tsinghua University
ketang@tsinghua.edu.cn

1 Introduction

Portfolio management asks how to map available financial data into portfolio weights or trading strategies that are optimal for an investor's objective. Classical portfolio theories and extant management practices typically tackle this problem indirectly: first estimate moments, expected returns, or pricing kernels, and then solve a separate portfolio optimization problem.[1] They have serious drawbacks due to large estimation errors in the first step, not to mention that the goals in the two steps are not necessarily aligned.[2] A related literature on direct portfolio construction moves closer to the allocation problem itself, arguing that portfolio construction should not be guided by intermediate statistical criteria different from the economic objective. But it usually does so by restricting either the objective or the policy space, failing to handle multi-period or path dependent portfolio goals, or to generate effective mappings from high-dimensional inputs to effective portfolio decisions dependent on both lagged states and cross-asset information.

We develop AlphaPortfolio (AP for short) as a new solution to the general investment management problem, innovating upon recent advancements in AI.[3] Taking the parametric portfolio policy (PPP) framework of Brandt, Santa-Clara, and Valkanov (2009) as the benchmark, we generalize direct portfolio construction along two dimensions. First, we allow broader portfolio-management objectives, including multi-period and path-dependent criteria such as Sharpe ratios computed over rebalancing windows, cumulative return with survival constraints, delegated-compensation payoffs, and cost-adjusted objectives. Second, we replace low-dimensional parametric portfolio rules with a much richer policy class built from modern sequence modeling, cross-asset attention, and reinforcement learning (RL). In particular, we adapt Transformer (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez,

[1] Mean-variance optimization based on the investor's preference is one example (Markowitz, 1952). Practitioners and researchers also routinely construct portfolios by characteristic-based sorting. Risk-parity as an objective is an attempt to shift focus away from the first moment.

[2] In a prediction-first workflow, the statistical model is trained to minimize forecast errors for returns in a purely statistical sense. However, investors do not value forecast errors symmetrically across all securities. Which errors matter most depends on the portfolio problem, including risk preferences, trading constraints, implementation frictions, and the management objective itself. As a result, improvements in global predictive accuracy need not translate into better portfolio outcomes. This gap becomes especially important once one moves beyond frictionless mean–variance settings to economically relevant environments with transaction costs, position constraints, dynamic budgets, or delegated-management considerations. For a more elaborate discussion of the potential advantages of direct optimization, see Appendix A1.

[3] The "Alpha" in "AlphaPortfolio" comes from the naming convention in the AI field as seen in the cases of AlphaGo, AlphaZero, and AlphaFold; it does not refer to the concept of "excess return" in finance and therefore would not require the resulting portfolio to have zero exposure to (risk) factors.

Kaiser, and Polosukhin, 2017) and deep RL — a learning paradigm proven to be effective in AI developments including computer vision, self-driving, and machine translation — for goal-oriented portfolio management.[4] The resulting end-to-end framework optimizes the investor's objective directly rather than separate step-by-step targets.[5]

First circulated in 2019, AlphaPortfolio provides the earliest general framework to combine sequence modeling, cross-asset modeling, and RL for direct portfolio construction. The contribution is layered: we formulate the general problem, introduce a modular policy architecture, and study a Transformer-based implementation with extensive finance-grounded evidence. Specifically, utilizing the strength of deep reinforcement learning (RL, e.g., Mnih et al., 2015; Silver et al., 2017) and the self-attention architecture of modern AI, we build AlphaPortfolio as a sequence of nested extensions of the benchmark PPP framework. We first replace the restricted linear PPP rule with a neural-network policy to enlarge the policy space. We then add a sequence representation extraction module (SREM) that learns both short- and long-range path-dependence from lagged firm and market states rather than only from contemporaneous characteristics. Next, we introduce a relation representation extraction module (RREM) that captures dependence across assets. Both the SREM and RREM are implemented using the Transformer self-attention mechanism, jointly forming a Panel Transformer architecture tailored to financial data. Finally, through interacting with the environment, RL algorithms iteratively map situations to actions to maximize a numerical reward. RL therefore provides the goal-directed heuristic search algorithm in a large policy space, accommodating unlabeled data, potential interactions between trading and system states, and flexible economic objectives.

Our key insight is that given the complexity of financial markets and asset price dynamics, searching using trial-and-error through a large modeling space to directly maximize arbitrary portfolio management objectives can be more effective and useful than attempt-

[4]In finance, the role of the Transformer differs from that in language or video applications. We do not assume that raw stock returns themselves exhibit strong deterministic sequence structure. Instead, the Transformer is used to summarize the history of lagged firm characteristics and market states into a representation that is useful for portfolio construction. Even if unconditional return autocorrelation is weak, the conditional relevance of lagged characteristics, fundamentals, trading signals, and market conditions for portfolio choice may still vary over time and across assets.

[5]AP demonstrates how economically motivated and tailored applications of AI can answer key questions in finance research and practice. One salient application beyond the now popular language models involves investment management because mass customization in investment advising has been extremely limited due to the lack of an effective approach to managing portfolios with flexible objectives. The model is generative in that it jointly considers the objective and the generation of the strategies/portfolios leading to a particular outcome of the objective. The RL approach basically generates repeatedly portfolios and uses the corresponding performance to finetune the policy.

ing to estimate exogenously specified moments of the return distributions or to price all assets accurately regardless of their relevance for constructing the desirable portfolio. Enlarging the modeling space allows the data to flexibly guide our search for a solution that researchers may not have come up with before. One particular underappreciated advantage of the enlarged policy space is the model's ability to adaptively learn which lags of which characteristics are informative for portfolio construction, rather than restricting attention to the most recent month as is standard in characteristic-based studies. The distillation results in Section 5 confirm that the same variables recur at multiple lag positions with different signs, indicating that the model discovers a richer temporal structure than any fixed lag rule could capture. This finding has implications beyond our specific model: it suggests that the standard practice of using only contemporaneous characteristics in portfolio sorts and cross-sectional regressions may discard economically relevant information about the time profile of firm states.

It is worth emphasizing that AlphaPortfolio's contribution extends beyond enlarging the policy space — it also enlarges the *objective space*. Conventional two-step approaches and even the PPP benchmark are most naturally formulated for one-period or additively separable objectives, such as expected utility of next-period wealth. Objectives like the Sharpe ratio computed over a multi-month rebalancing window are not additively separable across periods: they depend on the joint distribution of the entire return path, not on period-by-period rewards. Such objectives cannot be directly optimized within a predict-then-optimize pipeline, because there is no "correct" target Sharpe ratio to supply as a label in supervised learning. The end-to-end RL formulation handles these objectives naturally, because the trajectory-level reward in (3) can be any measurable functional of the realized path. We choose the Sharpe ratio as our baseline objective not only because it facilitates comparison across models in the literature, but also because it exemplifies the class of economically important objectives that become feasible only under direct, end-to-end construction.

Empirically, AlphaPortfolio consistently and significantly outperforms conventional approaches. Beyond maximizing performance metrics (e.g., Sharpe ratio) or beating existing models of portfolio management in a horse race, our findings support that portfolio construction should be guided by the economic objective and RL has advantages over the hitherto widely applied supervised learning, especially in accommodating flexible economic objectives. AlphaPortfolio does not merely merge forecasting with a downstream optimizer. It learns portfolio policies directly for general multi-period management problems, allowing the

relevant information in lagged firm states, market conditions, and cross-asset dependence to be extracted in a way that is disciplined by the ultimate investment objective.

Specifically, in our U.S. equity application, we use a 12-month window of lagged firm characteristics and market signals as inputs in the baseline specification. AlphaPortfolio is then trained to maximize the subsequent 12-month out-of-sample Sharpe ratio under monthly rebalancing.[6] The resulting strategy achieves Sharpe ratios above two in the main 1990–2016 test sample and in the large-stock subsamples excluding the bottom 10% or 20% of stocks by market capitalization, and remains strong in the additional 2017–2024 holdout period. Annualized factor-adjusted alpha also consistently exceeds 13.5%, while turnover and maximum drawdown compare favorably with both conventional and recent machine-learning-based benchmarks.[7]

The gains are economically informative rather than purely mechanical. Relative to a two-step implementation that uses the same Panel Transformer but then forms portfolios conventionally, RL-based AlphaPortfolio more than doubles out-of-sample Sharpe ratios. Comparisons against PPP, a neural-network direct-construction benchmark, AlphaPortfolio without cross-asset modeling based on deep learning, and indirect two-step portfolios show that the gains arise from direct construction, sequence modeling of lagged states, and cross-asset relationship modeling. In particular, the cross-asset modeling contributes materially on top of the sequence model, both utilizing the attention architecture (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, and Polosukhin, 2017), while the full model substantially outperforms indirect constructions based on the same desirability scores.

The findings are robust under various economic restrictions that hamper performances of many other ML strategies (Avramov, Cheng, and Metzker, 2019). The outperformance is not driven by short positions alone, ad hoc weighting, high-frequency trading, or particular industry sectors, and it persists under alternative turnover measures, exclusions of unrated and downgraded firms, and subsamples defined by market sentiment, volatility, and liquid-

[6]Unlike supervised learning in which the learner is told what the correct actions are in the training, RL discovers the best actions for some delayed rewards through trial-and-error search and utilizing feedback from the environment (Sutton and Barto, 2018, p.1). In the context of average OOS monthly Sharpe ratio, the reward is "delayed" in that it is computed over a multi-month window: A portfolio construction in one month could affect the future market environments and thus future portfolio construction once we take into consideration elements such as a manager's portfolio size and transaction costs.

[7]For example, Gu, Kelly, and Xiu (2020) report an out-of-sample Sharpe ratio of 1.35 with value-weighted portfolios, while Freyberger, Neuhierl, and Weber (2020) report 1.6 after excluding the smallest 10% stocks; the maximum drawdowns of common factors such as Fama-French-Carhart four factors over the same time period lie between 40%-60% whereas those of AP lie between 2%-10%.

ity. More broadly, AlphaPortfolio complements dynamic portfolio-choice frameworks such as Gârleanu and Pedersen (2013) by allowing transaction costs and related frictions to be incorporated directly into training rather than treated only as ex post robustness checks. We provide multiple illustrations that our framework naturally extends to objectives such as fund survival, delegated-management compensation, constrained dynamic investment-management problems, and performs well in realistic institutional investment settings.

Beyond demonstrating AP's robustness, flexibility, and practical relevance, we also aim to better interpret the 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 novel polynomial sensitivity analysis "projects" a complex model onto a space of linear models. We find that some usual suspects such as Tobin's Q play dominant roles, and so do inventory changes, changes in shares outstanding, and the like. In addition, we find higher-order terms affect AP's behavior but interaction effects do not (which could still be important for estimating assets' returns or a pricing kernel). Finally, we observe short-term reversals and identify important features that are dominant throughout and others that rotate in and out. Overall, economic distillations not only provide initial interpretations of complex models so that we can avoid pitfalls of AI applications when the market environment or policy changes, but also provide a sanity check on coding errors and model fragility.

Literature. Our paper contributes to three strands of literature, the last one being interdisciplinary and with initial studies mostly developed in the computer science field.

First, we contribute to studies on portfolio theory, especially direct portfolio construction. Classical portfolio theory, beginning with Markowitz (1952), typically follows a two-step procedure: estimate return moments or predictive objects, and then solve for portfolio weights. In practice, this approach is fragile because expected returns and large covariance matrices are difficult to estimate reliably, which can generate unstable and extreme portfolio allocations (e.g., Merton, 1980; Green and Hollifield, 1992; Best and Grauer, 1991). A large literature mitigates these problems using Bayesian priors, shrinkage, robust allocation, and constrained optimization (e.g., Jorion, 1986; Pástor, 2000; Goldfarb and Iyengar, 2003; Jagannathan and Ma, 2003; DeMiguel, Garlappi, and Uppal, 2007), while practitioners often rely on Black–Litterman or risk-parity style implementations (Black and Litterman, 1990, 1992; Jurczenko, 2015). These approaches are valuable, but they remain tied to the estima-

tion of moments or return forecasts and are less suited to high-dimensional, nonlinear, and dynamic investment problems.

A line of work pioneered by Brandt (1999) moves closer to the allocation problem by parameterizing portfolio weights directly as functions of observable states, exploiting that the relationship between optimal weights and predictors can be less noisy and more stable than estimates of the full conditional return distribution. Subsequent work develops parametric, semi-parametric, and dynamic portfolio policies (e.g., Aït-Sahalia and Brandt, 2001; Brandt and Santa-Clara, 2006; Brandt, Santa-Clara, and Valkanov, 2009), but suffers from misspecifications. Meanwhile, nonparametric or locally parametric estimators from sample analogues of the FOCs or Euler equations face the curse of dimensionality and limited reliable implementation beyond two predictors (e.g., Brandt, 2010), unless one imposes additional restrictions (e.g., Powell, Stock, and Stoker, 1989; Aït-Sahalia and Brandt, 2001).

AlphaPortfolio takes the parametric portfolio policy (PPP) framework of Brandt, Santa-Clara, and Valkanov (2009) as the natural benchmark to extend this literature with AI/ML techniques. On the problem side, it accommodates general portfolio objectives, including path-dependent and delegated-management criteria. On the solution side, it replaces low-dimensional parametric rules with a much broader policy class built from sequence modeling, cross-asset modeling, and RL. Because the same framework can encode investor-specific objectives and implementation constraints, it is also relevant to goal-based wealth management and robo-advising (Das, Ostrova, Radhakrishnanb, and Srivastavb, 2018; Capponi, 2022; Alsabah, Capponi, Ruiz Lacedelli, and Stern, 2019).

Our paper thus leads this recent wave of studies on direct portfolio construction using machine learning, including integrated mean-variance approaches (Butler and Kwon, 2023; Wang, Gao, Harvey, Liu, and Tao, 2026), implementable-efficient-frontier methods (Jensen, Kelly, Malamud, and Pedersen, 2024), deep extensions of PPP (Simon, Weibels, and Zimmermann, 2025), and genetic programming search (Liu, Zhou, and Zhu, 2024). By contrast, AlphaPortfolio is not tied to mean-variance optimization, efficient-frontier construction, or static utility: it treats portfolio choice as a general portfolio-policy problem with trajectory-level objectives, while learning the policy directly from the investor's objective (i.e., end-to-end) rather than integrating a predictive regression with an optimizer. It explicitly models lagged state dependence through a sequence module, potential action-state interactions in the RL framework, and cross-asset linkages captured in an attention network.[8]

[8]Liu, Zhou, and Zhu (2024) is a related heuristic search in a large solution space for the optimization; Al-

Second, we contribute broadly to ML applications in asset pricing. A growing body of work uses ML to forecast returns, estimate pricing kernels, recover latent factors, or extract information from large sets of structured and unstructured data (e.g., Gu, Kelly, and Xiu, 2020; Freyberger, Neuhierl, and Weber, 2020; Gu, Kelly, and Xiu, 2019; Fan, Ke, Liao, and Neuhierl, 2022; Chen, Pelger, and Zhu, 2020; Kelly and Xiu, 2021; Kelly, Kuznetsov, Malamud, Xu, and Zhang, 2024). These papers show that flexible models can improve prediction, factor estimation, and pricing-error reduction. Our contribution is complementary as we adapt it to direct portfolio construction itself. In that sense, AlphaPortfolio is closer to a portfolio-policy model than to a return-forecasting or pricing-kernel model.

The initial introduction of AlphaPortfolio (Cong, Tang, Wang, and Zhang, 2019) started a literature importing modern AI architectures into asset pricing beyond textual analyses. As another application of non-text-based large-scale embedding, Gabaix, Koijen, Richmond, and Yogo (2023) develop asset and investor embeddings from portfolio holdings and study how those latent representations explain valuations, return comovement, and portfolio decisions. Kelly, Kuznetsov, Malamud, and Xu (2025) is a subsequent application of Transformer in Finance by implanting it in the stochastic discount factor to improve conditional asset pricing. Broadly, these papers show that modern AI tools can be useful in finance when they are embedded in a well-defined economic decision problem rather than treated as generic black-box prediction devices. We differ in the object of interest and outputs, and further show how large deep learning models can be economically interpreted through "distillation" procedures that identify the dominant variables, nonlinearities, and time-varying patterns underlying its portfolio decisions.

Finally, and at a broader methodological level, our paper relates to advancements in generative modeling and interpretable AI. As an architecture for goal-oriented learning in large model spaces, AlphaPortfolio is related to offline or batch RL that learn policies from fixed historical data rather than from active experimentation (e.g., Fujimoto, Meger, and Precup, 2019; Kidambi, Rajeswaran, Netrapalli, and Joachims, 2020; Fu, Kumar, Nachum, Tucker, and Levine, 2020). This distinction is particularly relevant in finance, where online exploration is costly or infeasible. More broadly, the paper shows how generative modeling can be useful when deployed as a search algorithm for optimizing economic objectives rather than as a generic supervised prediction tool, addressing the broader call for economically

phaPortfolio differs in its broader portfolio-management objectives and its use of sequence models, attention network, and RL rather than relying on a static direct map or a specialized optimization layer.

grounded ML in finance (Karolyi and Van Nieuwerburgh, 2020). Subsequent work develops a data-driven-robust-control approach to corporate decision-making using RL (Campello, Cong, and Zhou, 2022), tree-based goal-oriented search for generating test assets, pricing factors, and heterogeneity groups (Cong, Feng, He, and He, 2023; Cong, Feng, He, and Li, 2023; Cong, Feng, He, and Wang, 2024), which answer different economic questions.

Related computer-science conference papers study proof-of-concept RL-based trading, under specific trading rules and objectives, and evaluate performance mainly as a prediction-and-execution system. For example, Wang, Zhang, Tang, Wu, and Xiong (2019) constitutes a special case under the AlphaPortfolio framework, with a non-scalable implementation and a focus on Sharpe ratio, without accommodating path-dependent objectives, implementation frictions, and state interactions. The Panel-Transformer implementation enables explorations of much larger policy spaces, while the explicit benchmarking to financial models (e.g., PPP) and economic distillations offers economic interpretation.[9]

The remainder of the paper proceeds as follows. Section 2 describes the general portfolio management problem, and our AI-inspired solution framework cast as a sequence of nested extensions from the PPP benchmark; Section 3 applies the AlphaPortfolio model to U.S. equities and analyzes the source of performance improvements; Section 4 studies robustness, alternative objectives, action-state interactions, and practical implementations. Section 5 introduces economic distillations to interpret the model; Section 6 concludes. The Appendices contain the foundations of reinforcement learning, a description of variable construction, implementation details of AP using both Transformer and alternative SREM and RREM modules, details of economic distillations using polynomial sensitivity analysis, etc.

[9]As a new finance application to combine gradient-based sensitivity analysis with sparse surrogate modeling, this paper identifies the dominant variables, nonlinearities, and time-varying patterns that account for a substantial share of AlphaPortfolio's investment decisions. It therefore adds to the recent computer science literature on model compression, distillation, and explainable AI. Existing distillation work typically compresses a complex model for computational efficiency or deployment (e.g., Buciluǎ, Caruana, and Niculescu-Mizil, 2006; Hinton, Vinyals, and Dean, 2015), while the explainable-AI literature emphasizes either surrogate modeling or feature-attribution methods to interpret black-box predictions (e.g., Ribeiro, Singh, and Guestrin, 2016; Sundararajan, Taly, and Yan, 2017; Krause, Perer, and Ng, 2016; Wu, Hughes, Parbhoo, Zazzi, Roth, and Doshi-Velez, 2018; Guidotti, Monreale, Ruggieri, Turini, Giannotti, and Pedreschi, 2018). Our application uses distillation primarily as an economic interpretation device. Cong, Li, and Wang (2024) formalizes and generalizes our distillation-based interpretation approach.

2 AlphaPortfolio for General Portfolio Management

We develop AlphaPortfolio as a new solution framework to the long-standing portfolio construction problem: how to utilize the financial data available to optimize an arbitrarily given investment objective. Relative to the literature, we enlarge not only the scope of the question (going from one-period or tightly structured utility problems to maximize arbitrary objectives that can be path-dependent in complex manners) but also the scope of the solution space (from searching over a small parametric family of portfolio rules or low dimensional empirical strategies to effectively exploring a much larger NN-induced policy space in a data-driven manner guided by the economic objective).

2.1 The General Portfolio Management Problem

Suppose there are I_t investable assets at date t in a discrete-time economy. Let $\boldsymbol{r}_{t+1} = (r_{1,t+1}, \ldots, r_{I_t,t+1})^\top$ denote the vector of simple asset returns realized from date t to $t+1$. At the beginning of period t, the portfolio manager (a.k.a., investor) observes a state $\boldsymbol{Z}_t \in \mathcal{Z}$, which may include current wealth W_t, asset characteristics $\boldsymbol{X}_t = (\boldsymbol{x}_{1,t}, \ldots, \boldsymbol{x}_{I_t,t})$, market-wide variables, and any other state variables relevant to portfolio management, such as budgets, cumulative performance, survival indicators, funding conditions, or mandate variables. The investor then chooses an action $\boldsymbol{a}_t \in \mathcal{A}(\boldsymbol{Z}_t)$. In the simplest case, the action is just the portfolio-weight vector $\boldsymbol{a}_t = \boldsymbol{w}_t = (w_{1,t}, \ldots, w_{I_t,t})^\top$, $\boldsymbol{w}_t^\top \mathbf{1} = 1$, but more generally $\boldsymbol{a_t}$ may also include consumption or withdrawals.[10]

For concreteness, if $\boldsymbol{a_t} = (\boldsymbol{w}_t, c_t)$, where c_t is consumption at t, then wealth evolves:

$$W_{t+1} = (W_t - c_t)\big(1 + \boldsymbol{w}_t^\top \boldsymbol{r}_{t+1}\big). \tag{1}$$

The state transition may in general depend on the action through a transition kernel $P(\boldsymbol{Z}_{t+1} \mid \boldsymbol{Z}_t, \boldsymbol{a}_t)$. Thus, our general framework allows for action-state interactions.

In the baseline empirical implementation below, we adopt the common price-taking assumption with respect to asset-price dynamics and use historical data as the environment; richer reduced-form interactions are considered later in the paper. To be explicit about what this assumption entails for the state-transition structure: in the baseline implementation,

[10]For clarity, we typically use lowercase symbols such as a, b to denote scalars, bold lowercase symbols such as $\boldsymbol{a}$, $\boldsymbol{b}$ to denote vectors, bold uppercase symbols such as $\boldsymbol{A}$, $\boldsymbol{B}$ to denote matrices.

the state $\boldsymbol{Z}_t$ consists entirely of lagged asset characteristics and market signals read from historical data, and these evolve exogenously—that is, they are not affected by the portfolio manager's actions. The transition kernel $P(\boldsymbol{Z}_{t+1} \mid \boldsymbol{Z}_t, \boldsymbol{a}_t)$ in (2) therefore reduces to $P(\boldsymbol{Z}_{t+1} \mid \boldsymbol{Z}_t)$, and the only channel through which actions affect the trajectory objective is through the realized portfolio return $R^p_{t+1}(\boldsymbol{\theta}) = \boldsymbol{w}_t(\boldsymbol{\theta})^\top \boldsymbol{r}_{t+1}$.

The Markov-control language in Section 2.1 is intentionally broader than what the baseline implements: it defines the general class of problems that AlphaPortfolio is designed to address, of which the price-taking, exogenous-state baseline is a tractable and empirically standard special case. The richer action-dependent transitions become operative in the extensions of Section 4.2, where transaction costs reduce net returns as a function of trading activity, dynamic budgets evolve with cumulative portfolio performance, and fund survival depends on the history of actions. Even in those extensions, we do not model endogenous feedback from the manager's trades to market-wide asset prices—a limitation discussed in Section 4.3—but the action-state interaction is genuine at the fund or manager level.

Given a policy $\pi_{\boldsymbol{\theta}}$, indexed by parameters $\boldsymbol{\theta}$, a finite-horizon investment episode of length T generates a trajectory $\tau = (\boldsymbol{Z}_1, \boldsymbol{a}_1, \boldsymbol{Z}_2, \boldsymbol{a}_2, \ldots, \boldsymbol{Z}_T, \boldsymbol{a}_T, \boldsymbol{Z}_{T+1})$. The trajectory distribution induced by the initial condition, policy, and transition kernel is:

$$P_{\boldsymbol{\theta}}(\tau) = \mu_1(\boldsymbol{Z}_1) \prod_{t=1}^{T} \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{Z}_t)\, P(\boldsymbol{Z}_{t+1} \mid \boldsymbol{Z}_t, \boldsymbol{a}_t), \tag{2}$$

where $\mu_1(\cdot)$ is the distribution of the initial state. For a deterministic policy, $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t \mid \boldsymbol{Z}_t)$ is degenerate at $\boldsymbol{a}_t = \pi_{\boldsymbol{\theta}}(\boldsymbol{Z}_t)$.

The investor's objective is a trajectory-level functional $H(\tau)$ which may depend on the entire realized path of states, actions, wealth, returns, turnover, drawdowns, factor exposures, or other economically relevant variables. The general problem becomes:

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim P_{\boldsymbol{\theta}}}[H(\tau)] = \int H(\tau)\, \mathrm{d}P_{\boldsymbol{\theta}}(\tau), \qquad \boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}). \tag{3}$$

This formulation is intentionally broad. It nests trajectory-level objectives such as the Sharpe ratio over a rebalancing window, cumulative return subject to survival constraints, delegated-compensation payoffs, and factor-neutral or cost-adjusted portfolio criteria. These also include portfolio-management objectives that are cumbersome to handle in a small parametric framework or in traditional, two-step-supervised-learning-based approaches.

An important, additively separable specification the literature routinely considers is:

$$H(\tau) = \sum_{t=1}^{T} \beta^{t-1} u_t(\boldsymbol{Z}_t, \boldsymbol{a}_t, \boldsymbol{Z}_{t+1}) + \beta^T \Phi(\boldsymbol{Z}_{T+1}), \tag{4}$$

where $u_t(\cdot)$ denotes period reward, β the discount factor, and $\Phi(\cdot)$ a terminal payoff. Under the usual Markov structure, this yields the standard dynamic-programming recursion

$$V_t(\boldsymbol{z}) = \sup_{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{z})} \mathbb{E}[u_t(\boldsymbol{z}, \boldsymbol{a}, \boldsymbol{Z}_{t+1}) + \beta V_{t+1}(\boldsymbol{Z}_{t+1}) \mid \boldsymbol{Z}_t = \boldsymbol{z},\ \boldsymbol{a}_t = \boldsymbol{a}]\,. \tag{5}$$

Here, additive rewards and Bellman recursion are nested special cases of the more general trajectory-level problem in (3). When $H(\tau)$ is not additively separable, Bellman recursion does not apply directly unless the state is augmented appropriately. AlphaPortfolio is designed to handle both conventional and such more general path-level objectives.

2.2 Parametric Portfolio Policies as Benchmark

Among direct portfolio optimization approaches, Parametric Portfolio Policies (PPP, Brandt, Santa-Clara, and Valkanov, 2009) provides a natural benchmark that specializes the general problem above along two dimensions. First, PPP is typically formulated for one-period or recursively structured objectives, typically with exogenous state dynamics and without rich path-dependent constraints. In other words, PPP gains tractability by working in a narrower subset of the general portfolio management objectives in (3). Second, instead of searching over a broad class of state-dependent portfolio policies, PPP assumes that portfolio weights are deterministic functions of observable predictors, i.e., $\boldsymbol{w}_t = f_{\boldsymbol{\theta}}(\boldsymbol{X}_t)$. In the benchmark linear PPP specification, $w_{i,t} = \bar{w}_{i,t} + \frac{1}{I_t}\boldsymbol{\theta}^\top \hat{\boldsymbol{x}}_{i,t}$, where $\bar{w}_{i,t}$ is the weight of asset i in a benchmark portfolio (e.g., the value-weighted market portfolio) and $\hat{\boldsymbol{x}}_{i,t}$ is the vector of standardized characteristics of asset i at time t, standardized cross-sectionally. Denote $\hat{\boldsymbol{X}}_t = (\hat{\boldsymbol{x}}_{1,t}, \ldots, \hat{\boldsymbol{x}}_{I_t,t})^\top \in \mathbb{R}^{I_t \times D_x}$, the vector form becomes:

$$\boldsymbol{w}_t = \bar{\boldsymbol{w}}_t + \frac{1}{I_t}\hat{\boldsymbol{X}}_t \boldsymbol{\theta}. \tag{6}$$

Under PPP, the portfolio problem becomes a low-dimensional parameter optimization

problem. For example, with a one-period expected-utility criterion,

$$\boldsymbol{\theta}^*_{\text{PPP}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_t \left[U\left(\left(\bar{\boldsymbol{w}}_t + \frac{1}{I_t} \hat{\boldsymbol{X}}_t \boldsymbol{\theta} \right)^\top \boldsymbol{r}_{t+1} \right) \right]. \tag{7}$$

When $U(\cdot)$ is quadratic or the evaluation criterion is mean-variance-based, the first-order conditions admit particularly tractable expressions, making PPP popular and important.

However, the restrictions and specializations are limiting in that PPP is not naturally designed for general path-level objectives such as Sharpe over a window, survival constraints, delegated-compensation payoffs, or other trajectory functionals unless one introduces substantial state augmentation. Moreover, PPP with linear portfolio weights in the predictors rules out nonlinear transformations and threshold effects ex ante, and treats assets symmetrically through the same coefficient vector, with no explicit mechanism for modeling cross-asset dependence beyond the budget constraint. While nonparametric or locally parametric estimators from sample analogues of the Euler equations mitigate misspecification (Brandt, 1999), the curse of dimensionality when deriving optimal weights using kernel methods or polynomial expansions limits reliable implementation beyond two predictors (e.g., Brandt, 2010), unless one imposes additional structures for dimension reduction (e.g., index regressions, Powell, Stock, and Stoker, 1989; Aït-Sahalia and Brandt, 2001), which again restrict the policy space in the optimization.

2.3 NN Expansion of the Policy Space and Desirability Scores

We first relax the solution-space restriction in PPP by replacing the linear mapping (6) with a flexible NN policy. The simplest such policy uses a NN involving a multilayer perceptron (MLP) to map the current asset state into an asset desirability score:

$$s_{i,t} = \text{MLP}_\pi(\boldsymbol{x}_{i,t}; \boldsymbol{\theta}_\pi), \tag{8}$$

where $\boldsymbol{\theta}_\pi$ denotes the parameters of the NN and $\boldsymbol{x}_{i,t}$ is the observable state vector associated with asset i at date t. The output $s_{i,t}$ is a score for portfolio construction. A higher value indicates that the asset is more attractive for the long position of the portfolio, while a lower value indicates greater attractiveness for the short position, all for a given portfolio objective.

To convert desirability scores into an implementable long-short portfolio, we rank all assets in descending order of their scores and let $o_{i,t}$ denote the rank of asset i at date t. Fix

a basket size G for both long and short positions. For assets in the top G ranks, we assign long weights using a Softmax transformation:

$$w_{i,t}^{+} = \frac{\exp(s_{i,t})}{\sum_{j:o_{j,t}\le G}\exp(s_{j,t})}, \qquad \text{if } o_{i,t} \le G. \tag{9}$$

For assets in the bottom G ranks, we assign short weights using

$$w_{i,t}^{-} = \frac{\exp(-s_{i,t})}{\sum_{j:o_{j,t}>I_t-G}\exp(-s_{j,t})}, \qquad \text{if } o_{i,t} > I_t - G. \tag{10}$$

The resulting long-short portfolio generator can be written compactly as

$$w_{i,t} = \mathrm{LSP}(\boldsymbol{s}_t) = \begin{cases} \dfrac{\exp(s_{i,t})}{\sum_{j:o_{j,t}\le G}\exp(s_{j,t})}, & o_{i,t} \le G, \\ -\dfrac{\exp(-s_{i,t})}{\sum_{j:o_{j,t}>I_t-G}\exp(-s_{j,t})}, & o_{i,t} > I_t - G, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where $\boldsymbol{s}_t = (s_{1,t}, \ldots, s_{I_t,t})^\top$.

This step already expands the solution space drastically relative to PPP: the mapping from characteristics to weights is no longer restricted to be linear or low-dimensional. At the same time, (8)–(11) still use only contemporaneous asset states. In our empirical decomposition, this intermediate model corresponds to the NN-RL benchmark.

2.4 Extracting Sequential and Cross-Asset Dependencies

An asset's investment value is closely related to both its historical performance and its relationships with other assets. Eq. (8) in Section 2.3 models the asset desirability score solely as a function of the observable state of asset i at date t, *i.e.,* $\boldsymbol{x}_{i,t}$—a restriction of the space of the portfolio generation function. We therefore enlarge the policy class by allowing dependencies on cross-sectional relationships among assets, as well as lagged histories of asset-specific and market-wide states.

Specifically, we extend the MLP in Eq. (8) into a stacked architecture consisting of a Sequence Representation Extraction Module (SREM) and a Relation Representation Extraction Module (RREM). The SREM is a sequence neural network that extracts an asset

representation from its historical data. For a look-back window of length $K+1$, define the historical sequence for asset i at date t as $\boldsymbol{Z}_i^{(t)} = (\boldsymbol{z}_{i,t-K}, \ldots, \boldsymbol{z}_{i,t-1}, \boldsymbol{z}_{i,t})$. The SREM takes $\boldsymbol{Z}_i^{(t)}$ as input and generates a representation vector for asset i:

$$\boldsymbol{u}_{i,t} = \text{SREM}\left(\boldsymbol{Z}_i^{(t)}; \boldsymbol{\theta}_{\text{SREM}}\right), \tag{12}$$

where $\boldsymbol{\theta}_{\text{SREM}}$ denotes the parameters of the SREM. The SREM can be implemented using any sequence neural network (Cong, Tang, Wang, and Zhang, 2020), such as a recurrent neural network (RNN), long short-term memory (LSTM), or Transformer. The role of the sequence model is not to assume based on human expertise or insights deterministic dependence in stock returns. Rather, it is to summarize in a data-driven manner lagged state variables whose portfolio relevance may be nonlinear, delayed, and time-varying.

Prior attempts to apply RL-based investment models from computer science have predominantly focused on extracting representations for individual assets (Jin and El-Saawy, 2016; Deng, Bao, Kong, Ren, and Dai, 2017; Ding, Liu, Bian, Zhang, and Liu, 2018), while largely overlooking cross-asset dependencies. AlphaPortfolio therefore adds a Relation Representation Extraction Module (RREM) to model how the investment relevance of one asset depends on other assets in the investable universe.

Let $\boldsymbol{U}_t = \left[\boldsymbol{u}_{1,t} \ldots \boldsymbol{u}_{I_t,t}\right]^\top \in \mathbb{R}^{I_t \times d_u}$ denote the collection of asset representations extracted by the SREM at date t, where I_t is the number of investable assets. The RREM maps these individual asset representations into relation-aware representations:

$$\boldsymbol{C}_t = \text{RREM}(\boldsymbol{U}_t; \boldsymbol{\theta}_{\text{RREM}}), \qquad \boldsymbol{c}_{i,t} = [\boldsymbol{C}_t]_{i,\cdot}, \tag{13}$$

where $\boldsymbol{\theta}_{\text{RREM}}$ denotes the learnable parameters of the RREM and $\boldsymbol{c}_{i,t}$ is the relation-aware representation of asset i. The RREM also induces a date-specific relation matrix

$$\boldsymbol{\Omega}_t = \Omega(\boldsymbol{U}_t; \boldsymbol{\theta}_{\text{RREM}}) \in \mathbb{R}^{I_t \times I_t},$$

whose element $\Omega_{ij,t}$ captures the dependence of asset i on asset j, conditional on the current asset representations $\boldsymbol{U}_t$. Thus, $\boldsymbol{\theta}_{\text{RREM}}$ parameterizes the relation-extraction module, whereas $\boldsymbol{\Omega}_t$ is the state-dependent cross-asset relation matrix induced by that module at date t.

Equivalently, the cross-asset dependency structure can be represented by a graph $G_t =$

$(\mathcal{V}_t, \mathbf{\Omega}_t)$, where the vertices $\mathcal{V}_t = \{1, \ldots, I_t\}$ correspond to investable assets and $\mathbf{\Omega}_t$ gives the edge weights. In graph-based implementations, $\mathbf{\Omega}_t$ can be interpreted directly as edge weights in a graph over assets; in attention-based implementations, it corresponds to the normalized attention weights. More generally, the RREM can be implemented using graph convolutional networks, graph attention networks, message-passing architectures, or attention-based architectures such as the Panel Transformer used in our baseline implementation.

Finally, the relation-aware representation is mapped into the desirability score

$$s_{i,t} = \mathrm{MLP}(\boldsymbol{c}_{i,t}; \boldsymbol{\theta}_{\mathrm{MLP}}),$$

where $\boldsymbol{\theta}_{\mathrm{MLP}}$ denotes the parameters of the MLP mapper. Equivalently, with the MLP applied row-wise,

$$\boldsymbol{s}_t = \mathrm{MLP}(\mathrm{RREM}(\boldsymbol{U}_t; \boldsymbol{\theta}_{\mathrm{RREM}}); \boldsymbol{\theta}_{\mathrm{MLP}}), \qquad \boldsymbol{U}_t = \begin{bmatrix} \mathrm{SREM}\left(\boldsymbol{Z}_1^{(t)}; \boldsymbol{\theta}_{\mathrm{SREM}}\right)^\top \\ \vdots \\ \mathrm{SREM}\left(\boldsymbol{Z}_{I_t}^{(t)}; \boldsymbol{\theta}_{\mathrm{SREM}}\right)^\top \end{bmatrix}.$$

The learnable parameters of the portfolio-generation function are therefore $\boldsymbol{\theta}_{\mathrm{SREM}}$, $\boldsymbol{\theta}_{\mathrm{RREM}}$, and $\boldsymbol{\theta}_{\mathrm{MLP}}$. The realized cross-asset dependencies at date t are summarized by the induced matrix $\mathbf{\Omega}_t$, not by a fixed $I_t \times I_t$ parameter matrix.

Alternative stacking orders. The baseline architecture applies the SREM asset by asset and then applies the RREM to the resulting cross-section of asset representations. This ordering is not essential. One could instead first apply a relation module to the cross-section of asset states at each lag and then apply a sequence module to the resulting relation-aware asset histories. For example, let

$$\boldsymbol{Z}_{t-\ell} = (\boldsymbol{z}_{1,t-\ell}, \ldots, \boldsymbol{z}_{I_t,t-\ell})^\top$$

denote the cross-section of asset states at lag ℓ. A relation-first implementation could form

$$\tilde{\boldsymbol{z}}_{i,t-\ell} = [\mathrm{RREM}_0(\boldsymbol{Z}_{t-\ell}; \boldsymbol{\theta}_{\mathrm{RREM},0})]_{i,\cdot},$$

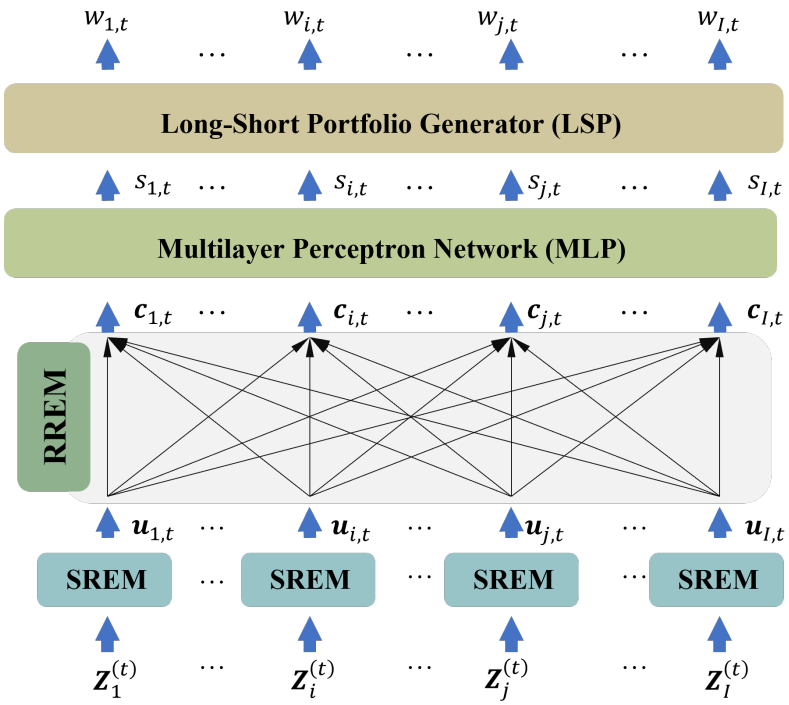

Figure 1: The Overall AlphaPortfolio Architecture.

and then apply the SREM to $(\tilde{\boldsymbol{z}}_{i,t-K}, \ldots, \tilde{\boldsymbol{z}}_{i,t})$. More generally, SREM and RREM can be composed in alternative stacked or sandwich architectures. We focus on the SREM-then-RREM ordering because it is simple, scalable, and performs well in our empirical implementation.

This completes the construction of AlphaPortfolio (Figure 1). Relative to PPP, the admissible policy space is now drastically larger. It is intentionally made very broad by using large neural networks and attention-based modules to represent portfolio policies. While it is not literally equal to the set of all conceivable trading strategies, it is close to unrestricted from the perspective of traditional finance modeling. The main discipline now comes from the portfolio objective, the data, the learning dynamics, and the implementable portfolio-construction map rather than from a tightly hand-specified functional form.

2.5 SREM and RREM Implementation via Panel Transformer

As discussed in Section 2.4, SREM and RREM can in principle be implemented using any sequence neural network and graph neural network. We extend the standard Transformer to panel data settings. The resulting Panel Transformer inherits the highly scalable self-

attention mechanism. The Panel Transformer first uses the self-attention mechanism in the Transformer encoder as its SREM to extract temporal dependencies from lagged asset characteristics. Define the historical sequence for asset i at date t as $\boldsymbol{Z}_i^{(t)} = (\boldsymbol{z}_{i,t-K}, \ldots, \boldsymbol{z}_{i,t-1}, \boldsymbol{z}_{i,t})$. For each lag $\ell \in \{0, \ldots, K\}$, the model maps $\boldsymbol{z}_{i,t-\ell}$ into query, key, and value vectors:

$$\boldsymbol{q}_{i,\ell} = \boldsymbol{W}_s^{(Q)} \boldsymbol{z}_{i,t-\ell}, \qquad \boldsymbol{k}_{i,\ell} = \boldsymbol{W}_s^{(K)} \boldsymbol{z}_{i,t-\ell}, \qquad \text{and} \qquad \boldsymbol{v}_{i,\ell} = \boldsymbol{W}_s^{(V)} \boldsymbol{z}_{i,t-\ell}, \tag{14}$$

where $\boldsymbol{W}_s^{(Q)}$, $\boldsymbol{W}_s^{(K)}$, and $\boldsymbol{W}_s^{(V)}$ are learnable parameter matrices. To quantify the dependence from date $t - \ell'$ to $t - \ell$, we define the attention weight as

$$\mathrm{ATT}_{i,\ell,\ell'} = \frac{\exp(\alpha_{i,\ell,\ell'})}{\sum_{m=0}^{K} \exp(\alpha_{i,\ell,m})}, \qquad \text{where} \qquad \alpha_{i,\ell,\ell'} = \frac{\boldsymbol{q}_{i,\ell}^\top \boldsymbol{k}_{i,\ell'}}{\sqrt{D_k}}. \tag{15}$$

Here, $\alpha_{i,\ell,\ell'}$ measures the relevance between dates $t - \ell'$ and $t - \ell$, and $\mathrm{ATT}_{i,\ell,\ell'}$ is its normalized counterpart obtained via the Softmax function. The representation associated with date $t - \ell$ is then computed as $\boldsymbol{h}_{i,t-\ell} = \sum_{\ell'=0}^{K} \mathrm{ATT}_{i,\ell,\ell'} \, \boldsymbol{v}_{i,\ell'}$. Concatenating the representations across all time steps yields the sequence representation for asset i at date t: $\boldsymbol{u}_{i,t} = \mathrm{Concat}(\boldsymbol{h}_{i,t-K}, \ldots, \boldsymbol{h}_{i,t})$. This temporal self-attention mechanism allows the model to learn which lags are informative, how they interact, and whether more distant information should be emphasized or attenuated. The SREM also incorporates other standard Transformer modules, such as multi-head attention and the feed-forward network. Additional implementation details are provided in Appendix A2.

The Panel Transformer then applies self-attention across assets as RREM to capture cross-sectional dependencies. Given the sequence representations $\boldsymbol{u}_{i,t}$ for all investable assets $i = 1, \ldots, I_t$ at date t, the model constructs

$$\boldsymbol{q}_{i,t}^c = \boldsymbol{W}_c^{(Q)} \boldsymbol{u}_{i,t}, \qquad \boldsymbol{k}_{i,t}^c = \boldsymbol{W}_c^{(K)} \boldsymbol{u}_{i,t}, \qquad \text{and} \qquad \boldsymbol{v}_{i,t}^c = \boldsymbol{W}_c^{(V)} \boldsymbol{u}_{i,t}, \tag{16}$$

where $\boldsymbol{W}_c^{(Q)}, \boldsymbol{W}_c^{(K)}, \boldsymbol{W}_c^{(V)}$ are learnable matrices shared across assets. The dependence of asset i on asset j is measured by

$$\mathrm{ATT}_{i,j,t}^c = \frac{\exp(\alpha_{i,j,t}^c)}{\sum_{j'=1}^{I_t} \exp(\alpha_{i,j',t}^c)}, \qquad \text{where} \qquad \alpha_{i,j,t}^c = \frac{(\boldsymbol{q}_{i,t}^c)^\top \boldsymbol{k}_{j,t}^c}{\sqrt{D_c}}. \tag{17}$$

In the notation of Section 2.4, the Panel Transformer implements the RREM through cross-

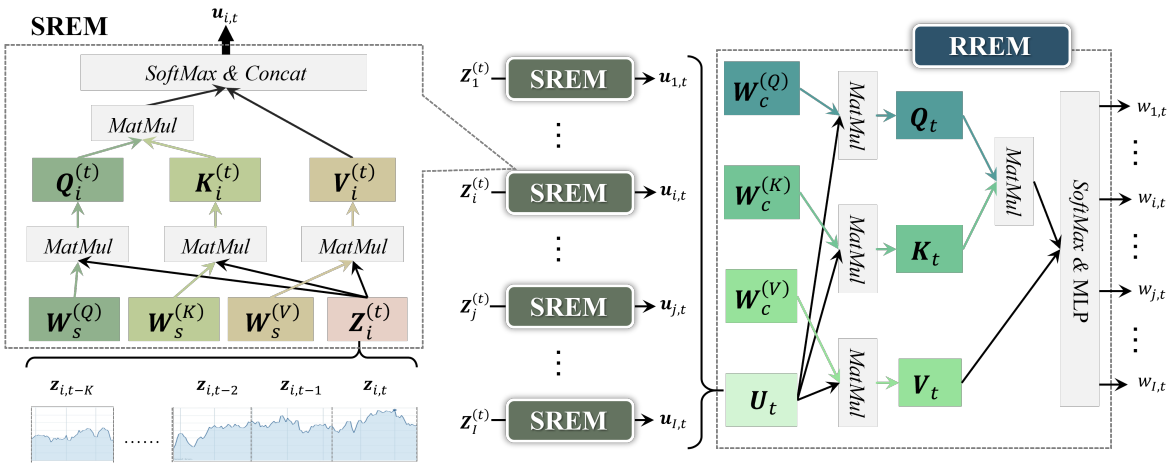

Figure 2: Architecture of the panel Transformer (MatMul indicates matrix multiplication).

asset self-attention. In particular,

$$\Omega_{ij,t} = \mathrm{ATT}^c_{i,j,t}, \qquad \boldsymbol{\Omega}_t = \left[\mathrm{ATT}^c_{i,j,t}\right]_{i,j=1}^{I_t}.$$

Thus, the induced relation matrix $\boldsymbol{\Omega}_t$ is date-specific and state-dependent, rather than a fixed $I_t \times I_t$ parameter matrix. Using these attention weights, the model aggregates information across assets to form the cross-asset representation as $\boldsymbol{c}_{i,t} = \sum_{j=1}^{I_t} \mathrm{ATT}^c_{i,j,t}\, \boldsymbol{v}^c_{j,t}$. The final desirability score of the panel Transformer is then obtained as $s_{i,t} = \mathrm{MLP}(\boldsymbol{c}_{i,t}; \boldsymbol{\theta}_{\mathrm{MLP}})$.

Let $\boldsymbol{s}_t = (s_{1,t}, \ldots, s_{I_t,t})^\top = \mathrm{PTF}(\boldsymbol{\mathcal{Z}}^{(t)}; \boldsymbol{\theta})$, where $\mathrm{PTF}(\cdot)$ denotes the full panel Transformer and $\boldsymbol{\mathcal{Z}}^{(t)} = \left(\boldsymbol{Z}_1^{(t)}, \ldots, \boldsymbol{Z}_i^{(t)}, \ldots, \boldsymbol{Z}_{I_t}^{(t)}\right)$. The portfolio policy is then given by

$$\boldsymbol{w}_t = \mathrm{LSP}\big(\mathrm{PTF}(\boldsymbol{\mathcal{Z}}^{(t)}; \boldsymbol{\theta})\big), \tag{18}$$

where $\mathrm{LSP}(\cdot)$ denotes the long-short portfolio generator defined in Eq. (11).

The architecture of the panel Transformer is illustrated in Figure 2. Here, both SREM and RREM are implemented using the highly scalable self-attention mechanism, which can be expressed as parallelizable matrix multiplications. This structure is highly compatible with GPU acceleration, enabling efficient large-scale model training and inference. Consequently, the framework supports modeling and computation much larger parameter spaces than other implementations, making AlphaPortfolio "large."

2.6 Training AlphaPortfolio via Reinforcement Learning

What remains is to solve (3) over the enlarged policy space defined by (18). The policy-learning step next does not alter the underlying economic problem in Section 2.1; it provides a numerical solution method for it. The object being optimized remains the same trajectory - level criterion $J(\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim P_{\boldsymbol{\theta}}}[H(\tau)]$. When $H(\tau)$ is additively separable and the state is Markov, this problem admits the Bellman recursion in (5). Standard numerical dynamic-programming methods would then approximate either the value function or the policy function and iterate on the Bellman operator. AlphaPortfolio instead parameterizes the policy directly as $\pi_{\boldsymbol{\theta}}$ in (18) and searches over $\boldsymbol{\theta}$ in an enlarged policy class rather than by explicit backward recursion on the value function, to maximize the same economic objective. In this sense, the learning step should be interpreted as approximate policy search for the portfolio-management problem, rather than as a different economic model.

A literal global search is computationally infeasible, especially once the policy is represented by large neural networks. Moreover, many portfolio management objectives do not come with labeled training targets in standard datasets, so standard supervised learning is not natural and is often infeasible. RL therefore provides a natural goal-directed search procedure (that does not require labeled data): The three core elements of RL, namely observed state, action, and reward, map directly into the portfolio-management problem. The environment entails evolving market states; the agent is the portfolio-generator; and the reward is the trajectory-level objective. Thus, the model is trained end-to-end on the economic objective itself rather than on intermediate prediction targets (e.g., expected returns, factor loadings, or pricing errors).

This distinction is economically important. When the model is trained directly on the portfolio objective, it need not expend equal modeling capacity on all assets or all forecast errors. Instead, it can place more weight on the signals, states, and cross-sectional distinctions that are most consequential for the eventual portfolio under the investor's objective and admissible set. For example, under long-only or constrained portfolio choice, improving precision for assets that will never be held is of limited value, whereas getting the ranking and relative desirability of the marginal and heavily weighted assets right is crucial. Likewise, when transaction costs or other implementation frictions matter, a fully objective-driven procedure can favor signals that are not only informative about returns but also more useful once turnover and feasibility are taken into account. This is one reason why direct training

on the economic objective can outperform approaches that first optimize a statistical loss and only later impose portfolio design.

Informally, the "self-learning" in AlphaPortfolio should be understood as a goal-oriented iterative search rather than unconstrained machine reasoning. Starting from an initial parameter vector, the model maps lagged states into portfolio weights, evaluates the resulting trajectory reward under the investor's objective, computes how the reward changes with respect to the parameters, and then updates the parameters in directions that improve the objective. Repeating this process over many trajectories gradually refines the policy. Thus, what replaces the tight functional-form or low-dimensionality restrictions in PPP is not the absence of discipline, but discipline imposed by the economic objective, the architecture, and the gradient-based search procedure.

Our empirical implementation uses a deterministic policy $\boldsymbol{a_t} = \pi_{\boldsymbol{\theta}}(\boldsymbol{Z}_t)$, where $\pi_{\boldsymbol{\theta}}$ is implemented by (18). Let $R^p_{t+1}(\boldsymbol{\theta}) = \boldsymbol{w}_t(\boldsymbol{\theta})^\top \boldsymbol{r}_{t+1}$ denote the realized portfolio return from t to $t+1$. The trajectory objective can be very general. A simple example is the Sharpe ratio computed from $\{R^p_{t+1}(\boldsymbol{\theta})\}_{t=1}^T$ over a T-period investment window. More generally, $H(\tau)$ can incorporate transaction costs, budget constraints, survival or drawdown penalties, delegated-compensation payoffs, CRRA utility of terminal wealth, or other portfolio-management considerations. Compared with PPP, which is most natural under much tighter objective restrictions, AlphaPortfolio can optimize these broader objectives directly.

Accordingly, the training objective is $J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim P_{\boldsymbol{\theta}}}[H(\tau)]$. When $H(\tau)$ is differentiable with respect to the realized portfolio-return path, the empirical implementation uses pathwise derivatives akin to deterministic policy gradients. For a given trajectory τ,

$$\nabla_{\boldsymbol{\theta}} H(\tau) = \sum_{t=1}^{T} \nabla_{R^p_{t+1}} H(\tau)\, \nabla_{\boldsymbol{\theta}} R^p_{t+1}(\boldsymbol{\theta}), \tag{19}$$

where $\nabla_{R^p_{t+1}} H(\tau)$ is the derivative of the trajectory objective with respect to the portfolio return in period $t+1$, and $\nabla_{\boldsymbol{\theta}} R^p_{t+1}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\big(\boldsymbol{w}_t(\boldsymbol{\theta})^\top \boldsymbol{r}_{t+1}\big)$. For example, if $H(\tau)$ is the Sharpe ratio over the investment window, then $\nabla_{R^p_{t+1}} H(\tau)$ is the derivative of that Sharpe ratio with respect to the return in period $t+1$. These derivatives are computed automatically by the deep-learning framework used in implementation.

At iteration n, let $\boldsymbol{\theta}_n$ denote the current parameter vector. We generate M trajectories

over the training sample and update parameters using the average trajectory gradient:

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \lambda \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_n) = \boldsymbol{\theta}_n + \lambda \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{\theta}} H(\tau_m; \boldsymbol{\theta}_n), \tag{20}$$

where λ is the learning rate and τ_m is the m-th sampled trajectory. This iterative optimization gradually refines $\boldsymbol{\theta}$ to maximize the expected trajectory objective.

Relation to dynamic programming. It is useful to clarify how this gradient-based policy search relates to the dynamic programming formulation in Section 2.1. Standard numerical methods, such as value function iteration, policy iteration, or fitted value iteration, solve the Bellman equation (5) by iterating on the value function until convergence, typically requiring discretization of the state-action space or parametric approximation of the value function. These methods are well suited to low-dimensional problems with additive, Markov reward structures, but face well-known difficulties when the state space is high-dimensional, when the objective is a non-additive trajectory functional such as the Sharpe ratio over a window, or when the policy class is a large neural network.

The policy-gradient approach used here targets the same objective, which is to maximize $J(\boldsymbol{\theta})$ in (3). However, it does so by directly differentiating the trajectory-level reward with respect to the policy parameters, bypassing explicit construction of a value function. When $H(\tau)$ is additively separable and Markov, this amounts to solving the same control problem as (5) via direct policy search rather than backward recursion; when $H(\tau)$ is path-dependent, it generalizes beyond the Bellman recursion while still performing goal-directed optimization. The trade-off is that policy-gradient methods forgo the contraction-mapping guarantees of exact dynamic programming in exchange for scalability to the high-dimensional, continuous policy spaces that arise in our application.

Note that if one wishes to work with stochastic policies instead of the deterministic implementation used here, score-function policy-gradient methods can be used; the empirical AlphaPortfolio implementation relies on deterministic policy generation and automatic differentiation because the portfolio map in (18) is differentiable almost everywhere. In implementation, gradients are computed conditional on the selected long-short basket in each update, while the ranking step is treated as locally fixed except at measure-zero tie events. Appendix A5 provides pseudo-code for the iterative training procedure together with a notation table for easy reference.

3 Application: Managing Portfolios of U.S. Equities

The AlphaPortfolio framework is general enough to accommodate both goal-based investment advising for households and active portfolio management by investment funds; our baseline empirical application emphasizes the latter, while the former is illustrated through the alternative objectives and constraints studied in Section 4. We apply the AlphaPortfolio model to trading public equities in the United States with monthly rebalancing. This allows us to not only demonstrate the efficacy of the new framework, but also compare AlphaPortfolio to other studies in asset pricing and investment management in the literature.

Before describing the data, it is useful to clarify how the general state notation in Section 2.1 maps to the empirical implementation. The general state $\boldsymbol{Z}_t$ was defined to potentially include wealth W_t, asset characteristics $\boldsymbol{X}_t$, market-wide variables, and other manager-specific variables such as budgets or survival indicators. In the baseline implementation, $\boldsymbol{Z}_t$ reduces to the panel of lagged asset-level characteristics and market signals: for each asset i, the input is the sequence $\boldsymbol{Z}_i^{(t)} = (\boldsymbol{z}_{i,t-K}, \ldots, \boldsymbol{z}_{i,t})$, where each $\boldsymbol{z}_{i,t}$ contains the 51 firm-level and market variables described below, and $K = 11$ corresponds to a 12-month lookback window. In other words, in the empirical implementation, the general state $\boldsymbol{Z}_t$ at date t is instantiated as the panel of lagged asset-level sequences, i.e. $\boldsymbol{Z}_t = \mathcal{Z}^{(t)} = \left(\boldsymbol{Z}_1^{(t)}, \ldots, \boldsymbol{Z}_i^{(t)}, \ldots, \boldsymbol{Z}_I^{(t)}\right)$. [11] Manager-specific state variables such as wealth, budgets, and survival indicators do not enter the baseline implementation but are incorporated in the extensions of Section 4.2, where the objective or constraints require them.

3.1 Data Description

Our original sample runs from July 1965 to June 2016 and contains approximately 1.76 million month-asset observations. Data for earlier dates are incomplete. We preserve the original design and test period 1990-2016 used in the initial public circulation of the paper, and then add 2017-2024 as a later holdout sample that is "truly" out-of-sample. Monthly stock return data are from the Center for Research in Security Prices (CRSP). Following standard practice in the empirical literature, we restrict attention to common stocks of firms

[11] A superscript $^{(t)}$ denotes a lagged sequence observed up to date t. For example, $\boldsymbol{x}_{i,t}$ refers to contemporaneous characteristics, whereas $\boldsymbol{Z}_i^{(t)}$ denotes the lagged feature sequence of asset i. The general state $\boldsymbol{Z}_t$ is the contemporaneous policy observation for the investable universe, but in the empirical implementation it is instantiated as the panel of lagged asset-level sequences observed up to date t. Thus, all lagged asset characteristics available by date t jointly constitute the current state observed by AlphaPortfolio.

incorporated in the United States and trading on the NYSE, Amex, or Nasdaq. Firms' balance-sheet data come from the Compustat database. To mitigate survivorship bias associated with backfilling, a firm must appear in the data for at least two years before entering the training sample. For OOS test, we only require a firm to be in the dataset for one year.

Similar to Freyberger, Neuhierl, and Weber (2020), we construct firm characteristics and market signals 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. Each input variable is treated as available only in the month after it becomes public, a date that lags behind the reporting date to start with. If a variable is not reported at the monthly frequency, we treat it as unchanged from the previous month. Overall, the baseline implementation uses 51 characteristics observed over a 12-month lookback window, for a total of 612 lagged inputs per asset-month. Appendix B describes the construction of input features.[12] The AlphaPortfolio framework can accommodate macroeconomic variables and alternative data sources, which we leave for future explorations.

3.2 Training and Empirical Details

Our baseline training objective is the out-of-sample Sharpe ratio computed over a subsequent 12-month investment window. To train the model, we use data from July 1965 to the end of 1989 and follow the construction of the portfolio outlined in Section 2, with G chosen such that the long and short positions each take 10% of all the stocks available. We also adopt the standard return normalization when computing the returns of a dollar-neutral portfolio that takes long and short positions of equal dollar size each month (with long weights summing to 1 and short to -1): P&L divided by total gross exposure (long plus short). Then, the return series, and thus Sharpe and regression alpha, are invariant to notional scaling (e.g., doubling both legs leaves returns unchanged). In long–short factor

[12]Tables 1 and 2 in Freyberger, Neuhierl, and Weber (2020) provide an overview of their input features and their summary statistics while Section A.1 describes 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.

studies it is common to report per-leg returns; for readers accustomed to this Fama-French convention, note that our per-gross returns are exactly one-half of per-leg returns.

In the baseline empirical implementation, we adopt a price-taking view with respect to asset-price dynamics. Although RL is known to allow interactions of actions (trading in our setting) and the environment (e.g., market state variables, price impact, etc.), in the baseline we focus on the trial-and-error search benefits of RL. Alternative objectives that interact with the market environment can be accommodated, as we illustrate in Section 4. In the baseline implementation, we do not use a separate validation sample; instead, strict OOS tests provide the main safeguard against overfitting and model selection bias in evaluating the AP performance. We use historical data to proxy for the environment and the rewards to judge the quality of training and adjust hyper-parameters of AP; as such, model selection is embedded in the exploration steps. The training involves 1,788,673 parameters and 640,722 raw stock-month observations, which renders our model over-parameterized and "large," relative to other investment or asset pricing models.

To start, we randomly initialize the parameters (over a wide range within the parameter space), randomly draw a month from the training set without replacement, and use inputs from the preceding 12 months (the drawn month included) and then evaluate performance on the subsequent 12 months (e.g., Sharpe ratio computed based on 12 monthly return observations) as the reward to update the parameters. We are not assuming months to be i.i.d.; rather, within each epoch we sample distinct start months and construct overlapping 24-month episodes consisting of a 12-month lookback window and a subsequent 12-month reward window. We repeat this 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 implementation, we use 30 epochs, which is sufficient for parameter convergence.[13]

After initial training, we evaluate the model out of sample from 1990 to 2016 with monthly rebalancing. All our results are obtained out-of-sample rather than relying on in-sample predictability adopted in traditional statistical tests. This is crucial to avoid overfitting with low signal-to-noise financial data. Note that the AP model is fine-tuned annually in our test samples (rolling updates), which renders our model a hybrid of offline and online RL once deployed as a live strategy. In other words, after seeing one year's performance, we use the

[13]As is standard in deep-learning training, the learning rate η is gradually reduced across epochs: 10^{-4} for the first 5 epochs, 5×10^{-5} for the next 10 epochs, and 10^{-5} thereafter. Convergence is assessed by the stabilization of the loss (negative reward) curve.

additional data 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 (2020). Updating at a lower frequency tends to reduce the OOS performance because of stale information, which works against achieving a superior performance.

3.3 Main Empirical Findings

Table 1 is the central empirical table for understanding how AlphaPortfolio improves upon conventional approaches. To interpret the decomposition, it is useful to distinguish the two dimensions along which AlphaPortfolio differs from or generalizes other models such as PPP, as the ways they contribute to performance are conceptually distinct.

The first is the *end-to-end principle*: optimizing the portfolio objective directly over the policy parameters, bypassing intermediate return estimation. This principle is shared with PPP's central insight. The comparison between the direct-construction models (Panels A–C) and the indirect two-step constructions (Panels D and E) isolates the contribution of this principle: using the same Panel Transformer scores but forming portfolios conventionally rather than through objective-driven training reduces Sharpe ratios by more than half.[14] The principle under the AlphaPortfolio architecture also allows more flexible optimization objectives, as we discuss later.

The second dimension is the *enlarged policy space*, which is new relative to PPP. The classical PPP motivation is that direct parameterization reduces estimation error by replacing $O(N^2)$ moment parameters with a low-dimensional function of characteristics. AlphaPortfolio moves in the opposite direction, estimating close to 2 million parameters from roughly 640,000 observations. The argument for why this works is therefore different from PPP's parsimony rationale. Rather than reducing the number of parameters, AlphaPortfolio disciplines a large parameter space through the economic objective itself, which acts as an implicit regularizer by directing modeling capacity toward the assets and signals most consequential

[14]The indirect-construction panels are particularly informative because they show that the gains do not come merely from using desirability scores as proxies for expected returns. If the scores from the sequence model are used within a traditional two-step procedure—first predicting expected returns and then constructing portfolios—performance deteriorates sharply.

for the portfolio.

The decomposition in Table 1 reveals that the gains from this enlarged policy space arise from several identifiable sources. First, Panel A shows that PPP generates high raw returns but with much higher volatility and substantially larger drawdowns, performing materially worse on the Sharpe-ratio objective. Replacing the linear PPP rule with a neural-network policy introduces nonlinear and threshold-like mappings from characteristics to weights, improving performance particularly in the large-stock subsamples (Panel B). Second, adding the sequence model (SREM) allows the policy to adaptively select which lags of which characteristics are informative — the distillation results in Section 5 confirm that the same variables recur at different lags with different signs, indicating that the model learns a richer lag structure than any fixed rule could capture (Panel C). Third, the RREM introduces inter-firm information, so that an asset's attractiveness depends on the contemporaneous states of other assets (Panel C versus Panel F). Fourth, the RL training accommodates (non-additively-separable) general objectives (including the Sharpe ratio over a rebalancing window), which are difficult to handle in PPP's standard formulation. In short, the relevant mapping from states to portfolio weights is substantially more complex than a linear function of contemporaneous characteristics, and the enlarged policy space is what allows the model to capture this complexity.

We also control for well-established factor models in the table, 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).

A perhaps surprising pattern is that AP's performance *improves* after excluding micro-caps, even though smaller stocks exhibit richer cross-sectional variation in characteristics and higher average returns. In the literature, small stocks often "perform better" largely because performance is evaluated using raw returns or factor-adjusted alphas, which reward high expected returns without directly penalizing volatility. By contrast, AlphaPortfolio in the baseline optimizes end-to-end the Sharpe ratio, which explicitly trades off return against volatility. While microcaps offer higher expected returns and greater characteristic dispersion, they also contribute disproportionately to portfolio variance and turnover. When

microcaps are excluded from the investable universe, the model operates over a more homogeneous cross section in which the mapping from characteristics to desirability scores is less contaminated by extreme observations, and the resulting portfolios are more stable. This interpretation is consistent with the observation that AP does not select small and illiquid stocks even when they are available — a consequence of the direct optimization of the Sharpe ratio rather than of an ex post filter.

Table 2 next evaluates the final AlphaPortfolio model against a broader set of benchmarks commonly used in the literature. In contrast to Table 1, which decomposes the gains from direct construction, sequence modeling, and cross-asset modeling, Table 2 is best read as an external-benchmark comparison. Three patterns stand out. First, AP's advantage is clearest in risk-adjusted performance rather than raw return: because the model is trained directly on the Sharpe-ratio objective, it delivers high returns with unusually low volatility, turnover, and drawdown rather than simply loading on the highest-return names. Second, the comparison with NP, one of the best-performing contemporary nonparametric ML portfolio strategies (Freyberger, Neuhierl, and Weber, 2020), highlights its efficacy and practical implementability. NP achieves a higher Sharpe ratio in the unrestricted sample, but this advantage disappears once the smallest stocks are excluded, whereas AP's performance improves in the larger-stock universes, consistent with AP learning signals that are concentrated in the more liquid and implementable part of the cross section. Third, the factor-adjusted alphas remain large and statistically significant across specifications even though the R^2 values vary across factor models, reflecting nonzero but moderate exposure to conventional factors rather than instability in the abnormal-performance estimates.

Columns (1)-(3) display the various moments of the returns of AP as well as metrics such as turnover. AP achieves an OOS Sharpe ratio of 2.0 in the full test dataset 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). Apparently, the AP performance is not driven by microcaps and can be implemented without liquidity concerns.[15] If we restrict our attention to the top 90 percentile of stocks in terms of market cap, one thousand dollars invested by AP at the start of 1990 would become 70,800 dollars

[15]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 illiquidity and trading frictions render many anomalies in academic studies infeasible for trading.

Table 1: Comparison with Alternative Models in Out-of-Sample Performance

This table reports performance for parametric portfolio policies (PPP), NN-RL Portfolio, AlphaPortfolio without RREM and AlphaPortfolio using Sharpe ratio as its training objective, focusing on long/short stocks from 1990 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 (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). The first 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.

	Performance				Excess Alpha					
	(1)	(2)	(3)		(4)	(5)	(6)	(7)	(8)	(9)
Firms	All	$> q_{10}$	$> q_{20}$	Factor Models	All $\alpha(\%)$	R^2	$> q_{10}$ $\alpha(\%)$	R^2	$> q_{20}$ $\alpha(\%)$	R^2
Panel A: Performance of PPP										
Return(%)	34.69	33.41	30.17	CAPM	28.34***	0.193	26.47***	0.240	22.76***	0.335
Std.Dev.(%)	28.77	28.23	25.53	FFC	27.91***	0.557	25.52***	0.660	21.82***	0.784
Sharpe	1.21	1.18	1.18	FFC+PS	27.04***	0.558	23.68***	0.662	19.28***	0.790
Skewness	1.66	1.54	0.96	FF5	24.22***	0.469	20.19***	0.569	16.12***	0.686
Kurtosis	7.77	9.71	6.28	FF6	28.41***	0.560	24.39***	0.664	20.10***	0.792
Turnover	0.16	0.14	0.14	SY	24.28***	0.560	23.16***	0.633	21.93***	0.629
MDD	0.52	0.38	0.36	Q4	27.01***	0.321	23.50***	0.421	19.50***	0.525
Panel B: Performance of NN-RL										
Return(%)	14.0	18.31	15.65	CAPM	14.31***	0.003	18.27***	0.0	15.36***	0.004
Std.Dev.(%)	11.54	9.67	9.07	FFC	14.7***	0.048	17.32***	0.266	14.63***	0.249
Sharpe	1.21	1.89	1.73	FFC+PS	13.17***	0.058	16.52***	0.27	13.73***	0.255
Skewness	2.12	3.41	4.08	FF5	14.61***	0.034	17.43***	0.267	14.0***	0.24
Kurtosis	15.99	20.61	29.86	FF6	15.38***	0.053	17.79***	0.274	14.42***	0.25
Turnover	0.48	0.45	0.48	SY	9.84***	0.059	14.69***	0.404	12.29***	0.372
MDD	0.42	0.02	0.01	Q4	15.19***	0.019	17.99***	0.269	14.66***	0.23
Panel C: Performance of AlphaPortfolio without RREM										
Return(%)	9.86	15.48	16.22	CAPM	9.35***	0.017	14.34***	0.097	14.71***	0.105
Std.Dev.(%)	7.9	7.28	9.3	FFC	10.45***	0.142	14.78***	0.369	14.85***	0.314
Sharpe	1.25	2.13	1.74	FFC+PS	10.66***	0.142	14.05***	0.374	14.17***	0.317
Skewness	1.11	1.64	4.62	FF5	10.88***	0.17	15.04***	0.382	15.03***	0.352
Kurtosis	4.65	5.3	40.13	FF6	11.59***	0.205	15.86***	0.437	15.91***	0.39
Turnover	0.28	0.24	0.24	SY	9.9***	0.152	17.03***	0.409	17.51***	0.361
MDD	0.19	0.03	0.02	Q4	12.2***	0.224	16.22***	0.474	16.23***	0.408
Panel D: Performance of Value Weight Indirect Portfolio Construction										
Return(%)	3.25	4.13	5.45	CAPM	2.75	0.013	3.15*	0.047	4.43***	0.062
Std.Dev.(%)	8.74	9.07	8.15	FFC	1.81	0.128	2.38	0.174	3.79**	0.169
Sharpe	0.37	0.46	0.67	FFC+PS	1.50	0.128	2.00	0.175	3.58**	0.170
Skewness	-1.79	-1.15	-1.65	FF5	2.56	0.134	3.69**	0.196	4.68***	0.184
Kurtosis	27.87	19.70	28.84	FF6	2.47	0.135	3.67**	0.196	4.75***	0.184
Turnover	0.29	0.33	0.33	SY	1.00	0.138	3.22	0.201	4.77**	0.193
MDD	0.14	0.13	0.10	Q4	2.56	0.118	3.68**	0.181	4.38***	0.159
Panel E: Performance of Equal Weight Indirect Portfolio Construction										
Return(%)	8.79	19.34	19.00	CAPM	9.38***	0.016	17.76***	0.072	17.51***	0.089
Std.Dev.(%)	9.37	11.72	9.98	FFC	8.49***	0.196	17.52***	0.274	17.30***	0.315
Sharpe	0.94	1.65	1.90	FFC+PS	8.54***	0.196	17.23***	0.274	17.13***	0.315
Skewness	2.47	6.01	3.90	FF5	7.54***	0.205	16.80***	0.280	16.89***	0.314
Kurtosis	19.87	66.84	30.48	FF6	8.09***	0.219	17.85***	0.315	17.78***	0.348
Turnover	0.31	0.32	0.33	SY	9.64***	0.194	20.57***	0.288	19.09***	0.361
MDD	0.07	0.03	0.03	Q4	8.39***	0.182	18.23***	0.301	17.91***	0.333
Panel F: Performance of AlphaPortfolio										
Return(%)	17.00	17.09	18.06	CAPM	13.88***	0.005	13.18***	0.088	14.01***	0.102
Std.Dev.(%)	8.48	7.39	8.19	FFC	14.24***	0.052	13.40***	0.381	14.69***	0.465
Sharpe	2.00	2.31	2.21	FFC+PS	13.71***	0.054	12.34***	0.392	13.30***	0.480
Skewness	1.42	1.73	1.91	FF5	15.25***	0.120	13.83***	0.426	14.66***	0.435
Kurtosis	6.35	5.70	5.97	FF6	15.63***	0.128	14.46***	0.459	15.79***	0.516
Turnover	0.26	0.24	0.26	SY	17.43***	0.037	15.75***	0.332	16.97***	0.394
MDD	0.08	0.02	0.02	Q4	15.95***	0.121	15.00***	0.495	16.17***	0.521

by the end of 2016.

Table 2: Out-of-Sample Performance of AlphaPortfolio from 1990 to 2016

Panel A compares AP's performance with one benchmark model proposed in Freyberger, Neuhierl, and Weber (2020). NP and AP denote the nonparametric model in Freyberger, Neuhierl, and Weber (2020) and the AlphaPortfolio, respectively. In the table, NP is tested in 1991-2014 (their test sample period), while AP is tested on the full test period (1990-2016) and AP* is tested on the same period as NP. In each month, AlphaPortfolio constructs a long-short portfolio of stocks in highest/lowest decile of desirability scores. The detailed investment strategy is described in Section 2.3. Parameters are initially obtained from the training periods, then fine-tuned once a year in the OOS periods (rolling update). q_n symbolizes the n^{th} NYSE size percentile. Portfolio performance metrics, with Return, Std.Dev., and Sharpe ratio are all annualized. Panel B presents the result of alphas generated by further adjusting portfolio returns 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), Hou-Xue-Zhang 4-factor model (Q4), IPCA 3- and 4-factor models, and Cong et al. (2025) P-Tree 3- and 4-factor models. Again, (1)-(3) present the alphas for the overall sample whereas (4)-(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.

Firms	All			$> q_{10}$			$> q_{20}$		
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Panel A: Performance of AlphaPortfolio and the Nonparametric Model									
	AP	NP	AP*	AP	NP	AP*	AP	NP	AP*
Return(%)	17.00	45.84	15.60	17.09	21.12	17.70	18.06	15.48	17.90
Std.Dev.(%)	8.48	16.66	8.20	7.39	13.27	7.60	8.19	14.90	8.60
Sharpe	2.00	2.75	1.90	2.31	1.60	2.33	2.21	1.04	2.08
Skewness	1.42	3.53	1.20	1.73	0.30	1.77	1.91	-0.50	1.88
Kurtosis	6.35	19.56	6.54	5.70	7.80	5.57	5.97	13.06	5.46
Turnover	0.26	0.69	0.26	0.24	0.74	0.24	0.26	0.74	0.26
MDD	0.08	0.10	0.08	0.02	0.27	0.02	0.02	0.36	0.08
Panel B: Excess Alpha for AlphaPortfolio									
Models	α(%)	R^2	t-stat	α(%)	R^2	t-stat	α(%)	R^2	t-stat
CAPM	13.9***	0.005	10.13	12.2***	0.088	11.63	14.0***	0.102	10.58
FFC	14.2***	0.052	10.26	13.4***	0.381	13.58	14.7***	0.465	12.36
FFC+PS	13.7***	0.054	9.36	12.3***	0.392	11.94	13.3***	0.480	10.88
FF5	15.3***	0.120	10.97	13.8***	0.426	14.43	14.7***	0.435	12.06
FF6	15.6***	0.128	11.14	14.5***	0.459	15.18	15.8***	0.516	12.97
SY	17.4***	0.037	9.90	15.8***	0.332	12.33	17.0***	0.394	11.40
Q4	16.0***	0.121	11.33	15.0***	0.495	15.92	16.2***	0.521	12.89
IPCA(3)	18.1***	0.145	9.96	16.5***	0.353	12.02	15.2***	0.271	9.15
IPCA(4)	17.4***	0.158	9.82	16.1***	0.359	11.95	14.8***	0.267	9.04
P-Tree(3)	13.9***	0.018	3.34	8.3**	0.075	2.36	12.2***	0.048	2.99
P-Tree(4)	14.1***	0.019	3.27	6.9*	0.082	1.89	10.6**	0.054	2.51

Both high returns and low volatility contribute to the high Sharpe ratio observed. Moreover, the full AP uses a lower frequency of rebalancing (monthly), turnover, and maximum drawdown relative to other (high-frequency) machine learning strategies or traditional, anomaly-based trading. The performance would be reduced by half if after ranking the assets using desirability scores, we value-weight the stocks instead. In addition, the average holding period is four months, which is easy to implement. Columns (4)-(9) report that AP has a significant and large annualized α after controlling for various factors, which is only lower than the baseline PPP. This holds for recently published latent factor models too. For example, controlling for IPCA factors constructed using the 36 characteristics in Kelly, Pruitt, and Su (2019), AP generates 17.4%, 16.1%, and 14.8% α in the full, $> q_{10}$, and $> q_{20}$ samples respectively, all significant at the 1% level.

Note that NN-RL and AP with or without RREM do 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 command 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.

Relative to AP without RREM, the full AlphaPortfolio further improves Sharpe ratios (0.47 in Sharpe on average) and risk-adjusted returns (3.53 on average). In the full sample, the value-weighted indirect portfolio achieves an out-of-sample Sharpe ratio of only 0.36, far below the Sharpe ratio of about 2 achieved by AlphaPortfolio. This piece of evidence indicates that the one-step, objective-driven RL construction plays a central role.

An intuitive interpretation is that the one-step construction changes not only how the portfolio is formed, but also what the model learns to pay attention to. A conventional indirect procedure can still use rich scores as proxies for expected returns, yet if those scores are learned without reference to the final investment problem, they need not emphasize the assets and distinctions that are most valuable for the realized portfolio. By contrast, AlphaPortfolio's objective-driven training aligns representation learning, ranking, and portfolio formation under the same economic target, which helps explain why the gains do not reduce to a generic return-prediction improvement.

Table 2 further demonstrates the efficacy of RL and AI for investment. Panel A compares AP with the "nonparametric" (NP) model and portfolio strategy in Freyberger, Neuhierl, and Weber (2020). AP compares favorably with most machine-learning-based strategies

in the literature. We pick NP as a benchmark because in addition to the fact that we use similar firm characteristics as in their paper for inputs, NP was arguably the best-performing machine learning model in asset pricing contemporary to our initial paper. NP achieves a higher Sharpe ratio on its test sample in 1991-2014. Once we focus on larger and therefore typically more liquid, 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.[16]

3.4 Further Diagnostics of AlphaPortfolio

Factor loadings. To better understand AlphaPortfolio, we examine its exposure to seven commonly used benchmark factor models. Specifically, we estimate the portfolio's factor loadings (βs) under each model and assess their statistical significance.

Table 3: Factor Loadings of AlphaPortfolio

This table reports the factor loadings of AlphaPortfolio under several benchmark multi-factor models. The three columns correspond to the full sample, and subsamples excluding the bottom 10% and 20% of NYSE stocks by market capitalization. "*," "**," and "***" denote significance at the 10%, 5%, and 1% levels.

Model	Factor	All	$> q_{10}$	$> q_{20}$
CAPM	MKT	0.041	0.128***	0.172***
FFC	MKT	0.003	0.068***	0.102***
	SMB	0.130***	0.310***	0.360***
	HML	-0.017	0.064*	0.157***
	UMD	-0.054*	-0.066***	-0.114***
FFC+PS	MKT	0.008	0.078***	0.113***
	SMB	0.129***	0.307***	0.357***
	HML	-0.015	0.067*	0.160***
	UMD	-0.053*	-0.064***	-0.112***
	LIQ	-0.015	-0.031**	-0.037**
FF5	MKT	-0.016	0.069**	0.101***
	SMB	0.012	0.221***	0.274***
	HML	0.023	0.087*	0.252***
	RMW	-0.322***	-0.231***	-0.233***
	CMA	0.150*	0.145**	0.017
FF6	MKT	-0.028	0.053*	0.074**
	SMB	0.018	0.229***	0.287***
	HML	-0.006	0.049	0.187***
	RMW	-0.312***	-0.218***	-0.209***
	CMA	0.169*	0.170**	0.061
	UMD	-0.049*	-0.064***	-0.109***
SY	MKT	-0.002	0.106***	0.106***
	SMBSY	0.126**	0.336***	0.382***
	MGMT	0.013	0.143***	0.170***
	PERF	-0.103***	-0.116***	-0.184***
Q4	MKT	-0.041	0.030	0.079***
	ME	0.049	0.225***	0.254***
	IA	0.053	0.162***	0.242***
	ROE	-0.300***	-0.316***	-0.323***

Table 3 reports AlphaPortfolio's loadings on standard benchmark factor models in the full sample and in subsamples excluding the bottom 10% and 20% of NYSE stocks by market

[16]The superior AP performance here should not be interpreted as invalidating other models whose focus is on minimizing pricing errors for asset pricing rather than directly optimizing portfolio performance.

capitalization. Two patterns stand out. First, the factor exposures change meaningfully once microcaps are excluded, especially along the market, size, and value dimensions. Second, despite these nonzero exposures, the portfolio is far from a simple repackaging of standard factors: the loadings are neither uniformly large nor stable across models, and the main performance results remain strong even after controlling for these benchmark factors. This evidence supports the interpretation that AlphaPortfolio exploits richer cross-sectional patterns than those captured by conventional factor portfolios alone.

The factor-loading profile is also consistent with the implementation results reported earlier. In particular, once microcaps are excluded, AlphaPortfolio exhibits larger market, size, and value exposures, which aligns with the lower turnover, lower drawdown, and greater implementability documented for the large-stock subsamples. Across models, one pattern is especially robust: the strategy loads positively on size-related factors and negatively on profitability-related factors, while market exposure tends to increase as the investable universe shifts toward larger stocks.

The factor exposures also help rationalize the relatively low turnover of AlphaPortfolio. The strategy loads strongly and significantly on size-related factors such as SMB, while its exposure to momentum (UMD) is modest and negative in the specifications where momentum is included. This is relevant because different factor styles imply very different trading intensities. Frazzini, Israel, and Moskowitz (2018) report that momentum has much higher turnover than other common factors, with estimated turnover of roughly 1.13 for UMD, compared with 0.27 for SMB and 0.45 for HML. Since AlphaPortfolio has limited exposure to the high-turnover momentum factor and instead loads primarily on the relatively lower-turnover size factor, its portfolio weights tend to adjust more gradually over time.

Industry attribution and allocations. We perform industry/style attribution tests and report the results in Table 4. After regressing AP's monthly OOS returns (1990-2016) on the 12 industries according to Fama and French, the intercept is both economically and statistically significant.[17] We also do not see significant loadings on most industry portfolios except for durables (positive), energy (negative), and retail (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

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

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 – 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	
$> q_{10}$														
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	
$> q_{20}$														
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	

versus value weighing. The Pearson correlation between the vector of portfolio weights and market capitalization is significantly negative in fewer than 15% of months. In that sense, the portfolio is typically much closer to value-weighted, which is easier to construct in practice than to the equal-weighted constructions often used in anomaly studies.

In fact, we use the softmax function to generate portfolio weights from desirability scores for two reasons. First, the optimization of deep neural networks relies on a gradient-based backpropagation algorithm (Rumelhart, Hinton, and Williams, 1986). Our softmax function is differentiable. In contrast, value weights bear no gradient relationship to desirability scores, implying that constructing value-weighted portfolios blocks the backpropagation procedure for model training. Second, our softmax-generated portfolio outperforms in OOS Sharpe ratio, risk-adjusted returns, portfolio turnover, maximum drawdown, etc. One can alternatively take long positions in the top 10% and short positions in the bottom 10% based on the score using market capitalization instead of the softmax transformation to determine portfolio weights. But the resulting portfolio has an OOS Sharpe ratio below 1.5 and risk-adjusted return below 5%, not to mention that turnover and maximum drawdown are all higher than those of AP.

Performance over time. Given our high-dimensional inputs involving known anomalies, there is a natural concern about data mining. Moreover, 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 anomalies that were known only ex post or have been traded away in recent years or on seeming anomalies from data snooping? To answer this, we restrict the test to a subsample of more recent years (2001-2016). As Table 5 reveals, AP's performance remains robust.

Table 5: Out-of-Sample Performance Post 2000

This table reports alphas for portfolios of long/short portfolios in the highest/lowest decile of desirability scores from 2001 to 2016. Return, Std.Dev., and Sharpe ratio are all annualized. 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 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		$> q_{10}$		$> q_{20}$	
					$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	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 3. Overall, the Sharpe ratio is particularly high at the start of the test sample but does not exhibit particular trends since the 2000s. Excess alphas fluctuate but show no sign of attenuation.

Performance in the 2017-2024 "true" test sample. The initial public circulation of our study in 2019 uses a test dataset ending in 2016. Table 6 reports AlphaPortfolio's performance in the additional holdout sample from 2017 to 2024. To preserve the spirit of

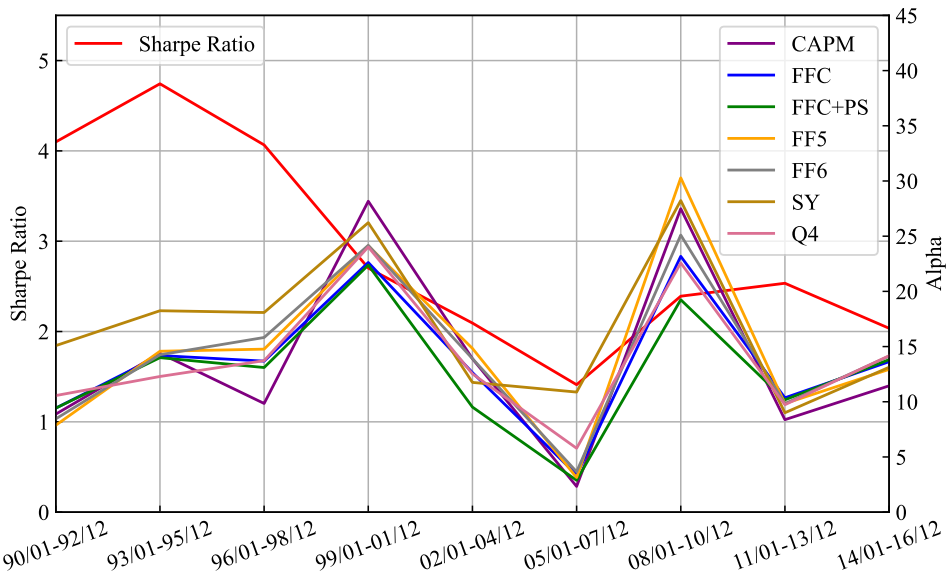

Figure 3: Trends of AlphaPortfolio Performance.

Table 6: Out-of-Sample Performance for AlphaPortfolio from 2017 to 2024

In each month, AlphaPortfolio constructs a long-short portfolio of stocks in the highest/lowest decile of desirability scores. Parameters are initially obtained from the training periods, then fine-tuned once a year in the OOS periods (rolling update). q_n symbolizes the n^{th} NYSE size percentile. Columns (1)-(3) display portfolio performance metrics, with Return, Std.Dev., and Sharpe ratio all annualized. Columns (4)-(9) further adjust portfolio returns 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), IPCA 4-factor model (IPCA(4)), and Hou-Xue-Zhang 4-factor model (Q4). Again, (4)-(5) present the alphas for the overall sample whereas (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		$> q_{10}$		$> q_{20}$	
					α(%)	R^2	α(%)	R^2	α(%)	R^2
Return(%)	19.01	21.27	17.71	CAPM	17.75***	0.008	20.27***	0.034	16.61***	0.035
Std.Dev.(%)	11.65	7.58	6.68	FFC	17.17***	0.16	20.95***	0.261	17.29***	0.315
Sharpe	1.63	2.81	2.65	FFC+PS	17.87***	0.163	20.46***	0.265	15.98***	0.352
Skewness	1.16	0.45	0.3	FF5	18.19***	0.21	21.54***	0.253	18.09***	0.266
Kurtosis	1.74	0.02	0.38	FF6	18.21***	0.21	20.93***	0.283	17.36***	0.322
Turnover	0.22	0.25	0.24	Q4	20.31***	0.103	22.38***	0.195	18.14***	0.22
MDD	0.06	0.03	0.02	IPCA(4)	16.78***	0.171	16.58**	0.054	16.15**	0.064

a genuine out-of-sample exercise, we keep the original model design fixed and retrain only using data available through 2016 before evaluating on the new period. The baseline Sharpe ratio and excess α remain strong in this holdout sample and is still higher in the large-stock subsamples. Overall, the additional holdout results support the view that the main findings are not artifacts of the original sample period.[18]

4 Robustness, Flexibility, and Practical Relevance

We demonstrate that the AlphaPortfolio (AP) framework is robust to standard economic and trading frictions, flexible in the portfolio objectives it accommodates, and practically implementable at scale. On robustness, we show that performance persists under alternative subsamples, exclusion of microcaps and unrated/downgraded names, transaction-cost and price-impact adjustments, and across market states (volatility, liquidity, sentiment). On flexibility, AP directly optimizes path-level objectives—including Sharpe over rebalancing windows, scale constraints, and delegated-management payoffs—without relying on two-step proxies. In particular, although the model specification treats portfolio managers as price-takers, the framework can accommodate rich action-state interactions, as illustrated through the fund survival objective. On implementation, AP yields moderate turnover, stable risk, and capacity consistent with institutional constraints typically required in the industry, and can be adapted to implementation features commonly encountered in institutional settings. Collectively, these results indicate that AP's improvements are not artifacts of a particular sample or objective and that the framework is compatible with several practically relevant portfolio-management settings.

4.1 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. Note that the results reported below after imposing various economic restrictions are only lower bounds on AP's performance, as we do not retrain AP on the subsamples excluding certain stocks or under specific restrictions. Instead, we simply set the portfolio weights

[18]Note that information on the Stambaugh and Yuan factors is not available for this sample period. Instead, excess alpha on IPCA 4-factor model is reported in Table 6.

of the excluded stocks or non-admissible stocks under the specific restrictions to zero. When we incorporate restrictions as constraints into training, performance likely improves.

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 may be driven by microcap and costly-to-trade stocks (Novy-Marx and Velikov, 2016; Hou, Xue, and Zhang, 2020). Avramov, Cheng, and Metzker (2019) find that machine learning techniques face similar issues: return predictability and the anomalous patterns are concentrated in difficult-to-arbitrage stocks and episodes of high limits to arbitrage.

Table 2 rules out that AP's performance is driven by the bottom 10% and 20% of stocks by market capitalization. If anything, performance improves after excluding microcaps, which shows that AP's results are not driven by the smallest stocks and remain strong in the more implementable large-stock subsamples. 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.

Turnover, shorting, and transaction costs. Following Koijen, Moskowitz, Pedersen, and Vrugt (2018); Freyberger, Neuhierl, and Weber (2020), we calculate turnover using $Turnover_t = \frac{1}{4}\sum_i \left|w^i_{t-1}(1+r^i_t) - w^i_t\right|$, where the coefficient $\frac{1}{4}$ avoids double counting (a factor of 2) and adjusts for the fact that the long/short strategies have \$2 exposure (another factor of 2). Gu, Kelly, and Xiu (2020) point out that many strategies have turnovers well above 100% monthly, using $Turnover^{GKX}_t = \sum_i \left|w_{i,t+1} - \frac{w_{i,t}(1+r_{i,t+1})}{1+\sum_j w_{j,t}r_{j,t+1}}\right|$ as an alternative turnover measure.[19] AP still has rather low turnovers relative to other traditional anomalies or machine learning strategies, whether we sum up the long/short-leg turnovers, or directly calculate the turnover of long-short portfolio using this alternative measure.

With low turnovers, AP's performance stands out even after incorporating transaction costs (but without retraining the model). For example, setting a cost at 0.1%, AP still yields OOS 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 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

[19]For proper normalization for long-short portfolios, we have added 1 in the denominator.

Table 7: Out-of-Sample Performance of Long-Only AP

This table presents the OOS performance of a long-only AlphaPortfolio taking positions only in the highest decile of desirability scores, with Return, Std.Dev., and Sharpe ratio all annualized. 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 (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). The first 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		$> q_{10}$		$> q_{20}$	
					$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	22.41	37.60	44.27	CAPM	12.88***	0.477	25.80***	0.508	31.79***	0.485
Std.Dev.(%)	19.28	25.08	27.63	FFC	12.54***	0.791	25.79***	0.798	30.97***	0.733
Sharpe	1.16	1.50	1.60	FFC+PS	11.02***	0.795	22.20***	0.806	28.20***	0.738
Skewness	0.67	1.03	1.55	FF5	9.91***	0.731	21.62***	0.755	27.84***	0.696
Kurtosis	5.64	4.72	7.55	FF6	12.10***	0.791	24.12***	0.802	30.31***	0.734
Turnover	0.38	0.44	0.48	SY	15.38***	0.720	26.29***	0.739	31.68***	0.681
MDD	0.24	0.27	0.25	Q4	14.51***	0.698	26.83***	0.714	33.01***	0.656

positions both contribute significantly to the performance. If anything, the long positions play a more dominant role. For the long-short AlphaPortfolio, 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.

For robustness, we also train a long-only AlphaPortfolio, and the results are reported in Table 7. The OOS Sharpe ratio and excess alphas are lower than the long-short AlphaPortfolio but higher than most other long-only strategies and anomalies.

Lookback window and holding periods. In Table 8, we show how AP performance is robust under different lookback windows (how far we look back for input variables) and holding periods. The various lookback windows involve a bias-variance trade-off and while looking back 6-24 months all produce high OOS Sharpe ratios, the 12 month window appears to be approximately the optimal choice.

The fact that the out-of-sample performance is still rather significant even for holding periods of 6 and 12 months indicate AP is not merely capturing temporary price patterns. AP allows long-term planning and portfolio management with lower rebalancing frequencies.

Table 8: Out-of-Sample Performance with Different Lookback Window and Holding Periods

Performance of long/short portfolios using the highest/lowest deciles of desirability scores from 1990 to 2016. We conduct the experiments for subsamples excluding microcap firms under the 10^{th} NYSE size percentile.

Lookback Window	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
		6 months				12 months				18 months				24 months		
Holding Period	Number of months															
	1	3	6	12	1	3	6	12	1	3	6	12	1	3	6	12
Return(%)	11.86	8.32	6.93	6.72	17.09	10.76	8.67	7.74	12.81	8.58	7.26	6.71	11.56	8.70	6.97	6.92
Std.Dev.(%)	6.44	6.07	5.41	5.77	7.39	7.59	7.29	8.19	6.45	6.49	5.90	6.33	6.48	7.02	7.18	7.94
Sharpe	1.84	1.37	1.28	1.16	2.31	1.42	1.19	0.94	1.99	1.32	1.23	1.06	1.78	1.24	0.97	0.78
Skewness	1.22	0.27	0.31	-0.22	1.73	1.08	1.28	0.96	1.36	0.59	0.57	0.39	1.53	1.21	0.79	2.05
Kurtosis	4.01	0.58	0.91	0.47	5.70	1.96	3.41	0.99	4.26	1.84	0.88	0.55	6.20	3.25	1.87	6.16
Turnover	0.21	0.25	0.26	0.27	0.24	0.27	0.29	0.29	0.23	0.25	0.26	0.28	0.19	0.22	0.25	0.25
MDD	0.04	0.05	0.05	0.04	0.02	0.02	0.03	0.03	0.04	0.04	0.03	0.03	0.03	0.03	0.04	0.02

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 9. 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 mostly an artifact that AP is not reestimated 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 OOS 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 OOS tests of AP only perform well during high investor sentiment and high

Table 9: OOS Performance of Portfolios Excluding Unrated (and Downgraded) Stocks

This table reports alphas for portfolios of long/short stocks in the highest/lowest decile of desirability 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 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, with Return, Std.Dev., and Sharpe ratio all annualized. 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 10: Out-of-Sample Performance of AP Under Various Market Conditions

This table reports annualized 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 volatility 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	(30.636)	(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

market volatility, AP may not be implementable in practice due to limits to arbitrage.

To investigate this issue, we examine the AP performance in subperiods 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 volatility 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 stocks in a month.[20] We divide the full sample into two subperiods using the median breakpoints of $SENT$, VIX, $MKTVOL$, and $MKTILLIQ$ over the whole sample period. We report the analyses in Table 10. For brevity, we only present baseline samples and FF6-adjusted α of the portfolios.

Performance is state-dependent, as expected under time-varying risk premia and limits to arbitrage, but AlphaPortfolio delivers economically and statistically significant results across market regimes and subperiods. In each case, the strategy's alphas remain positive and statistically significant, and Sharpe ratios remain within practical bounds. These findings indicate that the framework is not confined to a particular regime and that performance persists across time and market states.

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 to exhibit a high Sharpe ratio and significant α even after controlling for various factors, regardless of the market conditions. AP is particularly 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.

4.2 Alternative Objectives: Agency, Constraints, and Frictions

The framework directly optimizes any cumulative reward functional $H(\tau)$, including Sharpe over rebalancing windows, cost/impact–adjusted returns, drawdown/survival constraints, delegated-compensation payoffs, and wealth-dependent budgets, while nesting classical parametric policies (e.g., PPP) as special cases. This flexibility, rather than mere generality, is what allows us to accommodate a wider range of portfolio management objectives than typical two-step academic studies. Our use of "flexibility" emphasizes practical objective design and constraint handling. In addition, the policy class is general in the sense that it nests PPP and related parametric specifications and can represent nonlinear, path-

[20]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.

dependent, and cross-asset interactions when needed; we focus on flexibility because it is the economically relevant dimension for portfolio management.

Thus far, the baseline implementation uses a deterministic policy with the portfolio manager being a price-taker. This common assumption in many asset pricing studies implies a limitation on the scale of AP. Depending on the size of the portfolio, transaction costs and price impacts could differ significantly, and should be taken into consideration (Gârleanu and Pedersen, 2013; Novy-Marx and Velikov, 2016). More generally, an RL learner's action could alter the market state. While it is too complicated to model the interactions between actions and market environments completely, RL is known for excelling at incorporating such interactions, and the AP framework is flexible enough to accommodate various scenarios in portfolio management. We provide several illustrative examples, in which we retrain the model allowing the investment actions to adjust through interacting with the market environment specified by us or in the literature.

Note that unlike the robustness exercises in Section 4.1, the examples in this subsection retrain AlphaPortfolio under the modified objective or friction specification. In the first example, we consider the simple situation that AP's portfolio size is kept constant over time. To account for transaction costs and price impacts, we introduce a transaction fee of 10-100 basis points of the transaction amount. Note that in training the model, we allow the action (trading) to affect the state (return of assets after fees). The results are reported in Table 11. The cost consideration brings down the strategy's turnover, but OOS Sharpe ratios and other performance metrics remain comparable with the baseline.

In the second example, in addition to considering price impacts and transaction costs, we incorporate dynamic budget considerations. All returns are reinvested, which implies that the AP strategy dynamically changes with the portfolio size, which affects the trading costs and returns. This portfolio size limit is part of the market environment (it affects investment opportunities in practice as well), and interacts with the trading actions.

In order to make the trading costs under varying portfolio scale realistic, we adopt in the second example the model and parameter estimates from Frazzini, Israel, and Moskowitz (2018). For illustration, we take their trading cost model with input variables, "Beta * IndexRet * buysell," "Time trend," "Log of ME," "Fraction of daily volume," and "sqrt(Fraction of daily volume)." We examine the long-only AP with an initial budget of $10 million USD and retrain the model. The results are reported in Table 12. Compared with Table 7, it still achieves an OOS Sharpe ratio above 1.5 and other performance metrics comparable with the

Table 11: Out-of-Sample Performance of AlphaPortfolio Considering Trading Costs

This table reports Performance of portfolios of long/short stocks in the highest/lowest decile of desirability scores from 1990 to 2016, where (1)-(3) set transaction cost rate as 0.001 and (4)-(6) set transaction cost rate as 0.01. Within (1)-(3) and within (4)-(6), the first columns present the performance on the whole sample. The remaining four columns present performance on subsamples excluding microcap firms in the smallest decile and quintile, respectively. q_n symbolizes the n^{th} NYSE size percentile. Return, Std.Dev, and Sharpe are all annualized.

	Transaction Cost Rate 0.001			Transaction Cost Rate 0.01		
	(1)	(2)	(3)	(4)	(5)	(6)
Firms	All	Size $> q_{10}$	Size $> q_{20}$	All	Size $> q_{10}$	Size $> q_{20}$
Return(%)	15.73	14.04	14.92	16.59	16.33	15.44
Std.Dev.(%)	7.77	5.90	7.66	9.85	6.96	7.82
Sharpe	2.02	2.38	1.95	1.68	2.35	1.97
Skewness	2.04	1.24	2.53	2.08	1.49	2.36
Kurtosis	11.26	4.54	11.83	10.73	4.33	10.28
Turnover	0.20	0.23	0.25	0.16	0.19	0.20
MDD	0.03	0.02	0.02	0.06	0.02	0.02

Table 12: Out-of-Sample Performance of AP Considering Portfolio Scale and Price Impact

This table reports Performance of portfolios of long stocks in the highest slice of desirability scores from 1990 to 2016 with a budget constraint. 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 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		$> q_{10}$		$> q_{20}$	
					$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	36.70	43.57	38.36	CAPM	26.66***	0.347	32.17***	0.362	26.89***	0.401
Std.Dev.(%)	24.36	28.38	27.20	FFC	27.58***	0.626	30.97***	0.630	25.74***	0.655
Sharpe	1.51	1.54	1.41	FFC+PS	26.82***	0.626	28.64***	0.634	22.75***	0.661
Skewness	0.81	2.68	2.59	FF5	26.72***	0.570	27.81***	0.587	21.73***	0.217
Kurtosis	3.65	19.81	17.57	FF6	29.56***	0.633	30.48***	0.628	24.30***	0.658
Turnover	0.17	0.21	0.24	SY	33.72***	0.565	33.80***	0.557	27.09***	0.587
MDD	0.22	0.28	0.27	Q4	32.16***	0.600	32.77***	0.541	26.76***	0.556

baseline AP. It is worth mentioning that the turnover is almost half of the baseline AP once we take transaction cost and budget dynamics into consideration.

In the third example, we adopt a portfolio performance metric incorporating the delegated nature of investment management. We set the management objective to be the expected

compensation over multiple years for managing a long-only portfolio. The compensation consists of a management fee of 0.5% of the assets under management (AUM) and a carried interest of 10% on excess returns. The management fee and carry vary quite a bit in practice but the compensation structure is realistic (e.g., Ma, Tang, and Gomez, 2019). For a simple illustration, we set the career length of the manager to be four years and the return benchmark to be zero. For a fund with an initial AUM of $10 million USD with all after-fee proceeds reinvested, the manager earns on average a cumulative compensation of $2,094,850 USD over four years, with a standard deviation of $556,630 USD.[21] Table 13 reports other metrics of the AP performance. The lack of concern for volatility means AP focuses on high cumulative returns at the expense of Sharpe ratio. AP also seems to pick small and illiquid stocks, the exclusion of which reduces the Sharpe ratio by about a quarter.

Table 13: Out-of-Sample Performance under Compensation-Based Objective

This table reports Performance of portfolios of long/short stocks in the highest/lowest decile of desirability-scores from 1990 to 2016. The objective is set as manager's compensation. During each step in training, AP considers a 4-year trading period. Each year the compensation is calculated as $AUM * 0.5\% + Positive\_return * 0.1$. 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 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(%)	42.85	44.69	44.94	CAPM	38.96***	0.008	37.41***	0.089	37.11***	0.075
Std.Dev.(%)	23.92	29.76	36.48	FFC	40.98***	0.104	37.88***	0.205	37.37 ***	0.145
Sharpe	1.79	1.50	1.23	FFC+PS	40.89***	0.104	37.70***	0.205	36.58***	0.146
Skewness	3.24	8.67	11.0	FF5	43.70***	0.166	41.09 ***	0.240	40.91***	0.175
Kurtosis	19.01	112.25	157.99	FF6	44.95***	0.179	41.90***	0.244	41.48***	0.176
Turnover	0.18	0.19	0.20	SY	47.93***	0.091	43.73***	0.189	42.66***	0.139
MDD	0.07	0.04	0.06	Q4	46.04***	0.162	43.41***	0.229	43.19***	0.161

In addition to the alternative objectives considered above, we further evaluate performance under a Constant Relative Risk Aversion (CRRA) objective. Specifically, we compare PPP, our AlphaPortfolio without RREM, and the full AlphaPortfolio in Table 14.

The CRRA objective, which is used in the PPP framework, provides a relevant bench-

[21] We have 27 years in the test sample and we extrapolate the last three years into a four-year cycle.

Table 14: Out-of-Sample Performance of PPP, AlphaPortfolio without RREM and AlphaPortfolio with CRRA Objective

This table reports Performance of parametric portfolio policies (PPP), AlphaPortfolio without RREM (Ex-RREM) and full AlphaPortfolio (AP) using CRRA utility as its training objective, focusing on long/short stocks in the highest/lowest decile of desirability-scores from 1990 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 (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). The first 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.

	Portfolio Performance				Portfolio Excess Alpha					
	(1)	(2)	(3)		(4)	(5)	(6)	(7)	(8)	(9)
Firms	All	$> q_{10}$	$> q_{20}$	Factor Models	All $\alpha(\%)$	R^2	$> q_{10}$ $\alpha(\%)$	R^2	$> q_{20}$ $\alpha(\%)$	R^2
Panel A: Performance of Parametric Portfolio Policy										
Return(%)	16.36	11.17	9.08	CAPM	9.50***	0.359	4.61	0.361	2.37	0.432
Std.Dev.(%)	22.83	21.75	20.35	FFC	8.48***	0.623	3.70	0.679	1.60	0.773
Sharpe	0.72	0.51	0.45	FFC+PS	7.35**	0.625	2.95	0.680	0.66	0.774
Skewness	1.33	0.48	0.22	FF5	5.61*	0.584	-0.57	0.622	-3.04	0.709
Kurtosis	8.41	5.42	4.55	FF6	8.06***	0.634	2.18	0.691	-0.28	0.788
Turnover	0.11	0.12	0.11	SY	5.70*	0.642	1.07	0.623	0.76	0.622
MDD	0.51	0.43	0.40	Q4	6.78*	0.440	1.32	0.464	-0.98	0.543
Panel B: Performance of AlphaPortfolio without RREM										
Return(%)	10.42	9.12	6.99	CAPM	10.17***	0.005	8.29***	0.074	6.32***	0.072
Std.Dev.(%)	6.86	6.12	5.0	FFC	10.94***	0.124	8.73***	0.404	6.57***	0.386
Sharpe	1.52	1.49	1.4	FFC+PS	10.93***	0.124	7.87***	0.416	5.75***	0.402
Skewness	1.38	1.9	1.92	FF5	11.13***	0.182	8.08***	0.331	5.84***	0.325
Kurtosis	4.87	8.83	8.61	FF6	11.87***	0.232	8.92***	0.414	6.52***	0.405
Turnover	0.28	0.28	0.28	SY	11.19***	0.16	9.35***	0.38	6.6***	0.364
MDD	0.06	0.04	0.04	Q4	12.34***	0.25	9.07***	0.394	6.59***	0.383
Panel C: Performance of AlphaPortfolio										
Return(%)	22.47	24.04	51.0	CAPM	21.07***	0.044	20.99***	0.144	47.14***	0.057
Std.Dev.(%)	13.17	16.05	32.08	FFC	21.86***	0.121	21.9***	0.334	47.53***	0.141
Sharpe	1.71	1.5	1.59	FFC+PS	22.41***	0.122	21.41***	0.334	46.92***	0.141
Skewness	3.09	3.92	8.65	FF5	24.07***	0.299	25.12***	0.49	50.12***	0.181
Kurtosis	19.7	30.44	106.4	FF6	24.86***	0.314	25.93***	0.501	50.83***	0.183
Turnover	0.24	0.26	0.32	SY	21.94***	0.158	25.49***	0.361	52.52***	0.153
MDD	0.05	0.06	0.04	Q4	25.8***	0.308	26.45***	0.47	52.14***	0.177

mark for assessing portfolio strategies under risk-averse preferences. Our findings show that AlphaPortfolio is robust to the choice of optimization objective: even when trained under the CRRA criterion, it achieves consistently strong portfolio returns. Compared with directly optimizing for the Sharpe ratio, however, the Sharpe performance of AP decreases somewhat, reflecting the trade-off imposed by CRRA. Among the alternatives, PPP produces the weakest results, AP without RREM shows improvements, while the full AlphaPortfolio achieves the best overall performance.

Our goal is not to provide an exhaustive list of AlphaPortfolio objectives or market interactions. Instead, our illustrations serve to demonstrate the flexibility and versatility of the AP framework. Depending on the specific case in investment advising or trading, the objective and market environment can be correspondingly specified and incorporated into the training process. For one, AP allows for integration of transaction costs into the training process. This feature matters because transaction costs are not just an add-on to portfolio implementation; they also change which predictive signals are economically useful. A model trained without cost awareness may continue to favor high-turnover signals even when those signals generate little net value after trading frictions. By incorporating transaction costs directly into training, AlphaPortfolio can instead learn portfolio policies and underlying score dynamics that are better aligned with net performance, not merely gross predictive fit.

Note that to accommodate these market interactions and flexible management objectives under supervised learning, one has to label the data using a theoretical model or proxies based on historical observations, which may be unavailable, computationally expensive, or yield poor performance. For some objectives, without theoretical guidance or gross approximation, it is unclear what the historically "correct" action is when we train the model using supervised learning. RL, in contrast, is a framework that has been proven in the AI field to perform well in solving such problems.

4.3 Action-State Interactions: Partial-Equilibrium Illustrations

It is useful to distinguish two different notions of interacting with the market/system states. The first is the standard sequential-control setting in reinforcement learning, where current actions affect future rewards and possibly future portfolio-specific states, while the broader market environment is treated as exogenous. The second is a stronger feedback setting in which the decision maker's actions alter the evolution of future market states

themselves, such as price dynamics, liquidity, or investor confidence. The latter is closer to an equilibrium or market-impact problem and requires an explicit model for how actions feed back into the law of motion of the environment.

Our baseline empirical implementation of AlphaPortfolio belongs to the first category and should therefore be interpreted as a partial-equilibrium, price-taking framework with respect to asset-price dynamics. There, we use historical data to proxy for the market environment and do not claim to identify how the strategy's own trades would endogenously move future asset prices. Put differently, aside from explicitly modeled implementation frictions or manager-specific state variables, the return process is treated as exogenous to the portfolio manager's actions. This assumption is standard in much of the empirical portfolio-choice and asset-pricing literature and is most defensible when the strategy capacity or investment size is small relative to the market or when many heterogeneous investors with different objectives trade simultaneously.

At the same time, this clarification does not mean that AlphaPortfolio only accommodates static or myopic objectives. Even under a partial-equilibrium view for prices, the framework can still accommodate economically meaningful action-state interactions at the fund or manager level. Current actions can affect future portfolio-specific states and continuation values through turnover, implementation costs, dynamic budget constraints, assets under management, delegated-compensation terms, or fund survival. These are genuine state interactions in the manager's dynamic problem, even though they do not amount to a full endogenous equilibrium for market prices.

This distinction also helps clarify the relation to recent direct-construction approaches. Deep parametric portfolio policy models such as Simon, Weibels, and Zimmermann (2025) directly learn a nonlinear mapping from characteristics to portfolio weights and incorporate utility, transaction costs, and portfolio-weight restrictions, but they do not model an action-dependent law of motion for future market states. Integrated end-to-end Markowitz frameworks such as Wang, Gao, Harvey, Liu, and Tao (2026) embed exact, differentiable convex optimization layers into the neural networks. Such models likewise solve a sequence of constrained portfolio-design problems under exogenous return dynamics rather than a market environment whose transition law is altered by the investor's actions; nevertheless, they guarantee that the generated weights can strictly satisfy most complex economic constraints and portfolio structures, ensuring that heterogeneous investor preferences are met and can demonstrate theoretically how the end-to-end approach excels when downstream

tasks become complex. Similarly, Jensen, Kelly, Malamud, and Pedersen (2024) integrate trading-cost-aware portfolio optimization with machine learning and learn directly from an economic objective, yet the resulting framework remains centered on implementable portfolio choice under exogenous market dynamics. Thus, these recent studies are best understood as powerful advances in direct portfolio construction and constrained optimization, but not as models of endogenous market-feedback state transitions.

By contrast, the reinforcement-learning formulation underlying AlphaPortfolio is naturally compatible with action-dependent state dynamics once an explicit environment module is specified. In the general framework introduced in Section 2, the transition law $P(\boldsymbol{Z}_{t+1} \mid \boldsymbol{Z}_t, \boldsymbol{a}_t)$ allows the investor's actions to affect future states. In the baseline empirical implementation of this paper, we adopt a partial-equilibrium, price-taking view with respect to asset-price dynamics and therefore treat the return process as exogenous. However, the framework itself does not require this restriction.[22]

While a full market-feedback version of AlphaPortfolio would require an explicit environment module (e.g., one governing endogenous price impact, liquidity depletion, or strategic interactions with other market participants), and is beyond the scope of the current paper, we can still illustrate a richer, reduced-form action-state interaction through the setting of delegated investment where fund survival hinges on investor confidence coming from historical investment performance.

Specifically, suppose investors' wealth evolve according to $W_{t+1} = W_t\big(1 + R^p_{t+1}\big)$, where R^p_{t+1} is the realized portfolio return net of implementation costs. Define the stopping time

$$\tau^{\text{fail}} = \inf\left\{t \geq 1 : \frac{W_t}{W_0} \leq \bar{b}\right\},$$

where $\bar{b} \in (0, 1)$ is a failure threshold. In our illustration, $\bar{b} = 0.5$, corresponding to a 50% drawdown from initial investment value over the evaluation window. We then define the fund-survival objective as $H(\tau) = \sum_{t=1}^{T} \mathbf{1}\{t < \tau^{\text{fail}}\} R^p_{t+1}$, or, equivalently, as the cumulative return over the horizon with zero continuation value once the fund fails. This is a reduced-form way to capture the idea that sufficiently poor performance can trigger investor redemptions, loss of confidence, or shutdown of the fund.

This survival formulation differs in nature from simple transaction-cost adjustment. In

[22]Related work such as Campello, Cong, and Zhou (2022) illustrates how a corporate decision module similarly built using RL can be combined with a learned environment module for rich state-action interactions.

the latter, current trading affects contemporaneous net returns but does not alter whether future decisions remain feasible. In the present specification, current actions affect a state variable that governs the continuation of the investment process itself. The resulting problem is therefore genuinely path dependent: an aggressive action that raises current expected return may also increase the probability of investor confidence loss and fund termination.

Table 15 reports the out-of-sample performance of this "AP Fund" specification. The objective is the cumulative return over a 12-month window, with a transaction cost rate of 10 basis points and fund failure triggered by a 50% loss during that window. We interpret this exercise as an initial reduced-form exploration of manager-level state interactions. The point is not that we have solved for a full general-equilibrium feedback from trades to future prices. Rather, the example shows that AlphaPortfolio can accommodate dynamic interactions in which current portfolio choices affect future feasible states and continuation payoffs relevant to delegated portfolio management.

In this sense, AlphaPortfolio sits between two extremes. On the one hand, the baseline empirical implementation is partial equilibrium and price taking with respect to asset prices. On the other hand, the underlying RL framework is flexible enough to incorporate action-dependent environment dynamics once an explicit environment module is available. This feature is less natural in static direct-mapping or optimization-layer approaches, which are designed primarily to solve constrained portfolio choice problems conditional on exogenous market inputs rather than to learn in an action-dependent environment.

4.4 Practical Implementations in the Financial Industry

We next evaluate AlphaPortfolio under implementation settings commonly encountered in institutional portfolio management. In practice, portfolio mandates are shaped by considerations such as market capacity, liquidity, benchmark hedging, and short-selling restrictions rather than by a single theoretical objective such as the Sharpe ratio or CRRA utility. To assess the practical relevance of AlphaPortfolio, we consider several realistic specifications: (i) restricting the investment universe to the largest (1000, 2000, or 3000) capitalization stocks, effectively mimicking selecting stocks from Russell 1000, 2000, or 3000 stocks; (ii) training the model to maximize cumulative return, (iii) imposing short-sell constraints (long-only); (iv) hedging the portfolio using market indices (e.g., SPY); and (v) applying explicit liquidity and price screens.

Table 15: Out-of-Sample Performance of an AP Fund

This table reports Performance of a long-short portfolio taking long positions in assets with the highest 10% desirability scores from 1990 to 2016, and short positions in assets with the lowest 10% desirability scores. The objective is set as the cumulative return over a 12-month window and transaction cost rate is set as 0.1%. During training, AP will stop trading upon incurring a 50% loss in the 12-month window. 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 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. Returns and Sharpe ratios are annualized. 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		$> q_{10}$		$> q_{20}$	
					$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	26.61	24.82	24.01	CAPM	22.12***	0.047	18.95***	0.150	18.20***	0.168
Std.Dev.(%)	15.35	15.68	14.59	FFC	23.80***	0.243	19.76***	0.433	18.92***	0.487
Sharpe	1.73	1.58	1.65	FFC+PS	24.35***	0.243	18.65***	0.436	17.27***	0.493
Skewness	2.26	3.71	3.01	FF5	26.56***	0.358	22.26***	0.489	21.53***	0.548
Kurtosis	11.98	28.57	17.14	FF6	27.64***	0.381	23.01***	0.500	22.08***	0.555
Turnover	0.19	0.22	0.23	SY	30.14***	0.227	24.26***	0.384	22.95***	0.429
MDD	0.09	0.06	0.06	Q4	28.54***	0.366	23.88***	0.480	22.99***	0.530

Table 16 summarizes how AlphaPortfolio consistently generates strong risk-adjusted OOS performance across all specifications. Restricting the universe to the largest stocks by market capitalization does not materially weaken performance: Sharpe ratios remain between 1.57 and 2.00 across the top-1,000, top-2,000, and top-3,000 stock universes.

When the model is instead trained to maximize cumulative return, realized returns increase substantially while Sharpe ratios remain above one across all large-cap universes. This flexibility is practically useful because many investment mandates emphasize total return or capital growth more directly than mean-variance efficiency.

AlphaPortfolio also performs well under a long-only mandate, with Sharpe ratios ranging from 1.45 to 1.65 across the large-cap universes. This result suggests that the strategy's performance does not rely on short-selling and can be implemented in investment vehicles such as mutual funds and pension mandates where shorting may be restricted.

We further examine a benchmark-hedged implementation in which the strategy is long the AlphaPortfolio large-cap portfolio while shorting SPY to reduce market exposure. Performance remains strong, with Sharpe ratios between 1.54 and 1.82, indicating that the strategy generates economically meaningful excess returns beyond simple market exposure.

Table 16: AlphaPortfolio under implementation-oriented specifications

Specification	Universe	Return	Std.Dev.	Sharpe	Turnover	MDD	CAPM α	FFC α	FF6 α	Q4 α
Largest stocks (long–short)	Top 1,000	12.64	8.06	1.57	0.26	0.03	11.30***	10.84***	11.57***	12.00***
	Top 2,000	15.87	8.71	1.82	0.24	0.03	14.61***	14.43***	15.42***	15.86***
	Top 3,000	18.17	9.09	2.00	0.23	0.02	17.02***	16.84***	17.58***	18.06***
Cumulative-return objective	Top 1,000	15.95	12.36	1.29	0.28	0.06	13.10***	14.72***	17.77***	18.10***
	Top 2,000	24.85	15.99	1.55	0.25	0.06	21.54***	22.37***	26.34***	26.92***
	Top 3,000	32.03	17.93	1.79	0.24	0.05	28.83***	30.16***	34.62***	35.31***
Long-only mandate	Top 1,000	36.09	24.93	1.45	0.24	0.30	27.11***	26.02***	25.55***	26.22***
	Top 2,000	46.10	31.82	1.45	0.24	0.24	36.11***	34.65***	35.53***	36.51***
	Top 3,000	51.41	31.14	1.65	0.24	0.27	41.97***	41.44***	43.00***	43.44***
Long large-cap, short SPY	Top 1,000	16.02	10.38	1.54	0.23	0.06	14.54***	14.39***	15.15***	15.69***
	Top 2,000	26.03	15.52	1.68	0.25	0.07	23.55***	23.54***	25.46***	26.13***
	Top 3,000	29.08	15.98	1.82	0.24	0.04	27.01***	27.18***	29.22***	29.81***
Liquidity and price screens	Long–short	21.32	14.93	1.43	0.28	0.15	18.86***	20.83***	20.07***	21.03***
	Long-only	30.70	22.03	1.39	0.26	0.20	24.40***	26.47***	25.31***	26.61***

Notes: This table summarizes the out-of-sample performance of AlphaPortfolio under implementation settings commonly encountered in institutional portfolio management. Returns, standard deviations, and Sharpe ratios are annualized. The first four specifications consider the largest 1,000, 2,000, and 3,000 stocks by market capitalization during 1990–2016. The final specification imposes liquidity and price screens, restricting the universe to the largest 1,000 stocks with average daily dollar trading volume above $5 million and prices between $3 and $5,000 during 2007–2016. Factor-adjusted alphas are reported relative to the CAPM, the Fama–French–Carhart four-factor model (FFC), the Fama–French six-factor model (FF6), and the Hou–Xue–Zhang q-factor model (Q4). Detailed results for additional factor models are reported in Appendix C.

Finally, we impose explicit execution screens by restricting the universe to the largest 1,000 stocks with average daily dollar volume above $5 million and prices between $3 and $5,000. Both long-short and long-only implementations remain economically meaningful under these liquidity and price constraints.

Overall, these results indicate that AlphaPortfolio remains robust across implementation settings closely aligned with real-world portfolio management. Detailed factor-adjusted alpha results for each specification are reported in a series of tables in Appendix C.

5 Economic Distillation and Interpretation

A portfolio manager not only needs strong performance, but also a coherent economic explanation of what the model is learning and why it works. This is especially important for a complex deep-learning model, AlphaPortfolio included, both for scientific interpretation and for communication with investors and risk committees. We therefore adopt an "economic distillation" procedure that projects AlphaPortfolio onto a simpler modeling space and uses that projection to identify the variables, nonlinearities, and time-varying patterns most relevant to its portfolio decisions.[23] The distilled model "represents" or mimics the complex AI model and therefore can reveal important properties of the original model and help with its economic interpretation. While such an exercise does not illuminate any underlying mechanisms or causal links, it reveals useful patterns that both theorists and empiricists can explore for better economic understanding in future.

Specifically, we project AlphaPortfolio into the modeling space of sparse linear regression models using Polynomial-Sensitivity-Analysis (Cong, Li, and Wang, 2024). We first summarize the mapping from lagged asset features and cross-sectional context to AlphaPortfolio's desirability scores in the panel Transformer, consisting of self-attention-based SREM and RREM. We then examine the marginal contribution of each feature and inspect its comparative statics when other features change. The procedure allows us to identify the variables (or their higher-order terms or interactions) that matter the most in the model. Next, we use these variables and their higher-order and interaction terms as input variables to estimate

[23]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 AP holdings with the topics discussed in some relevant text documents, we may be able to develop a narrative or a better understanding of the model.(Cong, Liang, Yang, and Zhang, 2021; Cong, Liang, Zhang, and Zhu, 2025) do this using AlphaPortfolio as an example.

a Lasso regression model. We set penalty parameters such that 50-60 inputs are selected, which is comparable to the number of input time series in AlphaPortfolio.

5.1 Polynomial Sensitivity Analysis

To interpret AlphaPortfolio, we first measure the sensitivity of the model's desirability scores to each input characteristic. Let the panel Transformer be written as $s_{i,t} = \text{PTF}(\boldsymbol{Z}_i^{(t)}, \boldsymbol{\mathcal{Z}}_{-i}^{(t)})$, where $s_{i,t}$ denotes the score of asset i in month t of the trading trajectory, and $\boldsymbol{Z}_i^{(t)} = (\boldsymbol{z}_{i,t-K}, \ldots, \boldsymbol{z}_{i,t})$ denotes the historical feature sequence of asset i. Moreover, $\boldsymbol{\mathcal{Z}}_{-i}^{(t)}$ denotes the historical feature sequences of the other assets. Let $z_{q,k,i}^{(t)}$ denote the (q,k)-th element of $\boldsymbol{Z}_i^{(t)}$, that is, the value of characteristic q for asset i at lag k. The local sensitivity of the score with respect to this characteristic is defined as:[24]

$$\delta_{q,k,i}^{(t)} = \frac{\partial s_{i,t}}{\partial z_{q,k,i}^{(t)}}. \tag{21}$$

In our implementation, these derivatives are computed automatically using the `autograd` module in PyTorch. To measure the overall importance of each characteristic, we average the absolute sensitivities across all asset-month observations:

$$\bar{\delta}_q = \frac{1}{K \times \sum_{t=1}^{T} I_t} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{i=1}^{I_t} \left| \frac{\partial s_{i,t}}{\partial z_{q,k,i}^{(t)}} \right|, \tag{22}$$

where I_t denotes the number of available assets in the t-th month of the trading trajectory. Characteristics with larger values of $\bar{\delta}_q$ therefore exert greater influence on AlphaPortfolio's desirability scores.

Using these sensitivity measures, we identify the most influential characteristics and generate polynomial transformations of those variables. Specifically, we select the top $K\%$ most influential characteristics and construct polynomial terms up to degree p, including nonlinear transformations and interaction terms. These candidate features are then used as inputs in a sparse Lasso regression that approximates the asset desirability scores.

The resulting distilled model provides a simplified representation of the original deep learning system. In each out-of-sample month, we regress AlphaPortfolio's scores on the selected polynomial terms and use the fitted coefficients to reconstruct an approximate score

[24] Because gradient magnitudes are scale-dependent, sensitivity comparisons should be interpreted relative to the standardized inputs used in the empirical implementation.

for each asset. These reconstructed scores are then used to form portfolios using the same long–short construction as the original AlphaPortfolio strategy.

The purpose of this procedure is not to replace AlphaPortfolio with a simpler trading rule. Rather, it allows us to identify the economic variables, nonlinearities, and interactions that account for a substantial portion of the model's portfolio decisions. Because the distillation relies on information learned by the trained AlphaPortfolio model, the simplified model typically underperforms the full model. Nevertheless, if the distilled approximation retains meaningful portfolio performance, it suggests that a relatively small set of economically interpretable features explains much of what the full model is doing. Algorithm 1 in the appendix summarizes the distillation procedure.

5.2 Important Features and Dynamic Patterns

For readability, we refer to the variables in economic terms throughout the discussion. In particular, `ivc` denotes inventory change, `ipm` denotes pretax profit margin, `delta_so` denotes change in shares outstanding, `Idol_vol` denotes idiosyncratic volatility, `Ret_max` denotes the maximum daily return within a month, `C` denotes cash and short-term investments scaled by total assets, and `Q` denotes Tobin's Q.

We then aggregate the month-by-month distillation results using Fama–MacBeth regressions. For parsimony, Table 17 reports the top 50 dominant features in the degree-2 specification together with their t-statistics.

To investigate the time-varying effects of dominant characteristics, we plot heatmaps in Figure 4 for the top 15 terms in degree-2 polynomial functions to highlight their rankings in the OOS periods. We also outline basic statistics for both degree-2 and degree-1 polynomials in Table 18. 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%).[25] For degree-1 polynomials across Rank 1, 2, and 3, at the top again are ivc (97.8%), Idol_vol (43.2%), and ipm (21.9%). The other five important characteristics (free_cf, Ret_max, ret, delta_so, and C) account for 13.9% to 19.1% each.

From Figure 4 and Tables 17 and 18, we find that a relatively small set of stock features accounts for a substantial share of the distilled approximation to AlphaPortfolio's portfolio decisions. For example, the inventory change (ivc) plays a key role in our algorithm with

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

Table 17: T-statistics of Selected Features for Algorithm 1 with Polynomial of Degree Two

This table presents the results where Fama-MacBeth 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. For each sample, the table reports the 50 polynomial terms with the largest absolute Fama–MacBeth t-statistics.

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

a probability of more than 80% included in the top contributing factors in both degree-1 and 2 polynomials. Thomas and Zhang (2002) first document that ivc can negatively predict stocks' future returns, which is consistent with earnings management of firms. The persistent importance of inventory change suggests that this signal remains economically relevant for AP's portfolio decisions even in the post-2002 period. Short-horizon return variables, such as `ret_11` and `ret_10`, tend to enter with negative signs and large absolute t-statistics, especially in the large-stock subsamples, which is consistent with short-term reversal patterns documented in the literature (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.[26]

More generally, Table 17 shows that AlphaPortfolio depends on a richer lag structure than is typically assumed in standard characteristic-based portfolio studies. The recurrence of the same variables at multiple lag positions indicates that the model uses the time profile of signals, not just their current level. This is consistent with the sequence-modeling architecture in Section 2 and provides empirical evidence that longer lag structures can be economically relevant for direct portfolio construction.

Other prominent signals include Tobin's Q, pretax profit margin, cash and short-term investments, idiosyncratic volatility, maximum daily return within the month, free cash flow, and issuance-related variables such as change in shares outstanding. Some of these variables correspond to trading-related or mispricing-related signals, while others reflect firms' fundamentals and financing conditions. In that sense, the distilled AlphaPortfolio combines both market-based and firm-based information. This distinction is consistent with the idea that trading-related signals often operate through mispricing channels, whereas fundamental variables are more closely tied to risk and investment dynamics (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' nonlinear 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_volˆ2, Ret_max, ret) and financial (Q, C, Cˆ2, delta_so, ivc, ipmˆ2, ivcˆ2, investment, free_cf) signals. Table

[26] 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 4: The following three heatmaps illustrate how the ranking of dominant features changes 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), Pretax 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 provides a detailed description of the features.

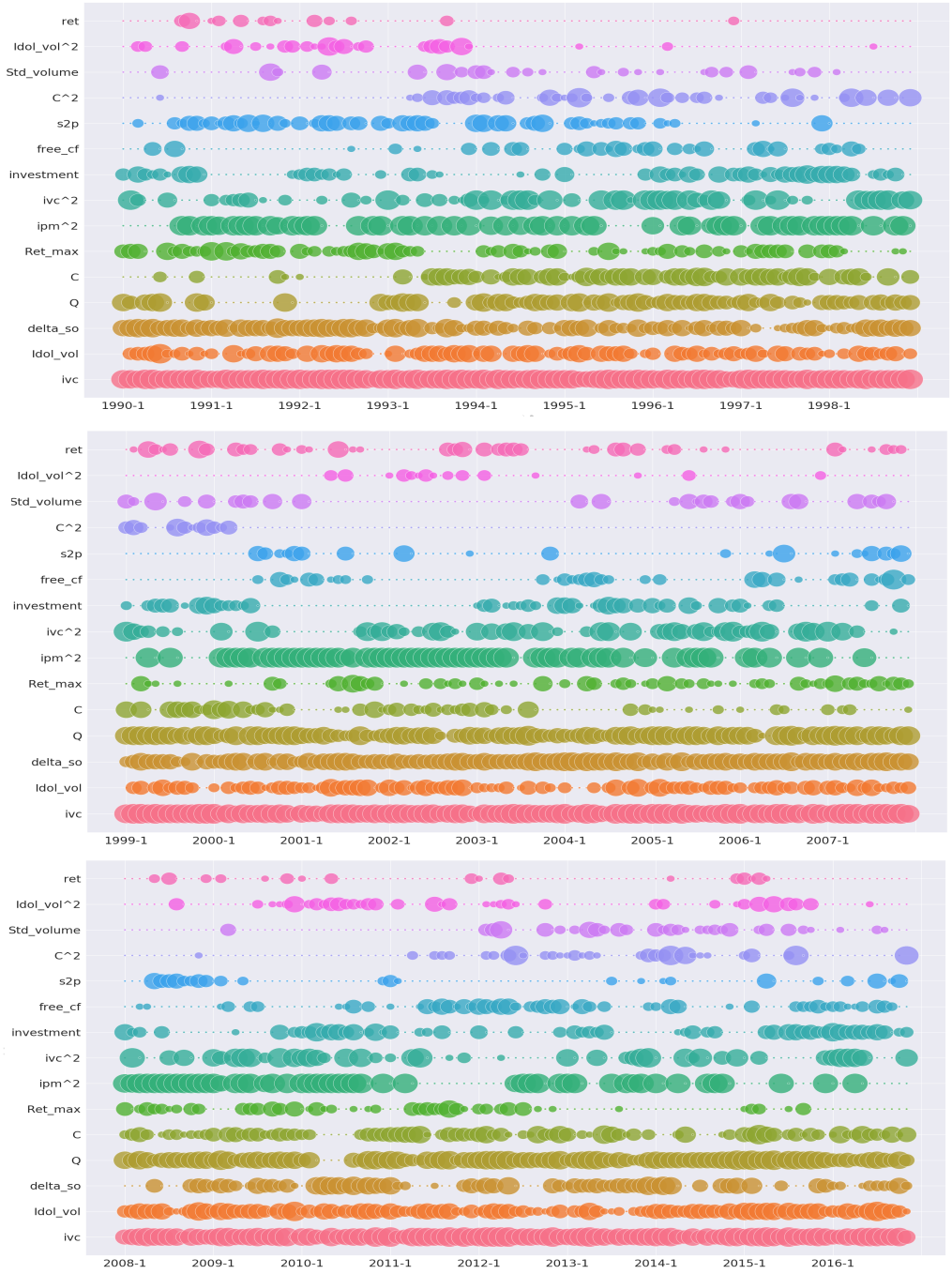

Table 18: Dominant Features after Economic Distillation

This table presents the probability of features ranked in top three across different months in the distillation Algorithm 1. In each month of the OOS periods, the AlphaPortfolio is distilled into a second-degree polynomial function (Panel A) and a first-degree polynomial function (Panel B). The 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%	ipmˆ2	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%

19 reveals that inventory changes (ivc) and pretax profit margin to the second power (ipmˆ2) take turns to play important roles in AlphaPortfolio construction and so do corporate liquidity (C, Cˆ2) 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.

5.3 Further Discussion

Distillation on the training set. Algorithm 2 shown next describes an alternative distillation exercise. While Algorithm 1 mimics AlphaPortfolio concerning OOS tests, Algorithm 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 Algorithm 1 and omit the details. Both algorithms can be extended to capture persistent latent variables in future work.

Table 19: 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 pairs 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_volˆ2, Ret_max, ret}, and financial subset contains {Q, C, Cˆ2, delta_so, ivc, ipmˆ2, ivcˆ2, investment, free_cf}.

Term 1	Term 2	Correlation	P-value
Panel A: Pairwise Correlation of Dominant Features			
ivc	ipmˆ2	-0.38	1.55×10^{-12}
delta_so	Cˆ2	-0.33	1.40×10^{-12}
C	delta_so	-0.31	2.09×10^{-8}
Cˆ2	ret	-0.26	1.47×10^{-6}
investment	s2p	-0.25	4.20×10^{-6}
Idol_vol	ipmˆ2	-0.24	1.22×10^{-5}
ipmˆ2	Cˆ2	-0.23	3.79×10^{-5}
Ret_max	Cˆ2	-0.22	8.43×10^{-5}
C	ret	-0.21	1.80×10^{-4}
delta_so	Idol_volˆ2	-0.2	2.90×10^{-4}
Panel B: Trading Signals vs. Financial Signals			
Trading_related	Financial_related	-0.33	8.00×10^{-10}

Economic distillations: the case of LSTM-GAT. Overall, economic distillation provides 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 economic environment changes.[27] 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 misspecification or coding error.

Appendix A4 reports the performance of an AP model based on LSTM and GAT. On the positive side, the LSTM-GAT model achieves an OOS Sharpe ratio of around 3 after excluding small-cap stocks. This confirms the superior performance of the AP framework and even outperforms the panel Transformer-based AP. On the negative side, the economic distillation reveals that LSTM-GAT does not have stable utilization of input features and suffers from exploding gradient issues, suggesting that the trained model goes to some ex-

[27] In our distillation exercises, nonlinear univariate terms appear more persistently than interaction terms. But this is not at odds with studies that emphasize interaction effects because those studies focus on estimating SDF or minimizing estimation errors in assets' return moments, not portfolio performance.

tremes and such RNN-like models may not be the most suitable.[28]

Overall, the distillation results suggest that AlphaPortfolio is not an arbitrary black box. A relatively small and economically interpretable set of signals, particularly inventory change, profitability-related terms, issuance-related variables, cash and liquidity variables, and selected trading-friction measures, accounts for a substantial portion of the portfolio decisions. At the same time, the importance of these signals rotates over time, which is consistent with the state-dependent and path-dependent structure emphasized earlier.

6 Conclusion

We develop AlphaPortfolio as a new solution to the general portfolio management problem, demonstrating how sequence modeling, cross-asset relation modeling, and reinforcement learning extend the literature by enabling end-to-end portfolio optimization for general management objectives over enlarged policy spaces. Empirically, AlphaPortfolio delivers strong out-of-sample performance in U.S. equities, even under economic and trading restrictions.

Three aspects of our findings deserve emphasis. First, while the end-to-end principle is shared with PPP, combining it with a sufficiently rich policy class yields dramatically larger gains. The enlarged policy space is disciplined not by parsimony but by the economic objective itself, which implicitly regularizes the model by directing capacity toward the most consequential signals. Second, the framework enlarges not only the policy space but also the objective space, enabling trajectory-level criteria not additively separable and therefore difficult to optimize within conventional frameworks. Third, the distillation analysis reveals that the optimal mapping from states to portfolio weights involves a richer lag structure than standard practice assumes, suggesting that conventional single- or short-period-lag portfolio sorts may leave economically relevant information on the table.

Admittedly, future implementations may replace gradient-based optimization with tree-based search, substitute the Transformer with more efficient models, or adopt entirely different algorithms. That said, the specific architectural choices introduced here—involving adapting the Transformer and attention mechanisms to financial panel data, and using rein-

[28]This is consistent with the fact that the machine-learning literature typically addresses vanishing and exploding gradients within the training sample, while such issues may reappear out of sample or in live deployment. Appendix A4 thus 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.

forcement learning for direct portfolio construction—have already catalyzed a growing body of subsequent work treating portfolio management problems end-to-end and in a more data-driven manner. We view this as consistent with a natural layered view of the paper's contributions: the methods we introduced serve as an effective first-generation implementation of the broader framework, and the fact that they have been widely recognized suggests that the economic framing and method-robust insights they enabled are indeed durable, even as the specific implementations continue to evolve. The performance decomposition in Table 1 reinforces this view, showing that the gains arise primarily from the first two layers: direct construction as an end-to-end approach and effective use of sequential and cross-sectional information through enlarged model classes, rather than from fine architectural details.

More broadly, the paper shows that generative AI methods can contribute to finance when embedded in a well-defined economic decision problem rather than used as generic prediction tools. Our distillation results further suggest that AlphaPortfolio is not an arbitrary black box: a relatively small set of economically meaningful signals accounts for a substantial share of its portfolio decisions. Future research can extend this framework to richer environment dynamics, additional asset classes, and other (delegated) investment-management settings.

References

Aït-Sahalia, 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 investors' risk preferences via portfolio choices, *Journal of Financial Econometrics*.

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.

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

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.

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.

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

Buciluǎ, 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.

Butler, Andrew, and Roy H. Kwon, 2023, Integrating prediction in mean-variance portfolio optimization, *Quantitative Finance* 23, 429–452.

Campello, Murillo, Lin William Cong, and Luofeng Zhou, 2022, Alphamanager: A data-driven-robust-control approach to corporate finance, *Available at SSRN 4590323*.

Capponi, Agostino, 2022, Robo-advising: personalization and goals-based investing, *University of California Berkeley, Berkeley*.

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.

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.

Cong, Lin William, Guanhao Feng, Jingyu He, and Xin He, 2023, Growing the efficient frontier on panel trees, *NBER Working Paper*.

Cong, Lin William, Guanhao Feng, Jingyu He, and Junye Li, 2023, Sparse modeling under grouped heterogeneity with an application to asset pricing, *NBER Working Paper*.

Cong, Lin William, Guanhao Feng, Jingyu He, and Yuanzhi Wang, 2024, Mosaics of predictability, *Available at SSRN*.

Cong, Lin William, Zimeng Li, and Jingyuan Wang, 2024, Polynomial sensitivity analysis as a model-agonostic explainer with an application to credit risk assessment, *Working Paper*.

Cong, Lin William, Tengyuan Liang, Baozhong Yang, and Xiao Zhang, 2021, Analyzing textual information at scale, in *Information for efficient decision making: Big data, blockchain and relevance* . pp. 239–271 (World Scientific).

Cong, Lin William, Tengyuan Liang, Xiao Zhang, and Wu Zhu, 2025, Textual factors: A scalable, interpretable, and data-driven approach to analyzing unstructured information, *Management Science* 71, 10727–10739.

Cong, Lin William, Ke Tang, Jingyuan Wang, and Yang Zhang, 2019, Alphaportfolio: Direct construction through deep reinforcement learning and interpretable ai, *Permanent Working Paper*.

——— , 2020, Deep sequence modeling: Development and applications in asset pricing, *Journal of Financial Data Science* Forthcoming.

Das, 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.

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.

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.

Fan, Jianqing, Zheng Tracy Ke, Yuan Liao, and Andreas Neuhierl, 2022, Structural deep learning in conditional asset pricing, *Available at SSRN 4117882*.

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

Frazzini, Andrea, Ronen Israel, and Tobias J Moskowitz, 2018, Trading costs, *Available at SSRN 3229719*.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber, 2020, Dissecting characteristics nonparametrically, *The Review of Financial Studies* 33, 2326–2377.

Fu, Justin, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine, 2020, D4rl: Datasets for deep data-driven reinforcement learning, *arXiv preprint arXiv:2004.07219*.

Fujimoto, Scott, David Meger, and Doina Precup, 2019, Off-policy deep reinforcement learning without exploration, in *International Conference on Machine Learning* pp. 2052–2062. PMLR.

Gabaix, Xavier, Ralph SJ Koijen, Robert Richmond, and Motohiro Yogo, 2023, Asset embeddings, *Available at SSRN 4507511*.

Gârleanu, Nicolae, and Lasse Heje Pedersen, 2013, Dynamic trading with predictable returns and transaction costs, *The Journal of Finance* 68, 2309–2340.

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

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, 2020, Empirical asset pricing via machine learning, *The Review of Financial Studies* 33, 2223–2273.

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*.

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*.

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, 2019–2133.

Jagannathan, Ravi, and Tongshu Ma, 2003, Risk reduction in large portfolios: Why imposing the wrong constraints helps, *The Journal of Finance* 58, 1651–1683.

Jensen, Theis Ingerslev, Bryan T. Kelly, Semyon Malamud, and Lasse Heje Pedersen, 2024, Machine learning and the implementable efficient frontier, Discussion Paper, 22-63 Swiss Finance Institute Research Paper Working paper; later public versions circulated in 2024.

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).

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

Kelly, Bryan, Boris Kuznetsov, Semyon Malamud, Teng Andrea Xu, and Yuan Zhang, 2024, Large and deep factor models, arXiv preprint arXiv:2402.06635 Revised February 2026.

Kelly, Bryan T., Boris Kuznetsov, Semyon Malamud, and Teng Andrea Xu, 2025, Artificial intelligence asset pricing models, Discussion Paper, 33351 NBER Working Paper.

Kelly, Bryan T, Seth Pruitt, and Yinan Su, 2019, Characteristics are covariances: A unified model of risk and return, *Journal of Financial Economics* 134, 501–524.

Kelly, Bryan T, and Dacheng Xiu, 2021, Factor models, machine learning, and asset pricing, *Machine Learning, and Asset Pricing (October 15, 2021)*.

Kidambi, Rahul, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims, 2020, Morel: Model-based offline reinforcement learning, *arXiv preprint arXiv:2005.05951*.

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

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.

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, Yang, Guofu Zhou, and Yingzi Zhu, 2024, Maximizing the sharpe ratio: A genetic programming approach, *Available at SSRN* 3726609.

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

Ma, Linlin, Yuehua Tang, and Juan-Pedro Gomez, 2019, Portfolio manager compensation in the us mutual fund industry, *The Journal of Finance* 74, 587–638.

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

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.

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

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, 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.

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.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams, 1986, Learning representations by back-propagating errors, *Nature* 323, 533.

Simon, Frederik, Sebastian Weibels, and Tom Zimmermann, 2025, Deep parametric portfolio policies, Discussion Paper, 23-01 CFR Working Paper Revised working paper; earlier versions circulated in 2022 and 2023.

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.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le, 2014, Sequence to sequence learning with neural networks, *NIPS'14* pp. 3104–3112.

Sutton, Richard S, and Andrew G Barto, 2018, *Reinforcement learning: An introduction* (MIT press).

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.

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio, 2018, Graph attention networks, in *The 6th International Conference on Learning Representations*.

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 Conference on Knowledge Discovery and Data Mining*.

Wang, Yijie, Hao Gao, Campbell R. Harvey, Yan Liu, and Xinyuan Tao, 2026, Machine learning meets markowitz, Discussion Paper, 34861 NBER Working Paper.

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. Implementation Details of AlphaPortfolio

A1 Promises of Direct Portfolio Construction and RL

Why a "direct" construction of portfolio can be effective, or at times more effective than indirect constructions involving supervised learning? Extant portfolio management approaches traditionally entail first minimizing pricing errors or estimating risk premia from historical samples, and then combining assets to achieve some common investment objectives. Supervised-learning-based approaches cannot flexibly incorporate transaction costs, economic constraints, and alternative objectives. Moreover, financial or economic data tend to be high-dimensional, noisy, and nonlinear, with complicated interaction effects and fast, non-stationary dynamics, rendering traditional econometric tools ineffective in capturing path dependence and cross-asset linkages.

To understand this better, let us write down a generic problem researchers tackle using supervised learning: $\min_{\boldsymbol{\theta}} H(\boldsymbol{\theta}) = (y - f(\boldsymbol{x}, \boldsymbol{\theta}))^2$, where $f(\boldsymbol{x}, \boldsymbol{\theta})$ is the model we train and $\boldsymbol{\theta}$, $\boldsymbol{x}$, and y represent the model parameters, input variables, and labeled outputs, respectively. For example, y could be an asset's return, and we are then simply minimizing pricing errors. With a known functional form of $\frac{\partial H(\boldsymbol{x},\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ to minimize H using gradient descents, the two-step paradigm for portfolio construction then takes the estimated function f^* as given to optimize the best portfolio strategy, z, to maximize a reward function R, i.e., $\max_z R(\boldsymbol{x}, z(f^*(\boldsymbol{x}, \boldsymbol{\theta})))$. If f^* provides the estimates of expected returns and variance-covariance matrix for the assets, and R is investors' next period mean-variance utility, then the optimal construction strategy z is the solution in Markowitz (1952). Many ML models make significant progress in overcoming the difficulty of estimating $f(\boldsymbol{x}, \boldsymbol{\theta})$ with high-dimensional data, but there is no guarantee that the $H(\boldsymbol{\theta})$-minimizing f^* also maximizes R in general. A simple case in point is the construction of a long-only portfolio. If some assets generate extremely negative returns and would not be included in the portfolio in any case, using historical data to train a model to minimize estimation errors in these assets' higher moments is not necessarily helping improving the portfolio's performance.

A direct construction formulates the optimization as $\max_\theta R(\boldsymbol{x}, z(\boldsymbol{x}, \boldsymbol{\theta}))$, where $z(\boldsymbol{x}, \boldsymbol{\theta})$ is a direct investment strategy model whose output is a portfolio, and the reward $R(\cdot)$, which could be the OOS Sharpe ratio, average returns after transaction costs, etc., is a general function of the dynamic portfolio strategy z and market environment $\boldsymbol{x}$. We often do not observe z (and R) directly, not to mention that in reality, z could alter $\boldsymbol{x}$ as well. Therefore, we use RL to "explore" different investment strategies (different $\boldsymbol{\theta}$) in different market environments (different $\boldsymbol{x}$) to see what rewards we get. This process is data-driven without requiring analytical models of z and is equivalent to a Monte Carlo sampling of the investment path and the reward function $R(\cdot)$. RL uses gradient-based solutions to guide the sampling with the objective of maximizing R and has

been found empirically to be more effective than supervised learning in complex environments, as seen in various AI applications, such as AlphaGo and self-driving.

A supervised learning approach to portfolio construction necessarily introduces a separation: $\max_z R(\boldsymbol{x}, z)$ and $\min_{\boldsymbol{\theta}} H(\boldsymbol{\theta}) = (z^* - z(\boldsymbol{x}, \boldsymbol{\theta}))^2$, i.e., we maximize the reward function with respect to portfolio construction strategy z, which in turn is estimated in a minimization problem requiring historical values of the optimal z^*. But z^* typically cannot be directly observed or accurately approximated or easily computed from the training data. Hence, a supervised learning approach for maximizing R also has to resort to Monte Carlo sampling in some sense. Traditional two-step constructions do not aim to maximize R when they generate the sampling of y. Even when the sampling is guided by maximizing R, a supervised learning formulation can still have a prediction loss gap between z^* and $z(\boldsymbol{x}, \boldsymbol{\theta})$. Take the OOS Sharpe ratio, for example. Even if one directly optimizes portfolio weights using supervised learning, one has to supply the "correct" maximal Sharpe ratio in the training, which requires either exhaustively computing possible portfolio constructions to get the historical maximal Sharpe ratios or introducing an intermediate proxy for it that is subject to model misspecification and approximation errors.

In contrast, deep RL can handle unlabeled data and is computationally feasible, making it well-suited for the direct construction task.[29] Moreover, it easily accommodates more general performance metrics as well as dynamic interactions between investment actions and the market environment, allowing portfolio constructions to incorporate price impacts, transaction costs, dynamic budget constraints, clients' long-range goals, etc. (Fischer, 2018).

A2 Technical Details of the Panel Transformer

To better connect the model architecture to the empirical implementation, we provide additional details on how AlphaPortfolio is built upon the panel Transformer and how it is implemented in practice.

A key advantage of the panel Transformer is that its computation can be expressed almost entirely in matrix form. This makes the architecture highly amenable to GPU-based parallelization and therefore well suited to large-scale model training and inference. We now describe this implementation in more detail.

Given the historical sequence for asset i at date t, represented by the matrix $\boldsymbol{Z}_i^{(t)} = \big(\boldsymbol{z}_{i,t-K}, \ldots, \boldsymbol{z}_{i,t-1}, \boldsymbol{z}_{i,t}\big)$, the SREM component of the panel Transformer uses parameter matrices $\boldsymbol{W}_s^{(Q)}$,

[29] We are not claiming that there cannot be an effective supervised-learning-based approach for direct construction. In contrast, we believe that it is an interesting topic for future research.

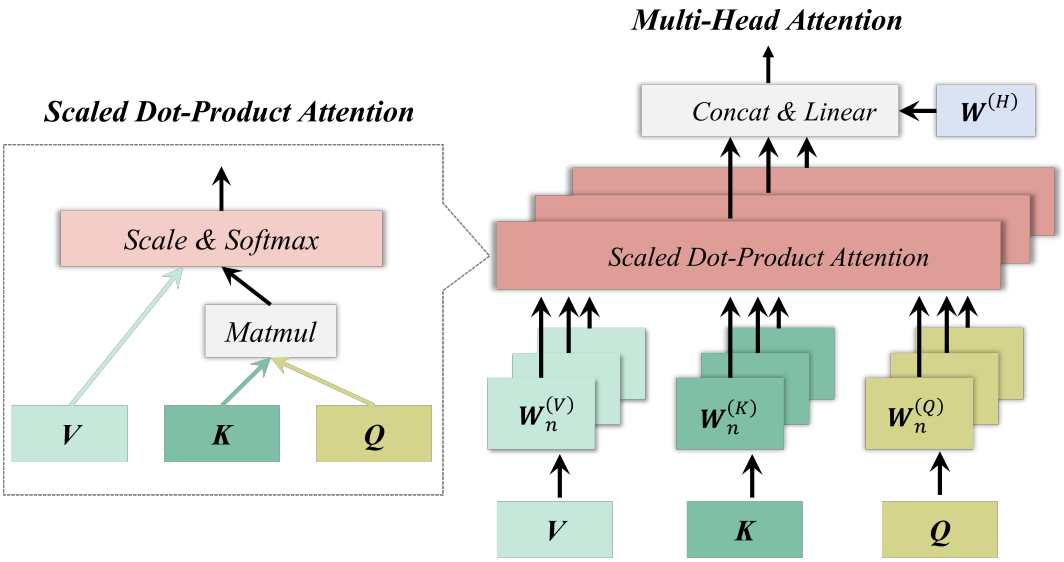

Figure 5: Scaled dot-product attention (left) and multi-head attention (right).

$\boldsymbol{W}_s^{(K)}$, and $\boldsymbol{W}_s^{(V)}$ to transform it into query, key, and value matrices:

$$\boldsymbol{Q}_i^{(t)} = \boldsymbol{W}_s^{(Q)} \boldsymbol{Z}_i^{(t)}, \qquad \boldsymbol{K}_i^{(t)} = \boldsymbol{W}_s^{(K)} \boldsymbol{Z}_i^{(t)}, \qquad \text{and} \qquad \boldsymbol{V}_i^{(t)} = \boldsymbol{W}_s^{(V)} \boldsymbol{Z}_i^{(t)}. \tag{23}$$

These matrices are generated simultaneously and can therefore be computed in parallel. The panel Transformer then applies scaled dot-product attention (as illustrated in the left panel of Figure 5):

$$\boldsymbol{H}_i^{(t)} = \text{SDPAttention}\left(\boldsymbol{Q}_i^{(t)}, \boldsymbol{K}_i^{(t)}, \boldsymbol{V}_i^{(t)}\right) = \text{Softmax}\left(\frac{\boldsymbol{Q}_i^{(t)} \boldsymbol{K}_i^{(t)\top}}{\sqrt{D_k}}\right) \boldsymbol{V}_i^{(t)}, \tag{24}$$

where D_k is the dimension of the key, query, and value vectors, and serves as the scaling factor to prevent the dot products from becoming excessively large.

The panel Transformer also inherits the multi-head attention mechanism from the standard Transformer. As illustrated in the right panel of Figure 5, multi-head attention can be viewed as applying scaled dot-product attention in N different feature subspaces and then concatenating the resulting outputs. Specifically, the output matrix $\boldsymbol{H}_i^{(t)}$ in Eq. (24) can be viewed as one attention head. Given N attention heads indexed by $n = 1, \ldots, N$, the multi-head attention output is

$$\begin{aligned} \boldsymbol{M}_{i,t} &= \text{MultiHead}\left(\boldsymbol{Q}_i^{(t)}, \boldsymbol{K}_i^{(t)}, \boldsymbol{V}_i^{(t)}\right) = \text{Concat}\left(\boldsymbol{H}_{i,1}^{(t)}, \ldots, \boldsymbol{H}_{i,N}^{(t)}\right) \boldsymbol{W}^{(H)}, \\ \boldsymbol{H}_{i,n}^{(t)} &= \text{SDPAttention}\left(\boldsymbol{Q}_i^{(t)} \boldsymbol{W}_n^{(Q)}, \boldsymbol{K}_i^{(t)} \boldsymbol{W}_n^{(K)}, \boldsymbol{V}_i^{(t)} \boldsymbol{W}_n^{(V)}\right). \end{aligned} \tag{25}$$

In addition to scaled dot-product attention, the SREM also adopts the feed-forward network (FFN) module of the standard Transformer. Given the multi-head attention output $\boldsymbol{M}_{i,t}$, the FFN

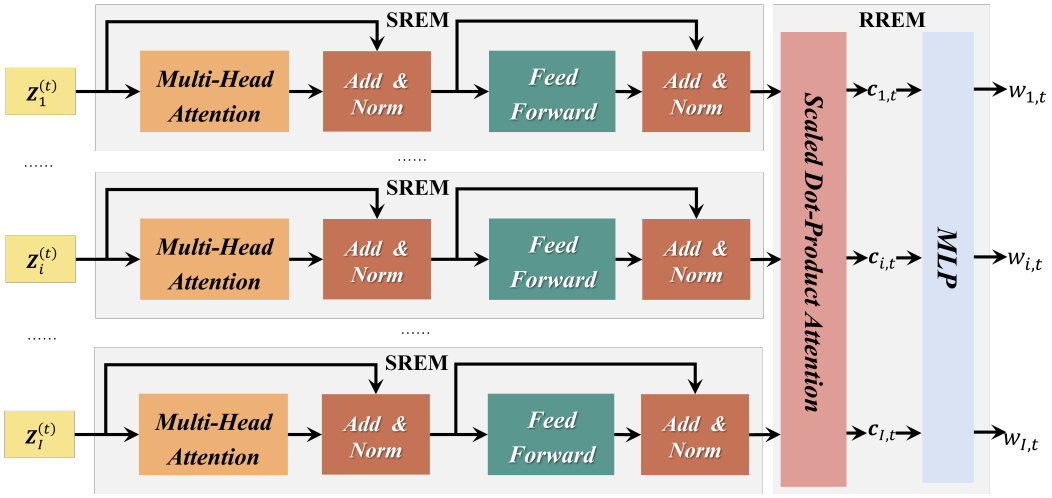

Figure 6: Architecture of the panel Transformer.

is implemented in matrix form as

$$\boldsymbol{U}_{i,t} = \boldsymbol{W}^{(F_2)}\text{ReLU}\left(\boldsymbol{M}_{i,t}\boldsymbol{W}^{(F_1)}\right), \tag{26}$$

where $\boldsymbol{W}^{(F_1)}$ and $\boldsymbol{W}^{(F_2)}$ are learnable parameters, and $\text{ReLU}(\cdot)$ denotes the rectified linear unit activation function. To facilitate training, the SREM also incorporates the residual connections and layer normalization used in the Transformer, represented by the Add & Norm modules in Figure 6.

Next, the RREM concatenates $\boldsymbol{U}_{i,t}$ across all assets $i = 1, \ldots, I_t$ into the unified matrix $\boldsymbol{U}_t = \text{Concat}\left(\boldsymbol{U}_{1,t}, \ldots, \boldsymbol{U}_{i,t}, \ldots, \boldsymbol{U}_{I_t,t}\right)$. The RREM then uses parameter matrices $\boldsymbol{W}_c^{(Q)}$, $\boldsymbol{W}_c^{(K)}$, and $\boldsymbol{W}_c^{(V)}$ to transform $\boldsymbol{U}_t$ into query, key, and value matrices:

$$\boldsymbol{Q}_t = \boldsymbol{W}_c^{(Q)}\boldsymbol{U}_t, \qquad \boldsymbol{K}_t = \boldsymbol{W}_c^{(K)}\boldsymbol{U}_t, \qquad \text{and} \qquad \boldsymbol{V}_t = \boldsymbol{W}_c^{(V)}\boldsymbol{U}_t. \tag{27}$$

These matrices are then combined through scaled dot-product attention:

$$\boldsymbol{C}_t = \text{SDPAttention}\left(\boldsymbol{Q}_t, \boldsymbol{K}_t, \boldsymbol{V}_t\right). \tag{28}$$

The i-th column of $\boldsymbol{C}_t$, denoted by $\boldsymbol{c}_{i,t}$, is the representation of asset i that incorporates both sequential dependence and cross-asset relational dependence. The final desirability score of asset i is then given by $s_{i,t} = \text{MLP}(\boldsymbol{c}_{i,t}; \boldsymbol{\theta}_{\text{MLP}})$.

Our model is implemented in PyTorch version 1.11.1 on four NVIDIA 3090 GPUs. To improve computational efficiency and fully utilize the available hardware resources, we use PyTorch's advanced APIs to automate data parallelism for the panel Transformer. As shown in Eqs. (23)–(28), all computations are formulated as matrix operations and can therefore be efficiently parallelized

and accelerated within the PyTorch framework.

The panel Transformer can also be stacked into multiple layers, as in the original Transformer architecture. However, because the scale and structure of data in neural machine translation differ substantially from those in our setting, we do not adopt the original configuration with six stacked layers. Instead, we find that a single layer already delivers strong performance. For computational efficiency, we also reduce the embedding dimension of the query, key, and value vectors from 512 to 256, and the FFN dimension from 2048 to 1024. The number of attention heads is set to four. More detailed hyperparameter settings of the panel Transformer are reported in Table A.1.

Table A.1: Hyperparameters of the Panel Transformer

Hyperparameter	Choice	Hyperparameter	Choice
Embedding dimension	256	Optimizer	SGD
Feed-forward network dimension	1024	Learning rate	0.0001
Number of heads	4	Dropout ratio	0.2
Number of stacked layers	1	Training epochs	30

A3 Technical Details for Economic Distillation

In economic distillation Approach 1 (Algorithm 1), we use gradients to select important characteristics, then adopt Lasso to select terms and coefficients.

Algorithm 1: pseudo-code for economic distillation Approach one

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

Output: Evaluation metrics of the test period;

for $n = 1$ to N **do**
  for each $\boldsymbol{X}^{(i)}$ in $\mathcal{X}_n$, generate 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 the selected terms, obtain corresponding coefficient;
  Re-generate 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 $R = \{r_1, \ldots, r_N\}$;

Here we report the evaluation metrics of the distillation for algorithm 1.

Table A.2: OOS Performance of Distillation for Algorithm 1 with Polynomial of Degree Two

In each month of the OOS periods, the AlphaPortfolio is distilled into a linear model, and desirability scores on the test sample are regenerated. Portfolios are formed according to the distilled desirability scores following the same strategy as in Table 1. q_n symbolizes the n^{th} NYSE size percentile. Return, Std.Dev., and Sharpe ratio are all annualized.

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		$> q_{10}$		$> q_{20}$	
					$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	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

Algorithm 2: Pseudo-code for Economic Distillation Approach two

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

Output: Evaluation metrics of test period;

Generate score $\boldsymbol{S}$ of all trading periods in $\mathcal{D}_{train}$ using AP;
Calculate gradient-based sensitivity 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 = α);
Regress $\boldsymbol{S}$ to the selected terms, obtain corresponding coefficients;
for $n = 1$ to N **do**
    Use $\mathcal{X}_n$ to construct selected terms;
    Regenerate score $\boldsymbol{s}'_n$ using selected terms and coefficients;
    Using $\boldsymbol{s}'_n$ to calculate the rate of return r_n of this trading period;
end for
Calculate evaluation metrics given $R = \{r_1, \ldots, r_N\}$;

A4 AlphaPortfolio Based on LSTM and GAT

We present here an alternative implementation of the AlphaPortfolio model, in which the SREM is implemented using a Long Short-Term Memory network with attention (LSTM-Att), and the RREM is implemented using a Graph Attention Network (GAT).

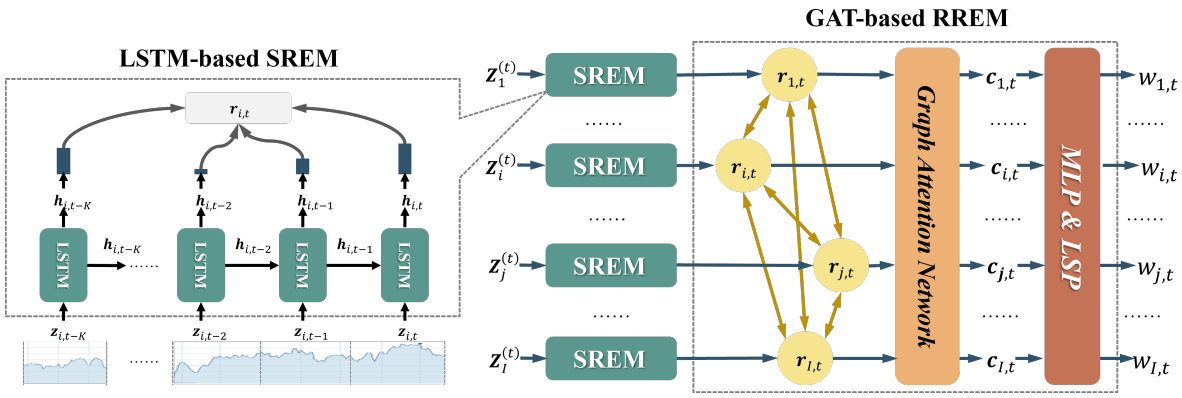

Figure 7: The Architecture of LSTM-GAT-based AlphaPortfolio.

We first describe the implementation of the SREM. Let the historical sequence for asset i at date t be $\boldsymbol{Z}_i^{(t)} = (\boldsymbol{z}_{i,t-K}, \ldots, \boldsymbol{z}_{i,t})$. For each lag $\ell \in \{t-K, \ldots, t\}$, the SREM recursively encodes $\boldsymbol{z}_{i,\ell}$ into a hidden representation using an LSTM:

$$\boldsymbol{h}_\ell = \mathrm{LSTM}\Big(\boldsymbol{z}_{i,\ell}, \boldsymbol{h}_{\ell-1}\Big), \tag{29}$$

where $\boldsymbol{h}_\ell$ denotes the hidden state at step ℓ. The terminal hidden state $\boldsymbol{h}_t$ summarizes the sequential dependence in the entire historical sequence $\boldsymbol{Z}_i^{(t)}$.

While $\boldsymbol{h}_t$ captures sequential dependence through the recursive structure of the LSTM, an additional attention mechanism is introduced to better account for global and long-range dependencies by leveraging all intermediate hidden states $\boldsymbol{h}_\ell$. Specifically, following the standard attention architecture (Sutskever, Vinyals, and Le, 2014), the attention-enhanced historical representation, denoted by $\boldsymbol{r}_{i,t}$, is computed as

$$\boldsymbol{r}_{i,t} = \sum_{\ell=t-K}^{t} \mathrm{ATT}\left(\boldsymbol{h}_t, \boldsymbol{h}_\ell\right) \boldsymbol{h}_\ell, \tag{30}$$

where the attention function $\mathrm{ATT}(\cdot,\cdot)$ is defined as

$$\begin{aligned} \mathrm{ATT}\left(\boldsymbol{h}_t, \boldsymbol{h}_\ell\right) &= \frac{\exp\left(\alpha_\ell\right)}{\sum_{\ell'=t-K}^{t} \exp\left(\alpha_{\ell'}\right)}, \\ \alpha_\ell &= \boldsymbol{w}^\top \tanh\Big(\boldsymbol{W}^{(1)}\boldsymbol{h}_t + \boldsymbol{W}^{(2)}\boldsymbol{h}_\ell\Big). \end{aligned} \tag{31}$$

Here, $\boldsymbol{w}$, $\boldsymbol{W}^{(1)}$, and $\boldsymbol{W}^{(2)}$ are trainable parameters. The resulting vector $\boldsymbol{r}_{i,t}$ in Eq. (30) serves as the output of the SREM.

We next describe the implementation of the RREM. The RREM is constructed using a Graph Attention Network (GAT) (Veličković, Cucurull, Casanova, Romero, Liò, and Bengio, 2018). Specifically, each asset is treated as a vertex in a graph, with vertex set denoted by $V = \{v_1, \ldots, v_i, \ldots, v_I\}$.

Let the adjacency matrix be $\boldsymbol{E} \in \{0,1\}^{I\times I}$, where the (i,j)-th entry satisfies $e_{ij}=1$ if there is a connection from v_i to v_j, and $e_{ij}=0$ otherwise. For each vertex v_i, its feature is given by the sequence representation $\boldsymbol{r}_{i,t}$ extracted by the SREM. In this way, the investable assets and their sequence representations are organized into a graph $G_t=\{V,\boldsymbol{E},\boldsymbol{R}_t\}$, where $\boldsymbol{R}_t=(\boldsymbol{r}_{1,t},\ldots,\boldsymbol{r}_{i,t},\ldots,\boldsymbol{r}_{I,t})$.

The RREM then applies a GAT to extract cross-asset relation representations:

$$\boldsymbol{C}_t=(\boldsymbol{c}_{1,t},\ldots,\boldsymbol{c}_{i,t},\ldots,\boldsymbol{c}_{I,t})=\mathrm{GAT}\Big(G_t;\boldsymbol{\theta}_{\mathrm{GAT}}\Big), \tag{32}$$

where $\mathrm{GAT}(\cdot)$ denotes the graph attention network and $\boldsymbol{\theta}_{\mathrm{GAT}}$ denotes its parameters. The output $\boldsymbol{c}^{i,t}$ is the relation-aware representation for asset i at date t. In the GAT implementation, we assume that adjacency matrix $\boldsymbol{E}$ is fully connected. This allows the model to learn cross-asset dependencies directly from the data without imposing restrictive prior assumptions. In practice, the graph can also be constructed using prior economic information, such as industry classifications, supply-chain linkages, common ownership, or return comovement. However, such prior structure may be misspecified and omit economically relevant links. We therefore adopt the fully connected graph as a more agnostic and flexible specification. Finally, $\boldsymbol{c}^{i,t}$ is mapped into a desirability score: $s_{i,t}=\mathrm{MLP}\left(\boldsymbol{c}_{i,t};\boldsymbol{\theta}_{\mathrm{MLP}}\right)$. Figure 7 illustrates the architecture of the LSTM-GAT-based AlphaPortfolio.

Table A.3 reports the out-of-sample performance of the LSTM-GAT-based AlphaPortfolio. In terms of out-of-sample performance metrics such as the Sharpe ratio, the LSTM-GAT specification even outperforms our panel Transformer-based AlphaPortfolio.

Table A.3: Out-of-Sample Performance of LSTM-GAT-based AP

This table presents the OOS Performance of LSTM-GAT-based AP. For each month in the OOS periods (1990-2016), AP constructs a long-short portfolio, which goes long the 10% of stocks with the highest desirability scores and shorts the 10% of stocks with the lowest desirability scores. The investment proportions are calculated according to Section 2.3. Return, Std.Dev., and Sharpe ratio are all annualized.

	(1)	(2)	(3)
Firms	All	size $> q_{10}$	size $> q_{20}$
Return(%)	16.90	15.48	15.07
Std.Dev.(%)	7.70	5.06	5.08
Sharpe	2.19	3.06	2.97
Skewness	1.63	0.86	1.06
Kurtosis	6.85	2.43	4.88
Turnover	0.46	0.39	0.40
MDD	0.03	0.01	0.02

However, our economic distillation reveals that LSTM-GAT does not have stable utilization

of input features or economic interpretability. This is consistent with the fact that LSTM deals with vanishing and exploding gradients only in the training sample (Heaven, 2019). Specifically, as seen in Table A.4, LSTM-GAT-based AP tends to select characteristics of the first/last position ("_0" and "_11") in the input sequence, a persistent pattern even when we adjust the number of months used to generate lagged inputs. This is indicative of exploding gradients, which means that the trained model overweights the start and the end of the input time series and foils any gradient-based interpretation of the model.

While the gradient exploding problem can be solved by gradient clipping during training (back-propagation), when we have a well-trained model and use it to test OOS, 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 focuses on the training stage and would not flag such technical issues in test samples. Our economic distillation in fact helps to detect modeling issues out-of-sample that the traditional CS approach neglects.

In addition, compared with the fully attention-based Panel Transformer architecture, the recursive computation in the LSTM implementation in Eq. (29) is less amenable to parallelization. The reason is that LSTM updates hidden states recursively over time, so the representation at each step must be computed conditional on the previous hidden state, which prevents full parallelization across the temporal dimension. By contrast, the panel Transformer relies on self-attention, which can be implemented largely through parallel matrix multiplications on modern hardware. Therefore, the panel Transformer architecture is computationally more scalable and better suited to large-scale model training. In our implementation, the LSTM-GAT-based AP contains only 125,506 parameters in our implementation due to computational constraints, whereas the panel Transformer-based AP contains 1,788,673 parameters — more than an order of magnitude more.

A5 Pseudo Codes for RL Training

The Algorithm 3 provides the pseudo-code for the iterative RL training process of AlphaPortfolio. To enhance clarity and facilitate understanding, we also provide a table summarizing the notation used in our method description (Table A.5). The table includes key symbols, their definitions, and relevant descriptions, offering a quick reference to the terminology and mathematical notation employed in the AlphaPortfolio model. This helps ensure the approach is easy to follow

Table A.4: Fama-MacBeth T-test Values of Selected Terms (LSTM-GAT-based AP)

This table presents the results of using Fama-MacBeth method to interpret Algorithm 1. Polynomial degree is set as one and for all terms with suffixes 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 ago. 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

Algorithm 3: Pseudo-code for Training via Reinforcement Learning

Input: Training dataset $\mathcal{D}_{train} = \{\boldsymbol{\mathcal{Z}}_1, \ldots, \boldsymbol{\mathcal{Z}}_N\}$ which consists of N trading periods; Length T of a multi-period investment (length of a trajectory); Number of training epochs *epoch*;

Output: Optimized parameter $\boldsymbol{\theta}^*$ trained on $\mathcal{D}_{train}$;

Initialize model parameters $\boldsymbol{\theta}_0$;
for $n = 0$ to *epoch* **do**
  for $m = 1$ to M **do**
    for t-th period in the trajectory τ_m **do**
      Generate portfolio using AlphaPortfolio with $\boldsymbol{\theta}_n$ as $\boldsymbol{a}_t = \pi_{\boldsymbol{\theta}_n}(\boldsymbol{\mathcal{Z}}_t)$;
    end for
    Generate the trajectory $\tau_m = \{\boldsymbol{\mathcal{Z}}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{\mathcal{Z}}_T, \boldsymbol{a}_T\}$;
    Generate reward for the trajectory τ_m as $\mathrm{H}_{\pi_{\boldsymbol{\theta}}}(\tau_m)$;
    Calculate gradient $\nabla_{\boldsymbol{\theta}} \mathrm{H}_{\pi_{\boldsymbol{\theta}}}(\tau_m)$ using Eq. (19);
  end for
  Update parameters $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + \lambda \cdot \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{\theta}} \mathrm{H}_{\pi_{\boldsymbol{\theta}}}(\tau_m)$ using Eq. (20);
end for
Set $\boldsymbol{\theta}^* = \boldsymbol{\theta}_n$;

Table A.5: Notation in the AlphaPortfolio Framework

Symbol	Definition
I_t	Number of assets available at date t
$i \in \{1, \ldots, I_t\}$	Asset index
$r_{i,t+1}$	Return of asset i from t to $t+1$
$\boldsymbol{r}_{t+1}$	Vector of asset returns $(r_{1,t+1}, \ldots, r_{I_t,t+1})^\top$
$\boldsymbol{w}_t$	Portfolio weight vector at time t
R^p_{t+1}	Portfolio return $R^p_{t+1} = \boldsymbol{w}_t^\top \boldsymbol{r}_{t+1}$
$\boldsymbol{x}_{i,t}$	Observable characteristics of asset i at time t
$\boldsymbol{X}_t$	Matrix of characteristics for all assets
$\boldsymbol{Z}_t$	State vector including $\boldsymbol{X}_t$ and other variables
$\boldsymbol{a}_t$	Action chosen by the investor at time t
τ	Investment trajectory $(\boldsymbol{Z}_1, \boldsymbol{a}_1, \ldots, \boldsymbol{Z}_{T+1})$
$H(\tau)$	Trajectory-level portfolio objective
$\pi_{\boldsymbol{\theta}}$	Portfolio policy parameterized by $\boldsymbol{\theta}$
$\boldsymbol{\theta}$	Model parameters
K	Lookback window length
G	Size of long and short baskets
M	Number of sampled trajectories in RL training
λ	Learning rate in parameter updates

Appendix B. Input Feature Construction

This section details the construction of the 51 input variables. Raw data come from three WRDS databases: CRSP, CRSP Compustat Merged, and Financial Ratio Firm Level (highlighted).

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

$$A2ME = \frac{AT}{(SHROUT * PRC)} \tag{33}$$

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

$$OA = \frac{\Delta\Big((ACT + CHE - LCT - DLC - TXP) - DP\Big)}{AT_{t-1}} \tag{34}$$

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.

$$C = CHE/AT \tag{35}$$

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

$$C2D = (IB + DP)/LT \tag{36}$$

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

$$CTO = SALE/AT_{t-1} \tag{37}$$

Debt2P: Debt to price is the ratio of long-term debt and debt in current liability to the market capitalization.

$$Debt2P = \frac{(DLTT + DLC)}{(SHROUT \times PRC)} \tag{38}$$

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

$$\Delta ceq = (CEQ_t - CEQ_{t-1})/CEQ_{t-1} \tag{39}$$

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

$$\Delta(\Delta Gm - \Delta Sales) = \frac{SALE_t - COGS_t}{SALE_{t-1} - COGS_{t-1}} - \frac{SALE_t}{SALE_{t-1}} \tag{40}$$

ΔSo: Log change in the split adjusted shares outstanding.

$$\Delta So = \log\left(CSHO_t * AJEX_t\right) - \log\left(CSHO_{t-1} * AJEX_{t-1}\right) \tag{41}$$

Δshrout: Percentage change in shares outstanding.

$$\Delta shrout = (SHROUT_t - SHROUT_{t-1}) / SHROUT_{t-1} \tag{42}$$

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

$$\Delta PI2A = \frac{\Delta (PPENT + INVT)}{AT_{t-1}} \tag{43}$$

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

$$E2P = IB / (SHROUT * PRC) \tag{44}$$

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

$$EPS = IB/SHROUT \tag{45}$$

Free CF: Cash flow to book value of equity.

$$FreeCF = \frac{NI + DP + WCAPCH + CAPX}{BE} \tag{46}$$

Idol vol: Idiosyncratic volatility is the standard deviation of the residuals from a regression 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.

$$Investment = (AT_t - AT_{t-1}) / AT_{t-1} \tag{47}$$

IPM: Pretax profit margin, $EBT/Revenue$.

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

$$IVC = \frac{2 * (INVT_t - INVT_{t-1})}{AT_t + AT_{t-1}} \tag{48}$$

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

$$Lev = (DLTT + DLC) / SEQ \tag{49}$$

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

$$LDP = \frac{\sum (RET - RETX)}{PRC} \tag{50}$$

ME: Size is the market capitalization.

Turnover: Turnover is volume over shares outstanding.

$$Turnover = VOL/SHROUT \tag{51}$$

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

$$NOA = \frac{left\,(AT - CHE - IVAO) - (AT - DLC - DLTT - MIB - PSTK - CEQ)}{AT_{t-1}} \tag{52}$$

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.

$$NOP = \frac{(DVC + PRSTKC - SSTK)}{ME} \tag{53}$$

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.

$$O2P = \frac{DVC + PRSTKC - (PSTKRV_t - PSTKRV_{t-1})}{ME} \tag{54}$$

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

$$OL = (COGS + XSGA)/AT \tag{55}$$

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

$$PCM = (SALE - COGS)/SALE \tag{56}$$

PM: The profit margin (operating income/sales).

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

$$Prof = GP/BE \tag{57}$$

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

$$Q = \frac{(AT + ME/1000 - CEQ - TXDB)}{AT} \tag{58}$$

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.

$$ROC = \frac{(DLTT + ME/1000 - AT)}{CHE} \tag{59}$$

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.

$$S2C = SALE/CHE \tag{60}$$

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

$$Sale_g = SALE_t/SALE_{t-1} - 1 \tag{61}$$

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

$$SAT = SALE/AT \tag{62}$$

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.

$$SGA2S = XGSA/SALE \tag{63}$$

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.

$$Tan = \frac{0.715 * RECT + 0.547 * INVT + 0.535 * PPENT + CHE}{AT} \tag{64}$$

Total_vol: Standard deviation of daily return in the month.

Appendix C. Additional Results for Institutional Implementations

This appendix reports the full factor-adjusted performance results corresponding to the institutional implementation exercises discussed in Section 4.4. The tables provide alphas relative to several commonly used asset-pricing models, including CAPM, the Fama–French–Carhart four-factor model, the Fama–French five- and six-factor models, the Pastor–Stambaugh liquidity model, the Stambaugh–Yuan mispricing model, and the Hou–Xue–Zhang q-factor model.

The first example restricts the investment universe to the largest stocks. As shown in Table C.1, we observe that AlphaPortfolio maintains high performance despite the narrower opportunity set. Both the top-1000 and top-2000 universes yield Sharpe ratios above 1.5 and the top-3000 universe yields a Sharpe ratio of 2, indicating that the method scales effectively to the more liquid segment of the equity market where many practitioners operate.

Table C.1: Out-of-Sample AP Performance with Largest Stocks from 1990-2016

This table reports performance of AlphaPortfolio using 1000, 2000, and 3000 stocks with the largest market capitalizations, focusing on long/short stocks in the highest/lowest decile of desirability-scores from 1990 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 (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). "*," "**," 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	1000	2000	3000	Factor	1000		2000		3000	
				Models	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	12.64	15.87	18.17	CAPM	11.3***	0.11	14.61***	0.083	17.02***	0.064
Std.Dev.(%)	8.06	8.71	9.09	FFC	10.84***	0.317	14.43***	0.307	16.84***	0.273
Sharpe	1.57	1.82	2	FFC+PS	10.19***	0.321	13.2***	0.319	16.64***	0.273
Skewness	2.7	3.02	3.23	FF5	11.34***	0.375	15.0***	0.361	16.80***	0.301
Kurtosis	17.69	17.45	16.57	FF6	11.57***	0.379	15.42***	0.371	17.58***	0.333
Turnover	0.26	0.24	0.23	SY	9.64***	0.331	13.16***	0.323	16.89***	0.241
MDD	0.03	0.03	0.02	Q4	12.0***	0.404	15.86***	0.396	18.06***	0.352

In the second example, we evaluate AlphaPortfolio trained directly to maximize cumulative return. Results from Table C.2 show that the strategy achieves high realized returns while still delivering Sharpe ratios above one.

From the fund perspective, many institutional investors face long-only restrictions. As shown in Table C.3, under this constraint, AlphaPortfolio continues to perform strongly. This demonstrates that the model does not rely on short-selling to achieve competitive performance, making it more

Table C.2: Out-of-Sample AP Performance with Largest Stocks Using Cumulative Return from 1990-2016

This table reports performance of AlphaPortfolio using 1000, 2000, and 3000 stocks with the largest market capitalizations, using cumulative return as its objective, focusing on long/short stocks in the highest/lowest decile of desirability-scores from 1990 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 (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). The first, second, and third columns present the alphas for the largest 1,000, 2,000, and 3,000 stocks, 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	1000	2000	3000	Factor	1000		2000		3000	
				Models	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	15.95	24.85	32.03	CAPM	13.1***	0.21	21.54***	0.17	28.83***	0.13
Std.Dev.(%)	12.36	15.99	17.93	FFC	14.72***	0.426	22.37***	0.364	30.16***	0.32
Sharpe	1.29	1.55	1.79	FFC+PS	14.09***	0.428	20.73***	0.37	29.82***	0.32
Skewness	2.31	3.2	3.38	FF5	16.84***	0.545	25.72***	0.533	33.52***	0.47
Kurtosis	14.04	19.69	22.15	FF6	17.77***	0.569	26.34***	0.539	34.62***	0.49
Turnover	0.28	0.25	0.24	SY	16.68***	0.41	24.17***	0.359	33.07***	0.31
MDD	0.06	0.06	0.05	Q4	18.1***	0.53	26.92***	0.497	35.31***	0.46

applicable in realistic fund settings.

Table C.3: Out-of-Sample AP Performance with Largest Stocks Long Only from 1990-2016

This table reports performance of AlphaPortfolio using 1000, 2000, and 3000 stocks with the largest market capitalizations, focusing only on long stocks in the highest decile of desirability-scores from 1990 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 (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). The first column presents the alphas for the largest 1,000 stocks, while the second column presents the alphas for the largest 2,000 stocks. "*," "**," 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	1000	2000	3000	Factor	1000		2000		3000	
				Models	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	36.09	46.1	51.41	CAPM	27.11***	0.516	36.11***	0.391	41.97***	0.365
Std.Dev.(%)	24.93	31.82	31.14	FFC	26.02***	0.658	34.65***	0.578	41.44***	0.596
Sharpe	1.45	1.45	1.65	FFC+PS	24.23***	0.661	32.04***	0.581	40.78***	0.596
Skewness	1.21	2.79	2.87	FF5	24.39***	0.655	33.9***	0.581	40.47***	0.583
Kurtosis	5.25	16.87	20.72	FF6	25.55***	0.665	35.53***	0.592	43.0***	0.612
Turnover	0.24	0.24	0.24	SY	23.85***	0.633	33.85***	0.563	43.46***	0.552
MDD	0.3	0.24	0.27	Q4	26.22***	0.66	36.51***	0.581	43.44***	0.588

Then, we consider a hedged strategy where AlphaPortfolio takes long positions in the largest stocks while shorting SPY. This setup reflects practitioners' desire to extract alpha while neutralizing broad market exposure. The result is shown in Table C.4. Even under this hedged specification, AlphaPortfolio still attains high Sharpe ratios, confirming the method's effectiveness in generating excess returns beyond simple market exposure.

Table C.4: Out-of-Sample AP Performance with Long Largest Stocks Short SPY for Hedging from 1990-2016

This table reports performance of AlphaPortfolio using 1000, 2000, and 3000 stocks with the largest market capitalizations, focusing on long stocks in the highest decile of desirability-scores while shorting the SPDR S&P 500 ETF Trust for hedging from 1990 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 (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). The first, second, and third columns present the alphas for the largest 1,000, 2,000, and 3,000 stocks, 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	1000	2000	3000	Factor	1000		2000		3000	
				Models	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	16.02	26.03	29.08	CAPM	14.54***	0.081	23.55***	0.102	27.01***	0.066
Std.Dev.(%)	10.38	15.52	15.98	FFC	14.39***	0.406	23.54***	0.443	27.18***	0.432
Sharpe	1.54	1.68	1.82	FFC+PS	13.49***	0.41	22.12***	0.448	26.62***	0.432
Skewness	3.07	3.58	3.68	FF5	14.69***	0.412	24.79***	0.479	27.98***	0.451
Kurtosis	16.23	21.55	24.76	FF6	15.15***	0.42	25.46***	0.487	29.22***	0.477
Turnover	0.23	0.25	0.24	SY	13.76***	0.357	23.9***	0.42	28.79***	0.393
MDD	0.06	0.07	0.04	Q4	15.69***	0.441	26.13***	0.493	29.81***	0.474

Finally, to verify the effectiveness of AlphaPortfolio after applying liquidity and price screens, we retain only the top 1,000 stocks by market capitalization whose average daily trading volume over the past period exceeds \$5M and whose price ranges between \$3 and \$5000. We then evaluate the performance of its long–short and long-only portfolios. The results in Table C.5 show that even under these constraints, AlphaPortfolio still maintains a relatively high Sharpe ratio (SR), demonstrating its robustness and practical applicability in real-world trading environments.

Collectively, these results demonstrate that AlphaPortfolio is both robust and practically effective, achieving strong risk-adjusted returns across a variety of investment constraints and objectives relevant to real-world portfolio management.

Table C.5: Out-of-Sample AP Performance with Largest Stocks Excluding Illiquid Stocks from 2007-2016

This table reports performance of AlphaPortfolio excluding stocks with an average daily volume below $5 million or a price above $5000 or below $3, focusing on long/short positions of the 1000 stocks with the largest market capitalization in the highest/lowest decile of desirability-scores from 2007 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 (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). 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)
Firms	Long Short	Long Only	Factor Models	Long Short		Long Only	
				$\alpha(\%)$	R^2	$\alpha(\%)$	R^2
Return(%)	21.32	30.7	CAPM	18.86^{***}	0.115	24.4^{***}	0.347
Std.Dev.(%)	14.93	22.03	FFC	20.83^{***}	0.405	26.47^{***}	0.518
Sharpe	1.43	1.39	FFC+PS	20.21^{***}	0.407	24.9^{***}	0.522
Skewness	2.99	2.06	FF5	19.99^{***}	0.339	25.24^{***}	0.495
Kurtosis	13.68	7.64	FF6	20.07^{***}	0.409	25.31^{***}	0.526
Turnover	0.28	0.26	SY	22.48^{***}	0.33	28.04^{***}	0.456
MDD	0.15	0.2	Q4	21.03^{***}	0.294	26.61^{***}	0.449

Online Appendices

OA1 Basics of Reinforcement Learning

Articles and books on the basics of neural networks and their applications are widely available by now. 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" and its defining features are "trial" (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.[30] 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.[31] 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

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

[31] 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 $P\left((s_{i+1}, R_{i+1})|s_i, a_i\right)$, 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:

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \Big| S_t = s\right] = \mathbb{E}_\pi\Big[G_t \Big| S_t = s\Big], \text{ for all } s \in S. \tag{65}$$

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

By iterated law of expectations:

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

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 a finite-horizon problem, 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:

$$q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\Big|S_t=s, A_t=a\right] = \mathbb{E}_\pi\Big[G_t\Big|S_t=s, A_t=a\Big], \text{ for all } s\in S, a\in A(s), \tag{66}$$

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,

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

which is also in the form of Bellman equation. Solving this Bellman equation for $q(s,a)$ is the process called value-based learning, such as 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 degrees).[32] 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.,

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

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,

[32]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

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H} R(s_t, a_t)\right] = \sum_{\tau} \mathbb{P}_{\pi_\theta}(\tau) R(\tau). \tag{67}$$

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.[33]

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

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

[33] For the exact way to compute policy gradient, please refer to Sutton and Barto 2008, Ch 13. Or for a simpler version, see this excellent blog post.

where $f_\theta(\cdot)$ is the artificial neural network with weights θ. In Figure A.1, we have input $\boldsymbol{s} = (x_1, x_2, x_3, x_4)$. 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 nonlinear function called activation function $g(\cdot)$ on (u_1, u_2).[34] 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 gradient algorithms to optimize in order to generate the optimal horizontal force and maximize the reward.

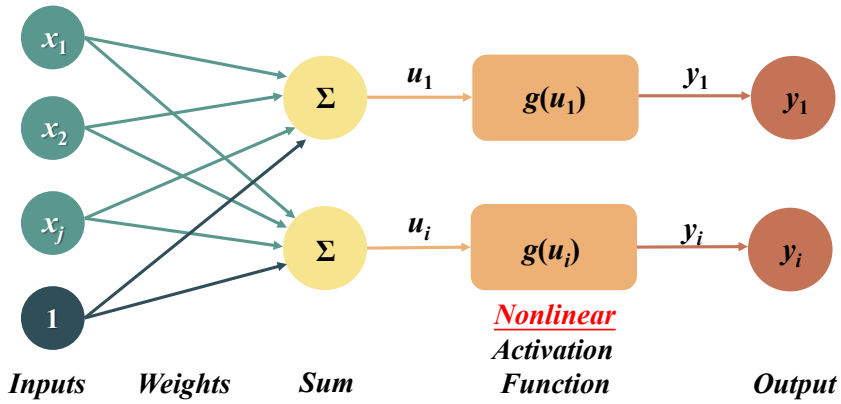

Figure A.1: A Single-Layer Artificial Neural Network

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.

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