NBER WORKING PAPER SERIES

MACHINE LEARNING MEETS MARKOWITZ

Yijie Wang
Hao Gao
Campbell R. Harvey
Yan Liu
Xinyuan Tao

Working Paper 34861
http://www.nber.org/papers/w34861

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

We have read the NBER disclosure policy. Yijie Wang was supported by the National Natural Science Foundation of China Grant No.72501204. Other authors have no funding or relevant financial relationship to disclose. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

Machine Learning Meets Markowitz
Yijie Wang, Hao Gao, Campbell R. Harvey, Yan Liu, and Xinyuan Tao
NBER Working Paper No. 34861
February 2026
JEL No. C45, C55, G11, G12

ABSTRACT

The standard approach to portfolio selection involves two stages: forecast the asset returns and then plug them into an optimizer. We argue that this separation is deeply problematic. The first stage treats cross-sectional prediction errors as equally important across all securities. However, given that final portfolios might differ given distinct risk preferences and investment restrictions, the standard approach fails to recognize that the investor is not just concerned with the average forecast error - but the precision of the forecasts for the specific assets that are most important for their portfolio. Hence, it is crucial to integrate the two stages. We propose a novel implementation utilizing machine learning tools that unifies the expected return generation process and the final optimized portfolio. Our empirical example provides convincing evidence that our end-to-end method outperforms the traditional two-stage approach. In our framework, each investor has their own, endogenously determined, efficient frontier that depends on risk preferences, investor-specific constraints, as well as exposure to market frictions.

Yijie Wang
Tongji University
yijiewang@tongji.edu.cn

Hao Gao
Tsinghua University
gaoh@pbcsf.tsinghua.edu.cn

Campbell R. Harvey
Duke University
Fuqua School of Business
and NBER
cam.harvey@duke.edu

Yan Liu
Tsinghua University
Economics and Management
liuyan@sem.tsinghua.edu.cn

Xinyuan Tao
New Jersey Institute of Technology
xinyuan.tao@njit.edu

1 Introduction

The usual approach to portfolio formation involves two steps: generating expected returns and a covariance matrix, and then optimizing the portfolio given the investor's risk preferences. The first stage usually involves a statistical cross-sectional prediction model whose precision is measured by criteria such as mean squared error (MSE). In this statistical model, forecast errors for all securities are treated equally. As a result, the expected returns for all securities are identical for all investors for a given prediction model.

What if a certain class of investor values forecast precision in a certain group of stocks more than another group of stocks? This investor may be willing to accept a lower cross-sectional R^2 to achieve the precision for the securities that are likely to be heavily weighted in their portfolio, given the investor's risk preferences. As such, there is an obvious link between the first stage (prediction) and the second stage (portfolio design). The misalignment between a statistical objective and a decision-oriented goal (achieved through portfolio weights) can lead to suboptimal investment choices.

The idea of bridging the two steps can be traced back to the fundamental work on conditional efficiency by Hansen and Richard (1987). Our paper provides a new unified, end-to-end (E2E) approach to portfolio design. We explicitly allow for investor heterogeneity in prediction as well as optimization. We utilize machine learning tools in our approach. While much attention has been generated by the promise of machine learning in the statistical prediction stage,[1] our innovation is to utilize these tools in a different way – to merge the prediction and optimization. In essence, we abandon the "predict then optimize" and, as a result, machine learning meets Markowitz.

In real-world asset management, this conceptual gap between prediction and decision-making can be further exacerbated. The suboptimality arising from this misalignment can be traced to at least three critical sources: the neglect of heterogeneous investor preferences, the failure to incorporate economic restrictions and the omission of real-world transaction

[1]There is a rapidly growing body of research on applying machine learning methods to stock return prediction. A few key examples include: Heaton et al. (2017); Gu et al. (2020); Freyberger et al. (2020); Chinco et al. (2021); Chen et al. (2024); Choi et al. (2024); Uddin et al. (2024), and the review by Bagnara (2024).

costs.[2]

First, models trained on a universal statistical objective fail to account for heterogeneous investor risk preferences. Different investors possess different levels of risk aversion, yet a standard ML model produces a single "one-size-fits-all" set of return forecasts. For instance, the needs of a highly concentrated hedge fund with high risk tolerance may substantially differ from those of a broadly diversified, more risk-averse pension fund. Although a model that minimizes a global MSE may serve the pension fund well, it is often suboptimal for other classes of investors, whose performance may be highly dependent on the predictive accuracy for a few specific high-conviction assets. Consequently, an alternative model with different weights on the signals that improves the prediction accuracy on these key assets may deliver superior performance for the hedge fund – even if it has a worse global MSE. Second, these models often overlook important economic restrictions. For instance, much of the documented alpha is concentrated in the short leg of the portfolio (Stambaugh et al., 2012), rendering it unrealizable in markets with stringent short-selling constraints.[3] Finally, the pursuit of predictive accuracy can lead to strategies that are prohibitively expensive to implement. To minimize forecast errors, a model may learn to rely on high-frequency signals, such as short-term reversal characteristics, which inherently demand high portfolio turnover. Although these strategies appear highly profitable on a gross basis, their alpha is often illusory. As documented by Novy-Marx and Velikov (2016), Patton and Weller (2020), Avramov et al. (2023) and Detzel et al. (2023), once realistic trading costs are taken into account, the high turnover can erode or even completely negate the predicted gains.

In this paper, we offer a way to potentially bridge this gap between statistical prediction and portfolio optimization using a novel end-to-end learning framework (E2E) designed for optimal asset management. Instead of training a model to minimize a statistical error like MSE, we integrate the portfolio construction process directly into the return prediction stage.[4] The model's parameters are thus trained to directly optimize a final economic ob-

[2]These issues were recognized in the early literature: Constantinides (1982) highlights the role of heterogeneous preferences; Frost and Savarino (1988) demonstrate the benefits of imposing constraints (restrictions) to mitigate estimation bias; and Magill and Constantinides (1976) and Constantinides (1979) provide foundational analyses of portfolio selection with transaction costs.

[3]Impediments to short selling generally manifest in two forms. First, regulators may explicitly prohibit short positions. Second, even where legally permissible, the cost of borrowing securities can be prohibitively high, rendering the strategy practically infeasible. This latter issue naturally falls into the category of transaction costs.

[4]While portfolio optimization sometimes requires both mean and higher moment inputs, our ML model

jective that embodies the investor's goals and constraints, leading to signals that are not merely statistically accurate but also prescriptively optimal for implementation.

To address the complex demands and practical challenges of optimal portfolio construction, we design a comprehensive E2E framework. Although prior E2E studies in portfolio construction have shown promise, they often focus on stylized problems with specific structural assumptions and are tested only on limited or synthetic datasets. To enhance the framework's applicability to a broader range of investment needs, we introduce several innovations by adapting the modules to embed key economic considerations into model training. First, we incorporate investor risk preferences directly into the learning objective, allowing the framework to generate portfolios along a preference-sensitive efficient frontier. Second, we introduce structural portfolio constraints such as long-only positions and limits on maximum holdings, thereby addressing real-world investment restrictions faced by institutional investors. Third, we extend the framework to explicitly account for transaction costs, enabling the model to internalize trading frictions by dynamically adjusting both signal composition and portfolio allocations. Together, these enhancements allow the E2E framework to bridge the gap between statistical prediction and realistic portfolio implementation, producing investment strategies that are both economically grounded and practically feasible.

Our empirical example uses a comprehensive daily dataset of 5,489 stocks listed in China's A-share market from 2010 to 2023, covering roughly six million stock-day observations and more than 400 company characteristics. We ensure that all predictors are available in real-time at the point of forecast (there is no look-ahead bias). For market-based signals, we use data available up to the close of the previous trading day. For fundamental characteristics derived from financial statements, we ensure that the data is incorporated only after its public announcement. The model is trained on daily stock returns using a rolling-window framework, where each four-year window consists of 39 months for training and 9 months for validation, followed by out-of-sample predictions for the next quarter.

We first begin our analysis by evaluating the conventional two-stage MSE-based approach. Consistent with prior literature (Gu et al., 2020; Drobetz and Otto, 2021; Leippold et al., 2022), we find that these models exhibit strong statistical predictive power: a long-short

focuses on generating the mean forecasts, which is the primary prediction target in the asset pricing literature. For our empirical work, we rely on a simple model which estimates the covariance matrix from historical realized returns.

portfolio strategy yields exceptionally high Sharpe ratios, exceeding 9.0 in some cases. However, a closer look reveals that these gains are largely driven by the short leg of the portfolio. Given the stringent short-sale restrictions in the China A-share market, these short-side profits are not realizable, rendering much of the implied alpha illusionary. In contrast, our E2E framework is explicitly designed for this constrained environment by incorporating the long-only restriction. Focusing on practically feasible long-only portfolios, the E2E framework significantly improves annualized returns compared to the MSE benchmark while maintaining similar volatility. These performance gains are statistically significant across all model specifications.

Second, we evaluate the impact of varying the level of transaction costs. As transaction costs increase, the E2E framework adaptively reduces portfolio turnover, stabilizes trading behavior, and preserves positive net returns. By contrast, the two-stage MSE approach cannot train with cost awareness, keeping a fixed high-turnover signal, leading to rapidly deteriorating performance and even negative returns when transaction costs reach 0.3%. Our tests further show that E2E achieves this resilience by dynamically adjusting signal importance, systematically down weighting high-turnover characteristics and emphasizing lower-cost factors as transaction costs escalate.

Finally, we explore the role of investor heterogeneity in risk preferences by benchmarking our E2E framework against both the traditional two-stage MSE approach and the parametric portfolio policy (PPP) method (Brandt et al., 2009), a representative decision-driven model without explicit optimization. Our analysis first compares the performance of the PPP and MSE approaches. For investors with a high risk tolerance, the decision-aware PPP method outperforms the statistical MSE approach by effectively identifying high-return opportunities. However, as risk aversion increases, the lack of an explicit optimization layer in PPP limits its ability to precisely control portfolio variance, causing it to fall behind the MSE approach. In contrast, our E2E framework combines the advantages of both paradigms, consistently dominating both benchmarks across the entire risk spectrum. Specifically, by integrating an explicit optimization layer into the learning process, E2E retains the structural precision of the MSE approach, enabling it to converge to the minimum-variance portfolio for risk-averse investors. Simultaneously, it leverages the decision-aware learning capability akin to PPP to maximize returns for risk-tolerant investors. This duality allows E2E to generate a superior efficient frontier that encompasses the strengths of both traditional and

parametric approaches.

Our paper makes three contributions. First, we propose a machine-learning-based approach as an alternative to the conventional two-stage "predict-then-optimize" paradigm. Prior applications of machine learning to asset pricing typically estimate returns first and then construct portfolios (Gu et al., 2020; Chen et al., 2024), a separation that often misaligns predictive accuracy with investment objectives. Related work on deep reinforcement learning introduces decision-awareness, but often in stylized trading environments with limited state or action spaces (Deng et al., 2016; Cong et al., 2021). Other prior decision-aware approaches in portfolio construction, such as the Parametric Portfolio Policies by Brandt et al. (2009), directly model weights as functions of predictive signals and do not internalize the outcomes from constrained optimization that economic agents often face. By contrast, we design an E2E learning framework that integrates prediction and portfolio choice in a unified architecture, providing a systematic and scalable approach that embeds economic constraints directly into model training.

Second, we provide several innovative designs that adapt the E2E framework into a systematic and implementable tool for portfolio construction by directly incorporating multiple real-world frictions. We allow investors to embed a wide range of portfolio objectives, spanning both cross-sectional rules, such as equal- or value-weighted portfolios, or time-series penalties, such as transaction costs. We also incorporate investor risk preferences and real-world structural constraints (long-only limits and concentration caps). Incorporating investor heterogeneity as an endogenous component of the learning process highlights an important but underexplored point: investors with different risk appetites should generate different valuations even when faced with the same characteristics. Our framework therefore contributes to a more investor-centric paradigm, bridging the traditional separation between return prediction, economic frictions, and investor utility.

To make the above designs implementable and compatible with the usual machine learning training paradigm, we also make several technical contributions. For example, the modeling of transaction costs inevitably involves agents' temporal portfolio rebalancing, which conflicts with the usual assumption of independent stock-time samples when training machine learning models for financial panel data. We propose a new cross-sectional sampling scheme that allows temporal dependence in portfolio weights, mimicking agents' real-world portfolio rebalancing decisions.

Third, we provide large-scale empirical evidence on the effectiveness of decision-aware learning. Our empirical application shows that the E2E framework consistently outperforms two-stage MSE benchmarks across model specifications, transaction cost levels, and portfolio constraints. The higher-frequency setting enables a more realistic calibration of trading costs than the monthly data commonly used in the literature and provides a systematic and practical solution to address multiple real-world constraints (Avramov et al., 2023). Our findings underscore the importance of integrating prediction, investor heterogeneity, and market frictions within a unified empirical design, producing strategies that are both economically grounded and practically implementable.

Our framework has other important applications that go beyond portfolio control. In the current era of big data and machine learning, the question of how economic agents process complex data and contribute to aggregate equilibrium outcomes is of first-order importance (Dou et al., 2025). Although an investigation of the general equilibrium implications of constrained machine learning agents is beyond the scope of the current paper, we provide an important tool in analyzing individual decision making in this environment. Performance evaluation and funding selection are another scenario in which agents face a wealth of performance-related signals but are constrained to select a few funds to allocate (Li and Rossi, 2020; Kaniel et al., 2023; Wu et al., 2021). Our framework provides an ideal tool for studying risk control (i.e., how investors should control risk exposures to systematic factors), fee management, and fund selection. We defer these applications to future research.

The remainder of this paper is organized as follows. Section 2 discusses the literature and Section 3 presents the general E2E framework and the portfolio construction strategy. Empirical results are reported in Section 4. The final section offers some concluding remarks.

2 Context

2.1 Asset pricing and return prediction

Machine learning (ML) methods have recently gained significant traction in finance, particularly in asset pricing. These approaches attempt to address longstanding econometric challenges such as high dimensionality, nonlinearity, and overfitting (Kelly et al., 2020; Gu et al., 2020; Kozak et al., 2020; Lettau and Pelger, 2020; Kelly et al., 2023). These issues are amplified by the proliferation of documented return predictors, often referred to as the

"factor zoo" (Cochrane, 2011; Hou et al., 2015; Harvey et al., 2016). Unlike traditional portfolio sorts or sparse linear models, ML frameworks flexibly attempt to capture nonlinearities, interaction effects, and time variation in factor loadings, providing a potentially more powerful approach to modeling a large set of return predictors.

In addition, ML has been widely applied to return prediction (Heaton et al., 2017; Gu et al., 2020; Freyberger et al., 2020; Chinco et al., 2021; Chen et al., 2024; Choi et al., 2024; Uddin et al., 2024). Gu et al. (2020) conduct one of the first large-scale comprehensive analyses using U.S. equity data and show that nonlinear models, particularly neural networks and boosted trees, consistently outperform traditional linear models in predicting stock returns by capturing complex interactions among firm characteristics. The success of these methods is not limited to the U.S., as subsequent research has demonstrated their effectiveness in international contexts, including the European (Drobetz and Otto, 2021) and Chinese (Leippold et al., 2022) stock markets. Recent evidence from Chen et al. (2024) further demonstrates the strength of deep neural networks in modeling nonlinear stochastic discount factors, significantly improving cross-sectional pricing accuracy and portfolio performance relative to linear benchmarks.

However, the application of ML techniques to asset pricing is not without pitfalls. Chen and McCoy (2024) offer a cautionary note: ML models often operate on large sets of predictors with missing values, and researchers commonly apply ML-based imputation to fill these gaps. Yet such sophisticated imputation procedures can introduce substantial estimation noise and may lead to underperformance. Chen et al. (2024) show that for a large set of predictors, a purely kitchen-sink application performs poorly; without an economic structure to discipline the algorithm, unconstrained deep networks deliver results that are even worse than simple models. Moreover, Avramov et al. (2023) and Wang (2025) question whether the high profitability delivered by advanced ML models can persist once realistic economic constraints and trading costs are imposed. They show that ML-based return predictability is largely concentrated in hard-to-trade segments of the market, such as microcap and distressed stocks or periods of heightened volatility, and that the profitability of these strategies drops sharply once these economic restrictions are taken into account.

2.2 Addressing Parameter Uncertainty

In the classical two-stage framework, estimation errors in expected returns are prone to being amplified during the optimization process, often resulting in unstable portfolio decisions. A stream of literature has sought to address this parameter uncertainty within the 'predict then optimize' framework. For instance, Jorion (1986) introduces a Bayesian estimation approach to mitigate the impact of estimation error on portfolio weights. Similarly, Jorion (1992) proposes a portfolio resampling technique to manage the instability of the optimizer. More recently, robust optimization techniques have also been employed to immunize portfolios against prediction errors, including norm-constrained approaches (DeMiguel et al., 2009) and distributionally robust optimization methods based on Wasserstein ambiguity sets (Blanchet et al., 2022).

Our approach incorporates estimation uncertainty but goes beyond that. Prediction uncertainty translates into ranking uncertainty in the downstream optimization problem and is therefore penalized by our return- or utility-optimizing agents. On the other hand, many other features of the return formation process are taken into account in our framework through the explicit optimization layer. For example, with long-only constraints, we care more about assets that are potentially on the long end. We save the limited degree of freedom for models to forge a better estimate for assets that we likely long. As such, different assets are weighted differently in reducing the overall MSE. This differential weighting is absent from Bayesian or other traditional approaches.

2.3 Integrated Estimation via SDF and Factor Models

Within the integrated portfolio construction framework, a strand of literature approaches the problem from an asset pricing perspective. Leveraging the duality principle established by Hansen and Jagannathan (1991), estimating the stochastic discount factor (SDF) that minimizes pricing errors is equivalent to solving for the conditional mean-variance efficient portfolio. Building on this insight, recent studies have employed machine learning to estimate the SDF directly, thereby achieving a "one-step" tangency portfolio construction.

Kozak et al. (2020) characterize factor investing as a search for SDF weights within a vast "factor zoo." They employ regularization techniques (shrinkage) to estimate SDF coefficients directly, bypassing the need for explicit estimation of the high-dimensional covariance

matrix. To better capture the nonlinear relationship between factors, Gu et al. (2021) and Chen et al. (2024) utilize deep neural networks, such as autoencoders and generative adversarial networks, to construct latent factors that better span the efficient frontier. Notably, Chen et al. (2024) integrate the "no-arbitrage" condition directly into the loss function as a penalty, exemplifying the integrated learning paradigm. More recently, Didisheim et al. (2024) introduce the Artificial Intelligence Pricing Theory, which theoretically advocates the 'virtue of complexity.' They argue that models utilizing a vast number of predictors can asymptotically recover the optimal tangency portfolio, outperforming sparse factor models.

However, while these SDF-based methods theoretically identify the maximum Sharpe ratio portfolio, they typically operate under the assumption of a frictionless market. As noted by Avramov et al. (2023), the resulting tangency portfolio often entails extreme leverage or aggressive short positions, which is infeasible for real-world asset investors.

2.4 Deep Reinforcement Learning in portfolio design

Deep Reinforcement Learning (DRL) has emerged as a powerful paradigm that integrates the sequential decision-making framework of reinforcement learning (RL) with the high-capacity representational power of deep neural networks (Arulkumaran et al., 2017). Compared to traditional RL methods that require hand-crafted features, DRL excels at automatic representation learning, allowing the agent to discover salient information from high-dimensional, unstructured data for its task. This capability has driven remarkable successes across diverse domains, including game playing (e.g., AlphaGo) (Silver et al., 2016), robotics (Zhao et al., 2020), natural language processing (Sharma and Kaushik, 2017), and computer vision (Le et al., 2022).

The capacity of DRL to process high-dimensional data and optimize policies based on a reward signal makes it a natural fit for financial applications such as option pricing and algorithmic trading (Jiang et al., 2017; Du et al., 2020; Cao et al., 2024). Recently, researchers have begun to apply DRL directly to portfolio construction, using investment profit or utility as a reward function to train the model (Deng et al., 2016; Heaton et al., 2017; Jiang et al., 2017; Yu et al., 2019; Pricope, 2021; Cong et al., 2021). Although these frameworks have shown some advantages over traditional supervised learning models, they are often constrained by the challenge of deriving policy gradients (Sutton et al., 1999) for complex

high-dimensional action spaces. This typically leads to a simplified portfolio formation process. For example, Deng et al. (2016) restrict the action space to a set of discrete choices of long, short, or neutral. Cong et al. (2021) advance the DRL application in direct portfolio construction by considering the entire cross-section of assets. However, due to the difficulty in deriving policy gradients, the final portfolio weights are derived heuristically by normalizing a terminal "winner score" rather than by solving a portfolio optimization problem. A key shortcoming of this design is its inability to enforce explicit constraints or embed specific objectives in the portfolio construction process. Consequently, the resulting portfolios cannot be guaranteed to align with an investor's utility function and often fail to incorporate investment constraints.

2.5 Parametric portfolio policy

In parallel to DRL, another stream of literature employs a parametric portfolio policy approach for direct portfolio construction. Brandt et al. (2009) models the portfolio weight of each asset directly as a simple linear function of a small set of its characteristics – an approach pioneered by Hansen and Richard (1987). Similar to the DRL approach, the advantage of this method is that the coefficients of this linear mapping can be trained by minimizing an investor's average decision loss over the historical sample. This ensures that the learned parameters are tailored to the investor's actual objective, such as realized utility or Sharpe ratio, rather than a statistical metric of predictive accuracy. Hjalmarsson and Manchev (2012); Ammann et al. (2016); DeMiguel et al. (2020) further extend this framework to include a larger set of firm characteristics and incorporate transaction costs; however, these studies still operate under a linear parametric mapping assumption.

Considering that the linear specification could be restrictive and may fail to capture the complex non-linear relationships between firm characteristics, recent research has begun to integrate the flexibility of deep learning into the parametric policy framework. Simon et al. (2025) improve the parametric policy approach by replacing the linear function with a deep neural network. Their work allows for the modeling of complex, non-linear relationships and makes a key contribution by introducing the concept of "economic regularization," where an investor's risk aversion naturally constrains model complexity and mitigates overfitting. Feng et al. (2023) employ deep learning to directly construct the tangency portfolio. Their method

uses an economically motivated feed-forward neural network to transform a large set of characteristics into a long-short "deep factor," which is designed to act as a market hedge and optimally combined with benchmark factors. Jensen et al. (2024) derive an "implementable efficient frontier" by incorporating transaction costs into the economic objective and using ML to learn the optimal weights. To achieve analytical tractability and derive closed-form solutions, their framework is built upon the assumption of quadratic transaction costs and generally without hard inequality constraints. Finally, Kelly and Xiu (2023) synthesize these developments and further formalize the methodological standard for the PPP approach. They outline a workflow where model parameters are optimized by maximizing in-sample utility, while tuning parameters are rigorously selected based on the validation set to optimize out-of-sample performance.

Although we share the same approach of direct portfolio construction, our framework fundamentally departs from these parametric policy approaches. The aforementioned studies, from the linear model of Brandt et al. (2009) to the recent deep learning models, all aim to establish a straight functional mapping from characteristics to the final portfolio weights. In this process, there is no explicit constrained optimization. Consequently, the investor lacks precise control over the portfolio as the output weights are mechanically determined by the model inputs and parameter values. For instance, even if an investor desires a simple equal-weighted top-quantile portfolio, achieving such a specific structural mandate is challenging using the approach in Brandt et al. (2009). In fact, models along this line oftentimes control the distance in weights between the output portfolio and some benchmark index in a loose fashion, which means that feasibility constraints may not be met. Moreover, for investors subject to multifaceted economic restrictions, the feasible set often constitutes a complex, high-dimensional space. Since parametric policies rely on direct mappings, they offer no theoretical guarantee that the generated weights will fall within this feasible region, thereby potentially violating essential trading constraints.[5] In contrast, our framework integrates an optimization layer into the learning process. This allows the model to closely represent the portfolio structure required by the investor, satisfying real-world economic needs such as trading constraints, transaction costs, and risk preferences.

[5]To mitigate this issue, Simon et al. (2025) propose adding violation penalties during training to mitigate this drawback; however, these "soft constraints" still cannot ensure feasibility, especially when applied to the out-of-sample data.

2.6 End-to-end learning frameworks

To address the limitations inherent in the ML, DRL and PPP frameworks, a new paradigm has recently emerged from the intersection of machine learning and operations research, often referred to as end-to-end (E2E) learning (Kotary et al., 2021; Donti et al., 2017; Costa and Iyengar, 2023), integrated learning (Butler and Kwon, 2023), or decision-focused learning (Mandi et al., 2024; Huang and Gupta, 2024). Although the E2E framework shares a common principle with DRL and PPP in that it is trained based on a decision loss, it fundamentally departs from most DRL and PPP applications in *how* the portfolio is derived. Unlike prior frameworks, which typically learn an unconstrained policy that maps inputs to portfolio weights, decisions within an E2E model are adaptively derived from a formal, constrained optimization problem (Sadana et al., 2025). This structure enables the decision maker to incorporate specific objective functions and constraints based on their targets or real-world restrictions. Therefore, the E2E framework can more faithfully represent the decision-making process, with the goal of achieving improved performance in a way that is consistent with investor objectives and constraints.

The key technical innovation enabling this E2E framework is the ability to calculate gradients through a constrained optimization problem. An important contribution in this area comes from Amos and Kolter (2017), who first proposed treating a quadratic program (QP) as a differentiable layer within a neural network. By applying the implicit function theorem to the Karush-Kuhn-Tucker (KKT) optimality conditions of the QP, they demonstrate how the downstream task (optimization) can be integrated into the prediction model. Subsequently, this technique was generalized to a broader class of optimization problems (Donti et al., 2017; Pogančić et al., 2019; Blondel et al., 2020; Elmachtoub and Grigas, 2022) and operationalized in libraries such as Cvxpylayers (Agrawal et al., 2019) and PyEPO (Tang and Khalil, 2024), making the approach applicable to a wider range of real-world problems.

Given its architectural properties, the E2E framework is well-suited for portfolio management. Recently, a stream of research has begun to apply this paradigm to portfolio construction. For instance, Butler and Kwon (2023) integrate prediction with mean-variance portfolio optimization using the E2E framework. Their work focuses primarily on the derivation of analytical solutions when the prediction function is linear, and the optimization problem is unconstrained or involves only equality constraints. While Lee et al. (2025) extend this inte-

gration to general non-linear prediction models, their theoretical gradient derivation remains reliant on the analytical solution of the unconstrained mean-variance optimization problem. Costa and Iyengar (2023) design a multiobjective distributionally robust E2E framework with both decision and accuracy consideration. However, their empirical validation is conducted in a limited dataset with only 20 assets and 8 predictive characteristics. In a different vein, Dominguez et al. (2025) propose a structured ensemble learning framework, demonstrating how diversity in the prediction layer can be formally linked to risk diversification in the resulting portfolio. Despite the promise demonstrated by these studies, they lack a comprehensive framework capable of accommodating complex real-world investment mandates and are often restricted by limited empirical validation. This gap is critical, as existing research has yet to fully recognize that the primary value of the E2E paradigm lies in its unique ability to navigate trading restrictions, market frictions and accommodate investor heterogeneity – precisely where traditional approaches typically falter.

Our research addresses these shortcomings on two fronts. First, we develop a unified and extensible E2E framework designed to solve a wide spectrum of asset management problems. This framework systematically incorporates investor risk preferences, real-world trading constraints, and transaction costs into the learning process. Its flexibility empowers managers to specify diverse portfolio construction needs, from cross-sectional rules like equal- or value-weighting to time-series objectives that penalize portfolio turnover. This generalizable architecture provides a systematic and adaptable solution that moves beyond the isolated problems addressed in prior research. Second, we validate this framework through a large-scale empirical investigation using a comprehensive dataset, thus demonstrating the practical appeal and effectiveness of our E2E approach in a realistic investment setting.

3 Method

We now detail the modeling framework,[6] beginning with a comparison between classical MSE-driven objectives and decision-aware alternatives. A simple two-asset example highlights the potential misalignment between prediction accuracy and investment decision soundness. To address this issue, we show how a general end-to-end learning framework integrates portfolio decisions directly into model training. We then examine the challenges of implementing this framework. Next, we focus on a specific subclass of differentiable convex problems: linear programs for constructing long-only, equal-weighted portfolios. The framework is further extended to accommodate value-weighted portfolios, which are widely used in practice. Lastly, we discuss the importance of cross-sectional structure in portfolio construction and how to connect information across multiple cross-sections, which is often overlooked by MSE-based approaches.

3.1 Preliminary

Throughout this paper, we employ an additive prediction error model following Gu et al. (2020), where the excess return of asset i (i.e., asset i's return in excess of a Treasury bill rate) at time $t+1$ can be expressed as:

$$r_{i,t+1} = \mathbb{E}_t[\tilde{r}_{i,t+1}] + \epsilon_{i,t+1}, \tag{1}$$

[6]Notation and terminology: We use bold letters for vectors, while scalars are in regular font. Random variables are designated by tilde signs (e.g., $\tilde{\boldsymbol{r}}$), their realizations are represented by the same symbol without tildes (e.g., $\boldsymbol{r}$), and their estimations are by the same symbol with hat signs (e.g., $\hat{\boldsymbol{r}}$). We denote by $\mathbf{e}$ the vector of all ones. For any $n \in \mathbb{N}$, we define $[n]$ as the index set $\{1, \ldots, n\}$. We use $\odot$ to denote element-wise product. For two vectors $\boldsymbol{a} = [a_1, ..., a_n]^\top$ and $\boldsymbol{b} = [b_1, ..., b_n]^\top$, their element-wise product is given by $\boldsymbol{a} \odot \boldsymbol{b} = [a_1 b_1, ..., a_n b_n]^\top$. The notation of $\mathbb{R}$ denotes the set of real numbers, and $\mathbb{R}_+$ represents the set of all positive real numbers.

where $i \in [N]$ and $t \in [T]$.[7] The expected return $\mathbb{E}_t[\tilde{r}_{i,t+1}]$ is represented as a function of a set of predictor variables $\boldsymbol{x_{i,t}}$, i.e.,

$$\mathbb{E}_t[\tilde{r}_{i,t+1}] = g^\star(\boldsymbol{x}_{i,t}). \tag{2}$$

This framework imposes several restrictions to ensure consistency and stability. First, the pricing function $g^\star(\boldsymbol{x}_{i,t})$ does not vary across stocks or over time. This allows the model to pool information from the entire panel, enhancing the reliability of risk premium estimates for any individual asset. Additionally, the true pricing function $g^\star(\boldsymbol{x}_{i,t})$ only depends on the contemporaneous predictor values $\boldsymbol{x}_{i,t}$, indicating that the predictions do not utilize historical data prior to time t and/or information from other stocks.

Unfortunately, the true pricing function $g^\star(\boldsymbol{x}_{i,t})$ is never accessible to the decision maker. Hence, in the empirical setting of asset pricing, a typical way to estimate $g^\star(\boldsymbol{x}_{i,t})$ is to construct a hypothesized parametric pricing function $g(\boldsymbol{x}_{i,t}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the vector of function parameters. For instance, when a simple linear model is adopted, the hypothesis class and the pricing function can be expressed as:

$$\mathcal{H} = \{g(\boldsymbol{x}_{i,t}; \boldsymbol{\theta}) \mid g(\boldsymbol{x}_{i,t}; \boldsymbol{\theta}) = \boldsymbol{x}_{i,t}^\top \boldsymbol{\theta}\}. \tag{3}$$

We remark that this framework encompasses a wide range of parametric models, such as polynomial regression (Ostertagová, 2012) and neural networks (LeCun et al., 2015). Once the hypothesis class is chosen, the objective is to determine an optimal set of parameters $\boldsymbol{\theta}^\star$. Recall that the true pricing function $g^\star(\boldsymbol{x}_{i,t})$ is independent of t and i, therefore, a common approach to determine the optimal $\boldsymbol{\theta}^\star$ is by minimizing a measure of prediction errors, represented by a loss function $\ell(\boldsymbol{\theta})$, based on the observed panel data. For instance, when the mean-squared error (MSE) is employed, an ordinary least squares (OLS) estimator could be obtained by solving

$$\ell(\boldsymbol{\theta}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} (r_{i,i+1} - g(\boldsymbol{x}_{i,t}; \boldsymbol{\theta}))^2. \tag{4}$$

[7] In practice, the number of stocks can vary across different time periods. However, for simplicity of mathematical presentation and consistent with prior work (Gu et al., 2020), we assume a balanced panel of stocks. This assumption is strictly for notational convenience and does not imply a fixed stock universe. In our empirical implementation, we employ a dynamic universe that accounts for changes in stock listings.

However, in portfolio management, obtaining the expected returns is not an end in itself, but rather a tool to support subsequent investment decisions. Consequently, even if a hypothesized pricing function shows strong predictive power, it does not necessarily align with the asset manager's specific investment objectives or economic restrictions. The effectiveness of a pricing function should be assessed not only by its predictive accuracy but also by how well it supports the broader goals of portfolio selection and risk management. We use a simple linear prediction model to illustrate the potential misalignment between prediction accuracy and investment decisions.

Example. *Consider a mean-variance portfolio optimization problem with four assets. Assets 1 and 2 represent large-capitalization stable firms, while Assets 3 and 4 represent smaller-capitalization high-growth firms. Assume that the return of Asset i is driven by two characteristics: Earning Yield x_{i1} and Growth x_{i2}, and that the value of the characteristic matrix, returns, and covariance matrix are given by*

$$\boldsymbol{x} = \begin{bmatrix} 0.3 & 0.1 \\ 0.35 & 0.2 \\ 0.1 & 0.9 \\ 0.2 & 1.0 \end{bmatrix}, \qquad \boldsymbol{r} = \begin{bmatrix} 0.32 \\ 0.39 \\ 0.92 \\ 1.04 \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 0.0064 & 0.00288 & 0 & 0 \\ 0.00288 & 0.0144 & 0 & 0.0084 \\ 0 & 0 & 0.36 & 0.294 \\ 0 & 0.0084 & 0.294 & 0.49 \end{bmatrix}.$$

Here, the return follows from a true (but unknown) pricing function that differs by firm type:

$$r_i = \begin{cases} x_{i1} + 0.2x_{i2} & \forall i \in [1,2] \\ 0.2x_{i1} + x_{i2} & \forall i \in [3,4], \end{cases}$$

where the large-caps are dominated by Earning Yield and the small-caps are primarily driven by Growth. With a conventional two-stage approach, the asset manager first estimates a single pricing function for all assets using a linear hypothesis class (3)*, leading to an OLS estimator*

$$g_O(\boldsymbol{x}; \boldsymbol{\theta}) = 0.652x_{i1} + 0.927x_{i2}.$$

The coefficients are chosen in OLS to have the lowest MSE. Now consider two alternative pricing functions where we slightly modify the OLS weights by 5%. Both of these functions will have inferior fits, as measured by MSE. The first function overweights the earnings yield, x_{i1}, which leads to a function that favors larger cap ("pro-large cap"). The second function overweights growth, x_{i2}, and the function tilts towards small cap ("pro-small cap"):

$$g_L(\boldsymbol{x};\boldsymbol{\theta}) = 1.05 \times 0.652x_{i1} + 0.95 \times 0.927x_{i2} = 0.684x_{i1} + 0.880x_{i2},$$
$$g_S(\boldsymbol{x};\boldsymbol{\theta}) = 0.95 \times 0.652x_{i1} + 1.05 \times 0.927x_{i2} = 0.619x_{i1} + 0.973x_{i2}.$$

Based on these three pricing functions, the predicted returns are:

$$\hat{\boldsymbol{r}}_O = \begin{bmatrix} 0.288 \\ 0.413 \\ 0.899 \\ 1.057 \end{bmatrix}, \quad \hat{\boldsymbol{r}}_L = \begin{bmatrix} 0.293 \\ 0.416 \\ 0.861 \\ 1.017 \end{bmatrix}, \quad \hat{\boldsymbol{r}}_S = \begin{bmatrix} 0.283 \\ 0.411 \\ 0.938 \\ 1.097 \end{bmatrix}.$$

Suppose that we have two investors with different levels of risk aversion, characterized by a risk aversion parameter η, where a higher value indicates a more conservative (more risk-averse) investor. Consider an investor with $\eta = 10$ (very risk averse) and another with $\eta = 1$ (more risk tolerant). They construct their portfolios by solving the mean-variance optimization problem using the predicted returns from one of the three pricing models:

$$\begin{aligned} \max \quad & \boldsymbol{w}^\top \hat{\boldsymbol{r}} - \eta \boldsymbol{w}^\top \boldsymbol{\Sigma} \boldsymbol{w} \\ \text{s.t.} \quad & \boldsymbol{w} \geq 0, \boldsymbol{w}^\top \boldsymbol{e} = 1. \end{aligned}$$

We evaluate the realized utility of the obtained portfolio weights by plugging them back into the utility function using the true returns

$$\mathbb{U} = \boldsymbol{w}^{\star\top} \boldsymbol{r} - \eta \boldsymbol{w}^{\star\top} \boldsymbol{\Sigma} \boldsymbol{w}^\star.$$

The results are presented in Table 1. For the conservative investor ($\eta = 10$), the pricing function g_L, which slightly emphasizes the factor driving the lower-risk large-cap stocks and de-emphasizes the factor driving the higher-risk small-cap stocks, yields the highest realized utility. Its predictions encourage a portfolio allocation that more effectively captures the true returns of low risk assets resulting in superior realized utility compared to using either the OLS model g_O or the small-cap focused model g_S. Similarly, for the aggressive investor, the pricing function g_S produces the best outcome. Although g_O is the best statistical fit overall, its predictions lead to portfolios whose risk-return trade-off is sub-optimal for both investors. This shows that a predictive model optimized solely for statistical accuracy across all assets may fail to maximize the realized utility if it ignores individuals' specific risk preferences.

Table 1: Realized utility (%) in four-asset example

This table reports the portfolio returns for investors with different risk preferences: risk-averse ($\eta = 10$) and more risk tolerant ($\eta = 1$), under three pricing models. The first model, $g_O(\boldsymbol{x}; \boldsymbol{\theta})$, represents a standard OLS estimation. The second model, $g_L(\boldsymbol{x}; \boldsymbol{\theta})$, is adjusted to be "pro-large-cap" by placing slightly more weight on the Earnings Yield factor, which is likely preferred by risk-averse investors. The third model, $g_S(\boldsymbol{x}; \boldsymbol{\theta})$, is adjusted to be "pro-small-cap" by placing slightly more weight on the Growth factor, which is likely preferred by risk-seeking investors. For each investor, among the three models, the one that generates the highest return (i.e., best performance) is marked with $\star$.

	$g_O(\boldsymbol{x}; \boldsymbol{\theta})$	$g_L(\boldsymbol{x}; \boldsymbol{\theta})$	$g_S(\boldsymbol{x}; \boldsymbol{\theta})$
$\eta = 10$	31.97	$32.01^\star$	31.90
$\eta = 1$	62.84	62.53	$62.90^\star$

3.2 End-to-end learning framework

In the end-to-end (E2E) learning setting, the objective of the asset manager is to determine a pricing function that minimizes the resultant decision loss (i.e., forming a portfolio that is optimal for the investor's utility).[8] Recall that at each cross-section t, the decision maker bases his decision on the predicted return vector

$$\hat{\boldsymbol{r}}_{t+1} = g(\boldsymbol{x}_t; \boldsymbol{\theta}) = [g(\boldsymbol{x}_{1,t}; \boldsymbol{\theta}), ..., g(\boldsymbol{x}_{N,t}; \boldsymbol{\theta})]^\top.$$

Therefore, we can express this decision process as a bilevel problem (Colson et al., 2005; Dempe, 2002; Sinha et al., 2017):

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & f_u\left(\{\boldsymbol{w}_t^\star, \boldsymbol{r}_{t+1}\}_{t=1}^T\right) \\ \text{s.t.} \quad & \boldsymbol{w}_t^\star = \underset{\boldsymbol{w}_t \in \mathcal{W}}{\arg\min}\, f_l(\boldsymbol{w}_t, g(\boldsymbol{x}_t; \boldsymbol{\theta})) \qquad \forall t \in [T]. \end{aligned} \tag{5}$$

[8]The term decision loss could be interpreted as negative utility. Since optimization problems are symmetric, minimizing the decision loss of the asset manager is equivalent to maximizing his utility. In this paper, we present the end-to-end framework using a minimization form to follow the convention of the optimization literature and be consistent with the MSE minimization problem.

Problem (5) consists of one upper-level problem and multiple lower-level problems.[9] To make this formulation concrete, let's consider a practical example of a mean-variance investor. Suppose that we are training our model using historical data from January 2010 to December 2020. In this context, T represents the set of all rebalancing periods (e.g., all market days or months) within this training window.

At each time $t \in [T]$, the lower-level problem receives the vector of predicted returns $\hat{\boldsymbol{r}}_{t+1} = g(\boldsymbol{x}_t; \boldsymbol{\theta})$, from the current prediction model parameterized by $\boldsymbol{\theta}$. Importantly, unlike the classical two-stage approach where $\boldsymbol{\theta}$ is a fixed parameter pre-determined by an independent statistical criterion, our framework treats $\boldsymbol{\theta}$ as a flexible decision variable for the entire bilevel problem. In this architecture, the task of the lower-level problem is to derive the optimal portfolio $\boldsymbol{w}_t^\star$ specifically for the given $\boldsymbol{\theta}$. This allows the prediction parameters $\boldsymbol{\theta}$ to be refined dynamically through a direct feedback loop from the final investment performance, rather than being isolated from the optimizer. In this example, we assume the lower-level problem is subject to long-only and full-investment constraints, which can be written as

$$\boldsymbol{w}_t^\star = \arg\min \left\{ -\boldsymbol{w}_t^\top \hat{\boldsymbol{r}}_{t+1} + \eta \boldsymbol{w}_t^\top \hat{\boldsymbol{\Sigma}}_t \boldsymbol{w}_t : \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^\top \boldsymbol{e} = 1 \right\} \quad \forall t \in [T].$$

Here, $\boldsymbol{\Sigma}_t$ is the estimated covariance matrix and η is the risk-aversion coefficient. The lower-level objective function is $f_l(\boldsymbol{w}_t, g(\boldsymbol{x}_t; \boldsymbol{\theta})) = -\boldsymbol{w}_t^\top \hat{\boldsymbol{r}}_{t+1} + \eta \boldsymbol{w}_t^\top \hat{\boldsymbol{\Sigma}}_t \boldsymbol{w}_t$ and the feasible region is $\mathcal{W} = \{\boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^\top \boldsymbol{e} = 1\}$.

The goal of the upper-level problem is to find the best set of model parameters, $\boldsymbol{\theta}^\star$, which guides the lower-level decisions. To achieve this goal, it optimizes the realized performance of the sequence of portfolios $\{\boldsymbol{w}_t^\star\}_{t=1}^T$ that were constructed during training, i.e.,

$$\min_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} -\boldsymbol{w}_t^{\star\top} \boldsymbol{r}_{t+1} + \eta \boldsymbol{w}_t^{\star\top} \hat{\boldsymbol{\Sigma}}_t \boldsymbol{w}_t^\star.$$

Here, the upper-level objective function $f_u\left(\{\boldsymbol{w}_t^\star, \boldsymbol{r}_{t+1}\}_{t=1}^T\right)$, evaluates the realized mean-variance utility. It is important to note that this framework is generalizable; while we use mean-variance utility for this example, the E2E architecture can easily accommodate alternative economic objectives, such as functions incorporating higher-order moments (e.g.,

[9] When the lower level problems naively set $\boldsymbol{w}_t^\star = g(\boldsymbol{x}_t; \boldsymbol{\theta})\ \forall t \in [T]$, our method reduces to the parametric portfolio policy approach proposed by Brandt et al. (2009).

skewness), other risk measures (e.g., conditional value at risk), or transaction cost penalties.

Intuitively, the above bilevel program can be viewed as a leader-follower game.[10]. Every time the leader (asset pricer) adjusts the pricing function parameter $\boldsymbol{\theta}$, the follower (portfolio manager) reacts optimally to the leader's action and generates the corresponding optimal portfolio allocations $\boldsymbol{w}_t^\star \; \forall t \in [T]$. The goal of this bilevel optimization problem is to learn a set of parameters $\boldsymbol{\theta}$ that minimizes the decision loss. However, solving problem (5) is generally challenging for two reasons. First, even in the case where the lower-level optimization programs $\min_{\boldsymbol{w}_t \in \mathcal{W}} f_l(\boldsymbol{w}_t, g(\boldsymbol{x}_t; \boldsymbol{\theta})) \; \forall t \in [T]$ are convex, the bilevel program is usually not convex in $\boldsymbol{\theta}$ (Butler and Kwon, 2023). Additionally, calculating the gradient of the objective function with respect to $\boldsymbol{\theta}$ requires differentiation through the argmin operator, which is difficult since the optimal solution of the optimization problem cannot be expressed as an explicit function with respect to the input (Donti et al., 2017).

That said, there exists a special case in which the bilevel problem becomes analytically tractable. When the hypothesized pricing function $g(\boldsymbol{x}_t; \boldsymbol{\theta})$ is linear and the optimization problem is nonrestrictive or only contains equality constraints, problem (5) admits a closed-form solution (Butler and Kwon, 2023). However, this setting is too restrictive and is at odds with the realities of modern asset management. Our work departs from this simplified case. First, with the rapid advancement of machine learning, complex non-linear models such as neural networks have become state-of-the-art in asset pricing, moving beyond linear pricing functions. Second, realistic portfolio construction is almost always subject to inequality constraints, such as long-only or maximum holding limits. These constraints, which are central to our framework, rule out an analytical solution. Therefore, by integrating non-linear prediction models and inequality constraints, we solve a more general version of the E2E portfolio problem that is infeasible with prior analytical approaches.

To address the general case, we adopt an iterative optimization approach to solve problem (5). Here, we build on the prior literature (Amos and Kolter, 2017; Donti et al., 2017; Agrawal et al., 2019) to restructure the problem into an equivalent end-to-end neural network. Figure 1 presents a straightforward visualization of this neural network. Unlike traditional deep learning models that use stochastic gradient descent with random batches

[10]The origins of bilevel optimization can be traced back to the field of game theory, specifically to the work of Von Stackelberg, who studied sequential decision-making in duopolies, often referred to as Stackelberg games (Von Stackelberg, 1934, 2010)

of samples, in the E2E model, we adopt batch gradient descent on a per-cross-section basis. During the forward pass, investors input stock characteristics into the $t \in T_B$ cross-sections. Next, the predicted returns $\hat{\boldsymbol{r}}_{t+1}$ $\forall t \in T_B$ are calculated based on the prediction function $g(\boldsymbol{x}_t; \boldsymbol{\theta})$. With the estimation $\hat{\boldsymbol{r}}_{t+1}$, the optimization layer solves the portfolio selection problem and returns the optimal allocation $\boldsymbol{w}_t^\star$. Finally, the overall decision loss is evaluated by $f_u\left(\{\boldsymbol{w}_t^\star, \boldsymbol{r}_{t+1}\}_{t \in T_B}\right)$.

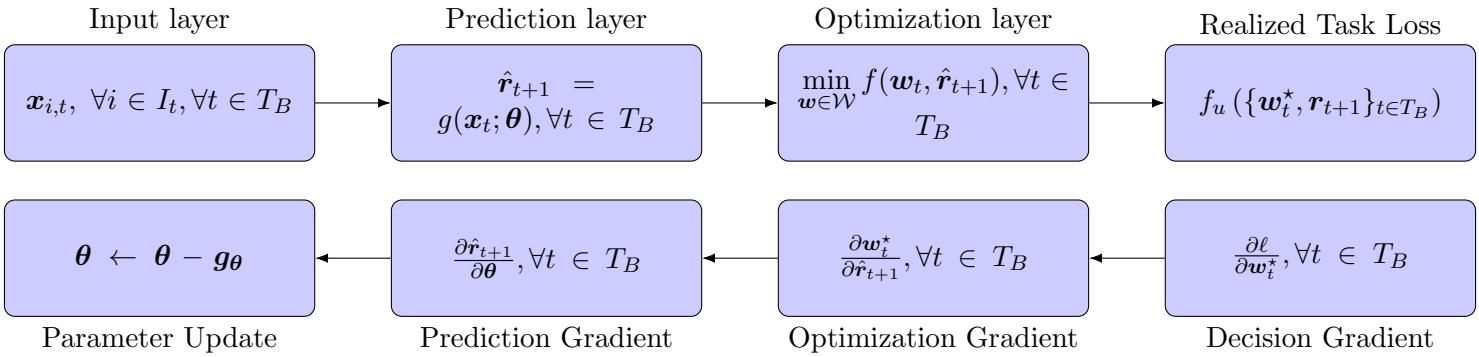


Figure 1: End-to-end neural network

This figure presents the overall structure of the end-to-end (E2E) framework, including a prediction layer, an optimization layer, and a realized loss function.

In the backward pass, the E2E network calculates the gradient of the predictive parameter $\boldsymbol{\theta}$ by the chain rule:

$$\boldsymbol{g}_{\boldsymbol{\theta}} = \frac{1}{T_B} \sum_{t=1}^{T_B} \left(\frac{\partial \hat{\boldsymbol{r}}_{t+1}}{\partial \boldsymbol{\theta}} \right)^\top \left(\frac{\partial \boldsymbol{w}_t^\star}{\partial \hat{\boldsymbol{r}}_{t+1}} \right)^\top \frac{\partial f_u}{\partial \boldsymbol{w}_t^\star}. \tag{6}$$

The partial derivative of the decision loss with respect to the optimal decision is simple (the term on the far right in (6)). In addition, the gradient of the prediction $\hat{\boldsymbol{r}}_{t+1}$ with respect to $\boldsymbol{\theta}$ can be handled efficiently with PyTorch auto-gradient. The main difficulty stems from computing the middle term $\frac{\partial \boldsymbol{w}_t^\star}{\partial \hat{\boldsymbol{r}}_{t+1}}$, capturing how the prediction influences the portfolio allocation decision. The next section details how to differentiate through the optimization layer, as in Figure 1.

3.3 Differentiable convex optimization

When integrating lower-level problems into a neural network, it can be viewed as an optimization layer (Amos and Kolter, 2017; Geng et al., 2021; Huang et al., 2021; Look et al., 2020) that delivers optimal portfolio allocations based on prediction $\hat{\boldsymbol{r}}_{t+1}$, i.e.,

$$\boldsymbol{w}_t^\star(\hat{\boldsymbol{r}}_{t+1}) = \quad \underset{\boldsymbol{w}_t \in \mathcal{W}}{\arg\min}\, f_l(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1}) \quad \forall t \in [T]. \tag{7}$$

Since lower-level problems have similar structures, the differentiation scheme for a given lower-level problem $t \in [T]$ can be easily extended to others. Without loss of generality, we represent the t-th lower-level problem using a general convex optimization problem of the form

$$\begin{aligned} \min_{\boldsymbol{w}_t} \quad & f_l(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1}) \\ \text{s.t.} \quad & \varphi(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1}) = 0 \\ & \phi(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1}) \leq 0, \end{aligned} \tag{8}$$

where $\boldsymbol{w}_t \in \mathbb{R}^N$ is the decision variable and $\hat{\boldsymbol{r}}_{t+1} \in \mathbb{R}^N$ is the input parameter, $f_l(\cdot) : \mathbb{R}^N \to \mathbb{R}$ is the objective function, $\varphi(\cdot) : \mathbb{R}^N \to \mathbb{R}^m$ is the set of m equality constraints, and $\phi(\cdot) : \mathbb{R}^N \to \mathbb{R}^n$ is the set of n inequality constraints. To ensure problem (8) being convex, the objective function $f(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1})$ and each inequality constraint $\phi_j(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1})$ $j \in [n]$ are required to be convex in $\boldsymbol{w}_t$. In addition, the function $\varphi(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1})$ must be affine (Boyd and Vandenberghe, 2004).[11]

Training the E2E neural network requires not only solving the optimization problem in the forward pass but also computing gradients in the backward pass. To differentiate through the optimization layer, we invoke an important concept from mathematical optimization known as KKT conditions (Karush, 1939; Kuhn and Tucker, 2013). The Lagrangian function of the convex optimization problem, given the predicted returns $\hat{\boldsymbol{r}}_{t+1}$, can be written as

$$\mathcal{L}(\boldsymbol{w}_t, \boldsymbol{\nu}_t, \boldsymbol{\lambda}_t)(\hat{\boldsymbol{r}}_{t+1}) = f_l(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1}) + \boldsymbol{\lambda}_t^\top \phi(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1}) + \boldsymbol{\nu}_t^\top \varphi(\boldsymbol{w}_t, \hat{\boldsymbol{r}}_{t+1}), \tag{9}$$

where $\boldsymbol{\nu}_t \in \mathbb{R}^m$ and $\boldsymbol{\lambda}_t \in \mathbb{R}^n$ are known as dual variables (Hoffmann et al., 1989; Luenberger,

[11] In real-world investment, the affine equality constraints $\varphi(\cdot)$ are widely used to enforce a full investment requirement or to impose certain neutrality constraints. Meanwhile, the inequality constraints $\phi(\cdot)$ can enforce short-selling restrictions, concentration caps, and risk budget constraints.

1997). The KKT conditions state that if f, φ, ϕ are differentiable at $\boldsymbol{w}_t^\star$, then $\boldsymbol{w}_t^\star$ is optimal if and only if the following conditions hold:

$$\begin{aligned} \nabla_{\boldsymbol{w}_t} f_l(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) + \partial_{\boldsymbol{w}_t}\phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top \boldsymbol{\lambda}_t^\star + \partial_{\boldsymbol{w}_t}\varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top \boldsymbol{\nu}_t^\star &= 0 \\ \varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) &= 0 \\ \phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) &\leq 0 \\ \boldsymbol{\lambda}_t^\star &\geq 0 \\ diag(\boldsymbol{\lambda}_t^\star)\phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) &= 0. \end{aligned} \tag{10}$$

The first equation represents the stationarity condition, which requires the gradient of the Lagrangian function being zero. The second and third conditions denote the primal feasibility constraints, while the fourth constraint represents the dual feasibility constraints. The last equation is known as the complementary slackness condition (Williams, 1970). Hence, the optimization layer can be viewed as a solver that finds the optimal primal-dual pair $(\boldsymbol{w}_t^\star, \boldsymbol{\nu}_t^\star, \boldsymbol{\lambda}_t^\star)$ that satisfies

$$G(\boldsymbol{w}_t^\star, \boldsymbol{\nu}_t^\star, \boldsymbol{\lambda}_t^\star, \hat{\boldsymbol{r}}_{t+1}) = \begin{bmatrix} \nabla_{\boldsymbol{w}_t} f_l(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) + \partial_{\boldsymbol{w}_t}\phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top \boldsymbol{\lambda}_t^\star + \partial_{\boldsymbol{w}_t}\varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top \boldsymbol{\nu}_t^\star \\ \varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) \\ diag(\boldsymbol{\lambda}_t)\phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) \end{bmatrix} = 0$$

$$\phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) \leq 0$$

$$\boldsymbol{\lambda}_t^\star \geq 0.$$

A good property of the convex optimization problem is that we only need to consider the stationarity, primal feasibility, and complementary slackness conditions when calculating the derivative (Amos and Kolter, 2017). Thus, we can view the optimization layer as a solver that finds the root of the implicit function

$$G(\boldsymbol{w}_t^\star, \boldsymbol{\nu}_t^\star, \boldsymbol{\lambda}_t^\star, \hat{\boldsymbol{r}}_{t+1}) = 0. \tag{11}$$

Next, we invoke the implicit function theorem to compute the Jacobian $\frac{\partial \boldsymbol{w}_t}{\partial \hat{\boldsymbol{r}}_{t+1}}$. Taking the

gradient with respect to $\hat{\boldsymbol{r}}_{t+1}$, we have

$$\begin{bmatrix} \nabla^2_{\boldsymbol{w}_t} f_l(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) + \sum_{i=1}^n \lambda_i^\star \nabla^2_{\boldsymbol{w}_t} \phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) & \partial_{\boldsymbol{w}_t} \phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top & \partial_{\boldsymbol{w}_t} \varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top \\ diag(\boldsymbol{\lambda}_t^\star) \partial_{\boldsymbol{w}_t} \phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) & diag(\phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})) & 0 \\ \partial_{\boldsymbol{w}_t} \varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial \boldsymbol{w}_t}{\partial \hat{\boldsymbol{r}}_{t+1}} \\ \frac{\partial \boldsymbol{\lambda}_t}{\partial \hat{\boldsymbol{r}}_{t+1}} \\ \frac{\partial \boldsymbol{\nu}_t}{\partial \hat{\boldsymbol{r}}_{t+1}} \end{bmatrix} + \tag{12}$$

$$\begin{bmatrix} \frac{\partial \left(\nabla_{\boldsymbol{w}_t} f_l(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}) + \partial_{\boldsymbol{w}_t} \phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top \boldsymbol{\lambda}_t^\star + \partial_{\boldsymbol{w}_t} \varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1})^\top \boldsymbol{\nu}^\star \right)}{\partial \hat{\boldsymbol{r}}_{t+1}} \\ \frac{\partial (\varphi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}))}{\partial \hat{\boldsymbol{r}}_{t+1}} \\ \frac{\partial (diag(\boldsymbol{\lambda}_t) \phi(\boldsymbol{w}_t^\star, \hat{\boldsymbol{r}}_{t+1}))}{\partial \hat{\boldsymbol{r}}_{t+1}} \end{bmatrix} = 0 \tag{13}$$

Solving the above system of equations yields the desired Jacobian matrix $\frac{\partial \boldsymbol{w}_t}{\partial \hat{\boldsymbol{r}}_{t+1}}$.[12]

The differentiation scheme for general convex optimization problems provides a systematic approach to E2E asset management. Although conventional deep reinforcement learning (DRL) methods can also train the model based on an investment reward function, their portfolio weights are often derived heuristically, e.g., through the normalization of a final "winner score" or Q-value (Cong et al., 2021). A key limitation of these DRL models is that the decision maker often lacks direct and precise control over the portfolio structure during the learning process. Consequently, the expected return may not accurately reflect the true investment gains or losses that would arise from applying the learned pricing function in practical investment settings.

In contrast, our optimization-based E2E framework empowers investors to flexibly and explicitly construct portfolios that align with their specific investment objectives and operational constraints. This is achieved by embedding a clearly defined portfolio optimization problem within the learning architecture. As a result, the gradients derived from this integrated optimization problem can guide the learning of the pricing function not just towards prediction accuracy but towards predictions that yield optimal portfolio decisions consistent with the investor's specified goals and restrictions.

[12]For high dimensional problems, we never want to explicitly compute and store the large Jacobian matrix $\frac{\partial \boldsymbol{w}_t}{\partial \hat{\boldsymbol{r}}_{t+1}} \in \mathbb{R}^{N \times N}$. Instead, we follow the chain rule and multiply it by some previous gradient vector $\frac{\partial f_u}{\partial \boldsymbol{w}_t^\star} \in \mathbb{R}^N$ to reduce dimensionality. We refer the reader to Amos and Kolter (2017) for the details of this computational trick.

3.4 Extension to equal-weighted portfolios

Following the previous discussion on differentiating general convex programs, we now focus on a specific subclass: linear programs. Linear programs are viewed as the simplest form of convex problems in the sense that they further restrict the objective function and each inequality constraint is affine in terms of the decision variable. It encompasses a broad class of portfolio construction problems. For example, if one asset manager wishes to maximize the average return of a portfolio that assigns equal weights to the top 10% of assets, the problem can be expressed as

$$\begin{array}{ll} \min_{\boldsymbol{\theta}} & -\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t^{\star\top} \boldsymbol{r}_{t+1} \\ \text{s.t.} & \boldsymbol{w}_t^{\star} = \arg\min\limits_{\boldsymbol{w}_t} \left\{ -\boldsymbol{w}_t^{\top} \hat{\boldsymbol{r}}_{t+1} : \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^{\top} \boldsymbol{e} = 1, \boldsymbol{w}_t \leq \dfrac{10}{N} \right\} \qquad \forall t \in [T]. \end{array} \tag{14}$$

Here, the top 10% stocks are implicitly included by the last constraint, which restricts the maximum holding of any single asset in the portfolio to $10/N$. With this constraint, the portfolio selects assets in a descending manner, until all the weights are used.[13] The portfolio structure can be easily adjusted by changing the maximum holding constraint. For instance, if one asset manager wishes to construct an equally weighted top 100 portfolio, then the last constraint becomes $\boldsymbol{w}_t \leq 0.01$.

The lower-level problems are linear problems as their objective function and constraints are affine in the decision variable $\boldsymbol{w}_t$. Despite this simplicity, the differentiation approach introduced in Section 3.2 faces challenges when applied to such problems. The primary reason is that the solution of a linear program $\boldsymbol{w}_t^{\star}(\hat{\boldsymbol{r}}_{t+1})$, a function of $\hat{\boldsymbol{r}}_{t+1}$, is typically piecewise constant. Consequently, the Jacobian $\frac{\partial \boldsymbol{w}_t}{\partial \hat{\boldsymbol{r}}_{t+1}}$ is zero wherever it is defined and undefined at the breakpoints between the constant pieces. To address this issue, we propose to smooth the piecewise constant solution function $\boldsymbol{w}_t^{\star}(\hat{\boldsymbol{r}}_{t+1})$ by introducing an L_2 regularization term

[13] Strictly speaking, if $N/10$, is not an integer, the marginal asset receives a residual weight less than $10/N$. Given the large cross-section of stocks in our sample, the impact of this residual weight is negligible. For investors requiring a strictly equal-weighted portfolio, the constraint can be adjusted to $\lfloor 0.1N \rfloor$ or $\lceil 0.1N \rceil$ based on their preference.

to the lower-level problem. Specifically, we modify the lower-level problem by

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & -\tfrac{1}{T}\textstyle\sum_{t=1}^{T} \boldsymbol{w}_t^{\star\top} \boldsymbol{r}_{t+1} \\ \text{s.t.} \quad & \boldsymbol{w}_t^{\star} = \arg\min_{\boldsymbol{w}_t} \left\{ -\boldsymbol{w}_t^{\top} \hat{\boldsymbol{r}}_{t+1} + \zeta \|\boldsymbol{w}_t\|_2 : \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^{\top} \boldsymbol{e} = 1, \boldsymbol{w}_t \leq \frac{10}{N} \right\} \qquad \forall t \in [T]. \end{aligned} \tag{15}$$

where ζ is a parameter that controls the Lipschitz continuity of the solution function. The regularized linear program (15) yields an optimal solution close to that of the original problem (14), while providing well-defined gradients. To illustrate the effect of this regularization term in a linear program, we present a straightforward numerical example in Appendix A.3.

3.5 Extension to value-weighted portfolios

This section extends the E2E framework from equal-weighted portfolios to value-weighted portfolios. This approach is favored for capacity and tradability, as it allocates more capital to larger and typically more liquid stocks within the top-ranked group, potentially reducing market impact and improving scalability relative to equal weighting. Value-weighted portfolios have commonly been perceived as difficult to embed within gradient-based direct portfolio construction frameworks. For example, Cong et al. (2021) point out that value-weighted portfolios block the backpropagation procedure for model training in their RL-based model. Concurrently, from an optimization standpoint, value-weighted portfolios are a combinatorial optimization problem, which does not admit a convex formulation as in (14).

To overcome this challenge, we propose a two-phase decomposition strategy to construct value-weighted portfolios within an end-to-end learning framework. The first phase addresses asset selection and initial weighting. Recall that the lower-level problem can be viewed as an implicit problem that yields the optimal portfolio based on the predicted returns $\hat{\boldsymbol{r}}_{t+1}$. Hence, in the first step, we follow the design of the lower-level problem in (14), which yields an equally weighted top-10% portfolio

$$\boldsymbol{w}_t^{\star} = \arg\min_{\boldsymbol{w}_t} \left\{ -\boldsymbol{w}_t^{\top} \hat{\boldsymbol{r}}_{t+1} : \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^{\top} \boldsymbol{e} = 1, \boldsymbol{w}_t \leq \frac{10}{N} \right\} \qquad \forall t \in [T].$$

The second phase explicitly incorporates value-weighting. We rescale the initial weights $w_{i,t}^{\star}$

by their market capitalization $v_{i,t}$, and renormalize the vector to obtain the final portfolio weights that reflect firm size.

$$\overline{w}_{i,t}^{\star} = \frac{v_{i,t} w_{i,t}^{\star}}{\sum_{i=1}^{N} v_{i,t} w_{i,t}^{\star}}, \quad \forall i \in [N], t \in [T].$$

This yields a portfolio where the selected assets are weighted according to their market capitalization. The entire process is encapsulated in a bilevel optimization problem where the upper-level function optimizes the model parameters $\boldsymbol{\theta}$ (governing the prediction $\hat{\boldsymbol{r}}_{t+1}$) to maximize the final portfolio's realized return, and the lower-level function executes the initial portfolio construction:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & -\tfrac{1}{T} \textstyle\sum_{t=1}^{T} \overline{\boldsymbol{w}}_t^{\star\top} \boldsymbol{r}_{t+1} \\ \text{s.t.} \quad & \boldsymbol{w}_t^{\star} = \arg\min_{\boldsymbol{w}_t} \left\{ -\boldsymbol{w}_t^{\top} \hat{\boldsymbol{r}}_{t+1} : \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^{\top} \boldsymbol{e} = 1, \boldsymbol{w}_t \leq \frac{10}{N} \right\} \qquad \forall t \in [T]. \end{aligned} \tag{16}$$

This formulation is amenable to end-to-end training. By differentiating through both phases using the chain rule, we can compute the batched gradient of the overall objective with respect to the model parameters $\boldsymbol{\theta}$:

$$\boldsymbol{g}_{\boldsymbol{\theta}} = \frac{1}{T_B} \sum_{t=1}^{T_B} \left(\frac{\partial \hat{\boldsymbol{r}}_{t+1}}{\partial \boldsymbol{\theta}} \right)^{\top} \left(\frac{\partial \boldsymbol{w}_t^{\star}}{\partial \hat{\boldsymbol{r}}_{t+1}} \right)^{\top} \left(\frac{\partial \overline{\boldsymbol{w}}_t^{\star}}{\partial \boldsymbol{w}_t^{\star}} \right)^{\top} \frac{\partial f_u}{\partial \overline{\boldsymbol{w}}_t^{\star}}. \tag{17}$$

Our approach effectively incorporates value weighting into an end-to-end portfolio construction framework, addressing an important limitation in prior studies.

3.6 Connecting multiple cross-sections

The classical MSE-driven approach in equation (4) treats asset predictions independently and aggregates errors uniformly over time and across assets. Although it can achieve prediction accuracy, this framework overlooks the inherent cross-sectional nature of portfolio construction and can be misaligned with investor' decision-making. In contrast, end-to-end models driven by cross-sectional investment decisions (Butler and Kwon, 2023; Elmachtoub and Grigas, 2022) more closely reflect how asset managers allocate capital at each rebalancing date. However, optimizing decisions within isolated cross-sections is insufficient, since

key performance measures such as volatility, Sharpe ratio, and transaction costs are defined over the time series of portfolio returns generated across successive cross-sections.

Therefore, we propose generalizing the E2E framework to bridge decisions across different cross-sections by endogenizing sequential dependence. Unlike prior E2E studies that often assume independent lower-level problems, we explicitly model the inter-temporal relationship between consecutive portfolios, allowing portfolio construction at time t explicitly dependent on the optimal portfolio of the previous period $\boldsymbol{w}_{t-1}^{\star}$. For instance, a transaction-cost-aware model is formulated by modifying the lower-level objective:

$$\boldsymbol{w}_t^{\star} = \underset{\boldsymbol{w}_t \in \mathcal{W}}{\arg\min} -\boldsymbol{w}_t{}^{\top} g(\boldsymbol{x}_t; \boldsymbol{\theta}) + \Phi(\boldsymbol{w}_t - \boldsymbol{w}_{t-1}^{\star}) \qquad \forall t \in [T],$$

where $\Phi(\boldsymbol{w}_t - \boldsymbol{w}_{t-1}^{\star})$ denotes the transaction cost of trading from portfolio $\boldsymbol{w}_{t-1}^{\star}$ to $\boldsymbol{w}_t$.

This introduction of temporal dependency imposes a unique requirement on the training procedure. Unlike conventional machine learning models that are trained on randomly shuffled mini-batches, our framework necessitates a sequential training scheme. This is because the lower-level optimization at any cross-section relies on the optimal portfolio from the prior period. In this case, a random shuffling of the data would sever this causal link, rendering the lower-level problems infeasible to compute.

Therefore, during training, the cross-sections must be processed in their natural chronological order, allowing the optimal portfolio from one step to serve as the input for the next. Although batching can still be employed for computational efficiency and gradient descent accuracy, each batch must comprise a consecutive block of time periods, thus preserving the temporal integrity of the data and ensuring that the calculated gradients correctly account for the path-dependent dynamics of the investment strategy.

To this end, by optimizing a time-aggregated performance metric, our proposed model is implicitly encouraged to learn temporal dependencies and adjust the pricing function not just for single-period prediction accuracy, but in a way that leads to desirable sequential portfolio outcomes over the long run.

In summary, while we leverage foundational tools from deep learning and differentiable optimization, our distinct methodological innovations are threefold: (1) we introduce a regularization technique to enable gradient propagation through linear programs with inequality constraints, addressing the non-differentiability of standard portfolio sorting and long-short

analysis; (2) we propose a two-phase decomposition strategy to incorporate value-weighting schemes, solving a combinatorial problem that typically blocks backpropagation; and (3) we generalize the learning process to internalize sequential dependencies, allowing the model to adaptively minimize path-dependent transaction costs. We now turn to a large-scale empirical analysis that examines these designs, highlighting the tangible advantages of our approach compared to the traditional two-stage paradigm.

4 Empirical Application

We now evaluate portfolio performance by comparing the E2E approach with a two-stage framework that minimizes mean squared error under different model specifications. We progressively incorporate additional economic restrictions. Specifically, we: (1) incorporate real-world long-only trading restrictions, (2) account for different transaction cost assumptions, and (3) allow for varying levels of investor risk aversion.

4.1 Data and model setting

Our analysis uses daily stock data from China's A-share market, covering the period from January 1, 2010 to December 31, 2023. The A-share market is characterized by stringent trading restrictions, most notably the limitations on short selling. Given our primary objective to evaluate the E2E framework against the traditional two-stage approach, this constrained environment serves as an ideal laboratory. We begin in 2010 because data availability becomes more comprehensive after this date. Our comprehensive sample encompasses 5,489 unique stocks listed during this period. The average number of stocks in each year can be referred to Table A.1 in Appendix A.1.

Our analysis employs a large collection of predictive characteristics that are updated on a daily basis. Following the cross-section of stock returns literature, we construct 415 firm-level characteristics.[14] In addition, we include 31 industry dummies corresponding to the first two digits of Shenwan Hongyuan Industrial Classification codes.

Our model is trained on daily stock return data using a rolling-window framework, with

[14]Our data have a substantial overlap with the list compiled by Harvey et al. (2016). Further, we consolidate our data with other prevailing studies on Chinese equity markets, such as Leippold et al. (2022).

each rolling period advancing by one quarter. For each window, we employ a four-year sample: the first 39 months are used for model training, followed by a 9 month validation period. The trained model is then used to predict daily returns for the next quarter (out-of-sample test period). For example, to generate forecasts for the period from April 1, 2015, to June 30, 2015, we train the model using data from April 1, 2011, to June 30, 2014, and validate it using data from July 1, 2014, to March 31, 2015. All daily returns used in the analysis are market-adjusted excess returns.[15] Using market excess returns removes the influence of market-wide movements, thereby enabling a more direct and fair comparison between our method and the conventional two-stage approach.

The construction of daily-frequency features and the subsequent model training on such a large and granular dataset are computationally intensive and require substantial resources. This scale of analysis stands in contrast to much of the existing literature (Gu et al., 2020; Leippold et al., 2022; Chen et al., 2024), which is typically based on monthly data. With more than 240 trading days per year in the China A-share market, our 14-year daily dataset yields 5,962,768 day-stock observations.

Although daily data are inherently noisier than their monthly counterpart, we choose this higher frequency for two reasons. First, it better reflects the operational horizon of many asset managers. Second, it enables a more realistic calibration of trading costs and market frictions, which can be understated in lower-frequency analyses. To ensure this realism, we adopt a practical daily rebalancing framework that incorporates a one-day execution lag. We assume that portfolio weights are determined using data up to day t, but trades are executed the following day. Thus, realized returns are measured from the volume-weighted average price (VWAP) between 9:30 AM and 3:00 PM on day $t + 1$ to the corresponding VWAP over the same window on day $t + 2$.

4.2 Trading restrictions: Long-only portfolios

We begin by comparing the performance of the end-to-end (E2E) framework and the traditional two-stage framework across two model classes: a linear specification and a nonlinear

[15] We proxy for the daily market return using the CSI 500 Index, which is a widely adopted indicator that represents the performance of the broader Chinese A-share market.

specification based on neural networks (NN).[16] Given the strict short-selling constraints in the China A-share market, we focus on long-only equal weighted portfolios throughout this initial analysis. To take this trading restriction into account, we train the pricing function using the equal weighted long-only E2E model:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & -\tfrac{1}{T}\textstyle\sum_{t=1}^{T} \boldsymbol{w}_t^{\star\top} \boldsymbol{r}_{t+1} \\ \text{s.t.} \quad & \boldsymbol{w}_t^{\star} = \arg\min \left\{ -\boldsymbol{w}_t^{\top} \hat{\boldsymbol{r}}_{t+1} : \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^{\top} \boldsymbol{e} = 1, \boldsymbol{w}_t \leq \frac{10}{N} \right\} \qquad \forall t \in [T]. \end{aligned} \tag{18}$$

Here, the lower-level problem determines a portfolio that assigns equal weights to the top 10% of stocks based on predicted returns $\hat{\boldsymbol{r}}_{t+1} = g(\boldsymbol{x}_t; \boldsymbol{\theta})$, while the upper-level problem aims to find the optimal pricing parameter $\boldsymbol{\theta}$ that maximizes the average portfolio return. The top 10% stocks are implicitly included with equal weights by the last constraint, restricting the maximum holding of the portfolio to $10/N$. Under this constraint, the portfolio selects assets in a descending manner until all the weights are used.

We compare our model with the standard two-stage approach, which learns the pricing function by simply minimizing the averaged MSE loss. For both the E2E and MSE models, their pricing functions are trained on the same training dataset. Next, the trained pricing function $g(\boldsymbol{x}_t; \boldsymbol{\theta}^{\star})$ is applied to the testing set for portfolio construction. We examine equal weighted portfolio performance for both a linear specification (Linear) and nonlinear specifications based on five neural network architectures (NN1 through NN5),[17] under both the MSE and E2E frameworks. Stocks are equally divided into deciles based on predicted returns, from the highest (H) to the lowest (L). H–L denotes the long–short portfolio. Avg and SD refer to the annualized daily average return and standard deviation, respectively, while SR denotes the daily average Sharpe ratio.

The results are summarized in Table 2. For now, we ignore transaction costs, so the results in Table 2 are hypothetical results designed to show the difference between methods. The overall pattern is broadly consistent with the findings in the literature. First, we observe that realized returns generally increase monotonically with the predicted values generated

[16] As noted in Gu et al. (2020), neural networks generally outperform alternative nonlinear models in equity return prediction.

[17] Detailed structures of NN1 to NN5 are presented in Appendix A.2.1. For example, NN1 features a neural network with one hidden layer and 32 neurons; NN2 contains two hidden layers, with 32 neurons in the first and 16 neurons in the second; ... NN5 consists of five hidden layers with 32, 16, 8, 4, and 2, respectively.

by machine learning models, in line with prior studies (Gu et al., 2020; Leippold et al., 2022; Chen et al., 2024). Second, we find that machine learning models consistently outperform linear models (Gu et al., 2020; Martin and Nagel, 2022; Dong et al., 2022) in terms of both long-short (H-L) portfolio average return (Avg) and Sharpe ratio (SR), under both the MSE and E2E frameworks. Among the machine learning models, NN3 consistently shows the strongest performance (Gu et al., 2020).

Although we focus on long only portfolios, we also report long–short (H–L) portfolios where we observe significantly higher Sharpe ratios and returns than in the long-only (H) portfolios. Such high H-L performance is unlikely to be realized in practice due to strict short-selling constraints in the A-share market. In both the E2E and MSE frameworks, the short-only (L) leg consistently delivers better results than the long leg (H), a pattern consistent with prior studies (Stambaugh et al., 2012; Dong et al., 2022; Muravyev et al., 2025), which shows that abnormal returns are often concentrated on the short side. However, these short-leg opportunities are typically harder to exploit because of limits to arbitrage such as high trading frictions and regulatory restrictions.

Within the only-only group (H), we find that the average returns (Avg) and Sharpe ratios (SR) of the E2E framework are consistently higher than those of the MSE approach, regardless of whether the underlying model is linear or nonlinear. Across the six models, the E2E framework improves the average annualized portfolio return from 38.0% to 42.5%, representing an increase of approximately 4.45 percentage points or approximately 12% relative to the MSE benchmark. In terms of the average Sharpe ratio, E2E improves from 1.86 to 2.08. For example, according to the best-performing machine learning specification (NN3), E2E achieves an Avg of 43.6% and an SR of 2.14, compared to 40.0% and 1.94 under MSE. Even under the weakest model specification (Linear/LR), E2E still produces 41.0% and 2.00, while MSE lags further behind at 34.2% and 1.73.

In addition, to assess the statistical significance of the performance differences between the E2E framework and the MSE approach, we conduct mean difference tests on the daily average returns of the long-only portfolios for each model. To account for potential heteroskedasticity and autocorrelation in the daily return series, we compute t-statistics using the Newey and West (1987) procedure. The number of lags is set to the integer part of $T^{1/4}$. The results are reported in the last column of Table A.3, denoted as t-stats. All values are marked with asterisks, indicating that E2E outperforms MSE at the 5% significance level or

Table 2: Portfolio performance of both MSE and E2E

This table reports equal-weighted portfolio performance under the MSE and E2E frameworks across a linear model and five neural network architectures (NN1–NN5). Avg and SD denote the annualized daily average return and standard deviation (in percent), and SR is the annualized Sharpe ratio. We ignore transaction costs so the results are hypothetical. Returns are nominal returns denominated in CNY. Models are trained on daily stock returns using a rolling-window procedure that advances quarterly. Each window spans four years, with 39 months for training, 9 months for validation, and the subsequent quarter for out-of-sample prediction. At the start of each test quarter, stocks are sorted into deciles based on predicted one-quarter-ahead returns, from highest (H) to lowest (L), with H–L representing the corresponding long–short portfolio. The portfolios are rebalanced at each market day from 9:30 AM to 3:00 PM.

		MSE			E2E					MSE			E2E		
	Groups	Avg %	SD %	SR	Avg %	SD %	SR		Groups	Avg %	SD %	SR	Avg %	SD %	SR
Linear	High(H)	34.2	19.8	1.73	41.0	20.5	2.00	NN3	High(H)	40.0	20.6	1.94	43.6	20.4	2.14
	2	25.8	19.3	1.34	28.9	19.5	1.48		2	26.9	19.3	1.39	28.9	19.1	1.51
	3	20.1	18.8	1.07	21.4	19.1	1.11		3	20.9	18.8	1.11	19.6	18.7	1.05
	4	15.5	18.7	0.83	14.7	18.8	0.78		4	16.3	18.4	0.88	15.2	18.5	0.82
	5	12.2	18.5	0.66	9.4	18.6	0.51		5	11.1	18.3	0.61	9.5	18.3	0.52
	6	5.9	18.4	0.32	2.6	18.4	0.14		6	6.4	18.2	0.35	4.5	18.3	0.25
	7	-0.3	18.3	-0.02	-2.3	18.2	-0.13		7	1.5	18.0	0.08	-0.4	18.2	-0.02
	8	-6.4	18.1	-0.35	-10.0	17.9	-0.56		8	-6.7	18.2	-0.37	-7.5	18.2	-0.41
	9	-17.5	18.1	-0.97	-21.0	17.8	-1.18		9	-20.2	18.2	-1.11	-20.4	18.2	-1.12
	Low(L)	-59.5	18.8	-3.16	-54.9	18.3	-3.00		Low(L)	-66.2	19.2	-3.45	-63.1	19.1	-3.30
	H-L	46.9	6.2	7.55	47.9	6.6	7.27		H-L	53.1	6.0	8.88	54.9	5.9	9.05
NN1	High(H)	39.4	20.4	1.93	43.4	20.4	2.12	NN4	High(H)	38.3	20.8	1.84	41.7	20.5	2.04
	2	26.5	19.2	1.38	28.5	19.1	1.50		2	28.2	19.4	1.45	28.1	19.3	1.46
	3	21.7	18.8	1.16	21.0	18.7	1.12		3	21.4	18.9	1.13	19.2	18.7	1.03
	4	16.5	18.5	0.89	14.5	18.4	0.79		4	15.7	18.6	0.84	11.2	18.5	0.61
	5	12.2	18.3	0.66	9.6	18.4	0.53		5	11.5	18.4	0.62	9.3	18.2	0.51
	6	6.8	18.1	0.37	4.8	18.2	0.26		6	7.2	18.1	0.39	5.2	18.1	0.29
	7	1.1	18.1	0.06	-1.6	18.1	-0.09		7	1.1	18.0	0.06	0.2	18.2	0.01
	8	-6.8	18.1	-0.38	-8.6	18.2	-0.47		8	-5.7	18.1	-0.32	-6.7	18.1	-0.37
	9	-18.5	18.4	-1.01	-20.8	18.2	-1.14		9	-18.7	18.2	-1.03	-19.5	18.3	-1.06
	Low(L)	-68.8	19.3	-3.57	-61.0	19.0	-3.20		Low(L)	-68.9	19.3	-3.57	-58.8	19.3	-3.05
	H-L	54.1	6.1	8.87	52.2	5.9	8.85		H-L	53.6	6.2	8.61	50.2	5.6	8.98
NN2	High(H)	37.2	20.6	1.81	41.9	20.4	2.06	NN5	High(H)	38.9	20.5	1.90	43.2	20.6	2.10
	2	28.0	19.3	1.46	28.1	19.2	1.46		2	28.1	19.3	1.46	26.5	19.2	1.38
	3	21.9	18.8	1.17	21.1	18.8	1.12		3	21.3	18.8	1.13	18.1	18.7	0.96
	4	16.2	18.5	0.88	14.9	18.4	0.81		4	15.5	18.6	0.84	12.6	18.4	0.69
	5	11.3	18.2	0.62	10.1	18.2	0.55		5	10.8	18.3	0.59	8.5	18.2	0.46
	6	6.2	18.2	0.34	4.5	18.1	0.25		6	7.5	18.3	0.41	4.4	18.2	0.24
	7	0.9	18.1	0.05	-0.2	18.1	-0.01		7	1.5	18.2	0.08	2.2	18.2	0.12
	8	-7.4	18.1	-0.41	-7.2	18.1	-0.40		8	-5.0	18.1	-0.28	-6.2	18.4	-0.34
	9	-17.9	18.3	-0.98	-20.7	18.2	-1.14		9	-21.6	18.2	-1.18	-20.2	18.4	-1.10
	Low(L)	-66.5	19.3	-3.44	-62.5	19.2	-3.25		Low(L)	-67.0	19.5	-3.43	-59.1	19.5	-3.03
	H-L	51.8	6.0	8.58	52.2	5.8	8.97		H-L	52.9	6.1	8.65	51.2	6.0	8.58

better across all six models. We also report additional performance metrics, such as turnover rate and maximum drawdown, for the long-only (High) portfolios in Table A.3. To ensure that our findings are not driven by illiquid micro-cap stocks, we also conduct a robustness check excluding the bottom 30% of firms by market capitalization. The results, reported in Appendix A.4.3, show that the performance dominance of the E2E framework remains robust to this exclusion.

Taken together, the results indicate that while MSE-based models demonstrate strong statistical predictive performance, particularly on the short side, their practical implementation is limited by constraints such as short-selling restrictions. In contrast, the E2E framework is designed to incorporate trading restrictions during training, resulting in more robust and implementable investment strategies.

4.3 Transaction costs

Both the E2E and MSE long-only portfolios exhibit high turnover rates, which can lead to significant transaction costs in real-world investment settings, thereby eroding the profitability of the trading signals. This underscores the necessity of adaptively accounting for these trading frictions in the investment process.

Due to the high turnover rate, in practice, investors rarely deploy the raw long-portfolio implied by the signal directly.[18] Instead, they usually include transaction costs into the portfolio construction stage, leading to the following post-signal optimization problem

$$\begin{aligned} \min_{\boldsymbol{w}_t} \quad & -\boldsymbol{w}_t^\top \hat{\boldsymbol{r}}_{t+1} + \tfrac{\gamma}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1 \\ \text{s.t.} \quad & \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^\top \boldsymbol{e} = 1, \boldsymbol{w}_t \leq \tfrac{10}{N}. \end{aligned} \tag{19}$$

Here, γ denotes a uniform trading cost rate, and $\|\boldsymbol{w}_t^\star - \boldsymbol{w}_{t-1}^\star\|_1$ represent the two-side portfolio turnover rate of the adjacent equal weighted long-only portfolios (Zhou et al., 2024)[19]. This formulation manages the trade-off between maximizing predicted returns and minimizing transaction costs. The turnover penalty ensures that any change in portfolio weights is only

[18] We assume rebalancing from 9:30 AM to 3:00 PM each market day, using returns corresponding to the value-weighted-average-price (VWAP).

[19] Other turnover measures such as $\left\| \boldsymbol{w}_t^\star - \frac{\boldsymbol{w}_{t-1}^\star \odot (\mathbf{1}+\boldsymbol{r}_t)}{1+\boldsymbol{r}_t{}^\top \boldsymbol{w}_{t-1}^\star} \right\|_1$ or $\left\| \boldsymbol{w}_t^\star - \boldsymbol{w}_{t-1}^\star \odot (\mathbf{1}+\boldsymbol{r}_t) \right\|_1$ could also be employed to account for the drift in previous weights due to returns. Given our daily rebalancing frequency, the impact of intra-period price drift on weight changes is typically small; hence, we adopt this simpler measure.

executed if the expected gain in returns is large enough to offset the incurred trading friction. By controlling portfolio-level turnover, this **MSE-Opt** approach filters out marginal trades and prevents alpha erosion by excessive transaction costs, leading to more practical and cost-efficient investment strategies.

Although incorporating transaction costs into the downstream portfolio optimization stage can reduce final turnover, its effectiveness is limited by the disconnect between prediction and optimization. The prediction model is trained solely to minimize a statistical error like MSE, without any awareness of the transaction costs to be imposed in the subsequent construction phase. As a result, it generates an identical pricing function and produces the same raw signals regardless of the transaction cost rates (TCRs). This inability to adapt is an important drawback, as the model cannot learn to favor signals with lower inherent turnover when trading frictions are high. To address this shortcoming, we introduce the following transaction-cost-aware E2E model, which integrates cost sensitivity directly into the training process:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \quad \tfrac{1}{T}\sum_{t=1}^{T} -\boldsymbol{w}_t^{\star\top}\boldsymbol{r}_{t+1} + \tfrac{\gamma}{2}\|\boldsymbol{w}_t^{\star} - \boldsymbol{w}_{t-1}^{\star}\|_1 \\ &\text{s.t.} \quad \boldsymbol{w}_t^{\star} = \arg\min\left\{-\boldsymbol{w}_t^{\top}\hat{\boldsymbol{r}}_{t+1} + \frac{\gamma}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1 : \boldsymbol{w}_t \in \mathbb{R}_+^N, \boldsymbol{w}_t^{\top}\boldsymbol{e} = 1, \boldsymbol{w}_t \le \frac{10}{N}\right\} \qquad \forall t \in [T]. \end{aligned} \tag{20}$$

To evaluate how varying transaction cost rates (TCRs) affect portfolio performance, we compare three frameworks:

- **MSE-Top**: A baseline framework in which we train the pricing function based on MSE. Next, we form a long-only portfolio directly from the raw predictions of the standard MSE model, without any subsequent transaction cost optimization.
- **MSE-Opt**: The training of the MSE signal is the same as the above method; however, the raw predictions are followed by a downstream transaction cost optimization problem.
- **E2E**: Our proposed end-to-end framework, which integrates asset pricing with the downstream transaction cost optimization problem.

Due to computational constraints, we fix NN3 as the prediction model for all approaches.[20]

[20]Our adoption of a rolling-window framework necessitates retraining the neural network 40 times for each

The NN3 architecture was first adopted in the asset pricing literature by Gu et al. (2020) and has since been widely applied, serving as a benchmark in Avramov et al. (2023) and as a base prediction model in Chen et al. (2024). We evaluate these strategies in a range of TCRs: 0, 10, 20, 30 and 40 basis points (bps). This testing range comprehensively covers the typical trading frictions observed in the Chinese A-share market. An investor in the Chinese stock market faces two fixed trading costs: a stamp tax and a commission fee. The stamp tax is a one-sided fee applied only to the seller. It was held constant at 10 bps from 2008 and lowered to 5 bps since August 2023. The commission fee for institutional investors was approximately 5 bps in 2012, but it decreased rapidly in subsequent years. In recent years, the commission fee for retail investors has typically ranged between 2 and 3 bps, while institutional investors usually enjoy even lower rates (Leippold et al., 2022). Hence, a reasonable estimate of TCR is between 10 to 20 bps.[21]

For simplicity in our empirical baseline, we assume a uniform transaction cost rate across all securities. However, it is important to emphasize that our end-to-end framework is sufficiently flexible to incorporate heterogeneous transaction costs, including security-specific market impact costs. In practice, trading frictions are inherently investor-driven and depend on trade size, the liquidity profile of the underlying assets, and trading execution skill. Those utilizing our approach can seamlessly replace the fixed transaction cost rates (TCRs) with asset-specific estimates based on non-linear impact functions tailored to their unique operational profiles.

To examine the impact of varying TCRs, we report a range of evaluation metrics. TO-P and TO-S refer to portfolio turnover and signal turnover. TO-P captures the turnover achieved after cost-aware portfolio optimization, while TO-S measures the signal-implied

of the five trading cost rates γ tested in Table 3, resulting in a total of 200 training runs. Each training process uses a dataset of approximately 1,000 cross-sectional observations, requiring substantial GPU memory and computational time. Given these resource demands, a comprehensive examination of all potential prediction models is computationally prohibitive given our limited computing resources.

[21]We adopt the full-day VWAP as the execution price, which implies an algorithmic execution strategy that distributes trading volumes continuously throughout the entire trading session. Unlike instantaneous execution strategies (e.g., open-to-open) that require immediate liquidity and often cross the bid-ask spread, a VWAP-based approach allows the use of passive limit orders. This significantly mitigates the market impact and reduces the effective bid-ask spread costs, making the fixed costs (stamp duty and commissions) the primary component of total transaction costs. However, we extend our analysis to higher levels (up to 40 bps) to account for scenarios with higher implicit transaction costs, such as price impacts from large trades or wider bid-ask spreads during periods of low liquidity. This allows us to stress-test the robustness of each framework under more conservative and realistic cost assumptions.

turnover, calculated based on the top 10% of stocks ranked by predicted returns, before any cost adjustments. MDD-1D indicates the maximum single-day drawdown. MDD, MDD Begin, MDD End, and MDD Length capture the magnitude, start and end dates, and duration of the largest drawdown during the sample period. We also report t-statistics (t-stats) for mean difference tests of daily average portfolio returns between each pair of models.

We document several interesting findings in Table 3. First, unsurprisingly, across all three frameworks, higher TCRs lead to lower portfolio performance, and both annualized return (Ann Ret) and Sharpe ratio (Ann SR) decline, while risk (Ann SD) increases slightly. In parallel, both TO-P and TO-S fall as models become more conservative in trading, and maximum drawdowns grow more severe. Notably, the portfolio return of MSE-Top turns negative at TCRs of 30 and 40 basis points.

Second, the proposed E2E framework consistently outperforms both MSE-based benchmarks in all cost settings. While MSE-Opt incorporates a cost-aware optimization step after signal generation, reflecting a common industry practice, its performance remains meaningfully below that of E2E. Compared to MSE-Opt, E2E delivers about 3–4 percentage points higher annualized return, and the return improvements are statistically significant at the 5% level or better, with t-statistics ranging from 1.74 to 3.45 across different levels of TCR. E2E also achieves a Sharpe ratio of roughly 0.2 higher. The performance gap between E2E and MSE-Top increases even more, both economically and statistically, as TCR increases.

Third, while both MSE-Opt and E2E adjust to rising TCRs by reducing TO-P, only E2E responds at the signal level (TO-S). In MSE-Opt, signal turnover (TO-S) remains unchanged because the pricing model is trained without knowledge of transaction costs, and cost adjustments occur only at the portfolio optimization stage. By contrast, the E2E framework reduces both TO-P and TO-S as TCR increases, indicating that it internalizes cost sensitivity not just in post-prediction optimization but also during signal formation. This dual adjustment mechanism reflects the benefit of jointly learning pricing functions and cost-aware portfolio decisions in an integrated framework. We also provide a factor importance visualization in Appendix A.4.4.

Table 3: Long-only portfolio performance after accounting for transaction costs

This table reports the long-only portfolio performance using the NN3 model, after accounting for different levels of transaction cost rates (TCRs) in basis points. MSE-Top serves as a baseline, forming long-only portfolios directly from the raw predictions of the MSE model without accounting for transaction costs. MSE-Opt uses the same MSE signal but adds a second-stage optimization, considering TCRs. E2E refers to our proposed end-to-end framework, which incorporates cost awareness directly into the training of the pricing function. Ann Ret, Ann SD, and Ann SR denote the annualized return, standard deviation, and Sharpe ratio (in percent), respectively. TO-P and TO-S refer to one-sided portfolio turnover and signal turnover. TO-P captures the realized turnover after cost-aware portfolio optimization, while TO-S measures signal-implied turnover, computed based on the top 10% of stocks ranked by predicted returns, prior to any cost adjustments. MDD-1D indicates the maximum single-day drawdown. MDD, MDD Begin, MDD End, and MDD Length capture the magnitude, start and end dates, and duration of the largest drawdown over the sample period. t-Stats report the mean difference tests for daily average portfolio returns between one pair of two models, where $**$ indicates that the latter method outperforms the former at the 5% significance level or better. Within each cell, the first value indicates the difference between MSE-Top and MSE-Opt, and the second value for MSE-Opt and E2E.

TCR	Method	Ann Ret (%)	Ann SD (%)	Ann SR	TO-P (%)	TO-S (%)	MDD-1D (%)	MDD (%)	MDD Begin	MDD End	Length	t-stats
0	MSE-Top	40.0	20.6	1.94	60.6	60.6	8.27	22.42	2015-06-19	2015-07-06	18	
	MSE-Opt	40.0	20.6	1.94	60.6	60.6	8.27	22.42	2015-06-19	2015-07-06	18	0
	E2E	43.6	20.4	2.14	58.6	58.6	8.31	23.59	2015-06-19	2015-07-06	18	1.74**
10	MSE-Top	25.3	20.6	1.23	60.6	60.6	8.27	23.03	2015-06-19	2015-07-06	18	
	MSE-Opt	29.2	20.7	1.41	41.7	60.6	8.20	22.54	2015-06-10	2015-07-06	27	7.54**
	E2E	33.7	20.5	1.64	31.6	53.8	8.17	22.91	2015-06-10	2015-07-06	27	3.45**
20	MSE-Top	10.6	20.6	0.51	60.6	60.6	8.27	44.71	2016-11-18	2022-04-26	1986	
	MSE-Opt	24.0	20.8	1.16	30.3	60.6	8.27	22.79	2015-06-10	2015-07-06	27	16.87**
	E2E	29.1	20.4	1.42	18.7	52.8	8.20	22.07	2015-06-19	2015-07-06	18	3.30**
30	MSE-Top	-4.2	20.6	-0.20	60.6	60.6	8.27	134.47	2016-11-18	2023-10-19	2527	
	MSE-Opt	20.1	20.8	0.97	22.5	60.6	8.36	24.42	2015-06-10	2015-07-06	27	23.68**
	E2E	23.0	20.4	1.12	13.3	51.0	8.50	25.24	2015-06-10	2015-07-06	27	1.83**
40	MSE-Top	-18.9	20.6	-0.91	60.6	60.6	8.27	235.76	2016-06-23	2023-12-22	2739	
	MSE-Opt	18.1	20.7	0.87	16.9	60.6	8.46	25.49	2015-06-10	2018-02-05	445	30.50**
	E2E	21.8	20.4	1.07	10.8	50.5	8.60	26.99	2015-06-10	2015-07-06	27	2.14**

4.4 Role of investors' risk preferences

In this section, we examine the role of investor heterogeneity in risk preferences in model prediction and portfolio construction. The early work of Constantinides (1982) emphasized the importance of investor heterogeneity, such as differing attitudes toward investment risk. In the tradition approach, the first-stage model is trained to predict returns across the full cross-section of stocks, assuming a homogeneous level of risk aversion. Individual investors then apply a downstream optimization to construct a portfolio based on their own preferences, but do so using a shared prediction model that is not trained with their specific objectives in mind. The E2E framework aligns the learning objective with each investor's risk-return trade-off, the framework generates a tailored pricing function that better aligns with the heterogeneous investment goals.

We solve the following risk-aware E2E model:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \tfrac{1}{T}\sum_{t=1}^{T} -\boldsymbol{w}_t^{\star\top}\boldsymbol{r}_{t+1} + \tfrac{\gamma}{2}\|\boldsymbol{w}_t^{\star} - \boldsymbol{w}_{t-1}^{\star}\|_1 + \eta\sqrt{\boldsymbol{w}_t^{\star\top}\hat{\boldsymbol{\Sigma}}_t\boldsymbol{w}_t^{\star}} \\ \text{s.t.} \quad & \boldsymbol{w}_t^{\star} = \arg\min \left\{ -\boldsymbol{w}_t^{\top}\hat{\boldsymbol{r}}_{t+1} + \frac{\gamma}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_1 + \eta\sqrt{\boldsymbol{w}_t^{\top}\hat{\boldsymbol{\Sigma}}_t\boldsymbol{w}_t} : \begin{array}{l} \boldsymbol{w}_t \in \mathbb{R}_+^N, \\ \boldsymbol{w}_t^{\top}\boldsymbol{e} = 1, \\ \boldsymbol{w}_t \le 0.1 \end{array} \right\} \quad \forall t \in [T], \end{aligned} \tag{21}$$

where η controls the level of risk tolerance and γ denotes the transaction cost rate. In the above optimization problem, the lower-level problem determines optimal portfolio allocations based on predicted returns $\hat{\boldsymbol{r}}_{t+1}$ and empirical standard deviation (SD) estimates $\hat{\boldsymbol{\Sigma}}_t$, while the upper-level problem optimizes the pricing parameter $\boldsymbol{\theta}$ to maximize the realized mean–SD utility. The constraint $\boldsymbol{w}_t \le 0.1$ ensures there are no extreme positions in portfolios.

We choose a mean–standard deviation (Mean–SD) optimization framework rather than the traditional mean–variance (Mean–Var) approach for several reasons. First, in the forward propagation, we need to solve high-dimensional portfolio optimization problems. Modern large-scale optimization solvers are fundamentally built upon conic programming. Since the Mean–SD formulation is naturally cast as a second-order cone program (SOCP), it enjoys inherent compatibility with these solvers, allowing for more direct and numerically robust solutions. Second, the Mean–SD formulation also stabilizes the backward pass. In our framework, gradients of the optimal weights with respect to input parameters are computed via

the Implicit Function Theorem. For a traditional Mean–Var objective, this requires inverting the Hessian ($2\eta\boldsymbol{\Sigma}$). In high-dimensional settings, $\boldsymbol{\Sigma}$ could be ill-conditioned with small eigenvalues, causing the inverse to produce exploding gradients that destabilize training. In contrast, the Mean–SD objective introduces the portfolio standard deviation as a scaling factor in the derivative's denominator. This term acts as a form of adaptive regularization, dampening gradient sensitivity to conditioning issues and ensuring better convergence.

Importantly, these computational advantages come at no theoretical cost. For any given set of expected returns and a covariance matrix, the Mean–SD and Mean–Var frameworks generate the same efficient frontier. That is, any portfolio on the Mean–Var efficient frontier can also be found on the Mean–SD efficient frontier, and vice versa. Therefore, employing the Mean–SD framework does not sacrifice the theoretical foundations of portfolio selection established by Markowitz (1952).

In our implementation, the risk component of the portfolio objective requires a covariance matrix $\hat{\boldsymbol{\Sigma}}$. For this, we use a one-year look-back window of daily returns to estimate asset volatilities. However, we adopt a simplified approach that considers only the diagonal components of the covariance matrix. This simplified covariance structure is a deliberate design choice. We compare all three methods with the same simplified covariance matrix.

There are other reasons to focus on the expected returns generation process. First, our analysis is based on returns in excess of the market. By removing the dominant market factor, the remaining residual returns exhibit significantly reduced absolute correlations compared to raw returns. Within a domestic equity universe, unlike multi-asset portfolios containing fixed income or commodities, idiosyncratic risk tends to dominate the residual component. Second, the expected return generating process has a larger influence on portfolio performance than the covariance estimation. By simplifying the covariance matrix, we ensure a fair comparison between the E2E and benchmark models while simultaneously enhancing the computational efficiency of the training pipeline. Finally, and most importantly, this simplification is just a convenient choice – our E2E framework is general enough to incorporate full covariance matrices, as well as alternative utility functions.

To comprehensively evaluate our risk-aware E2E model, we compare it with two distinct benchmarks. The first is the traditional two-stage "predict-then-optimize" paradigm, referred to as the MSE-Opt approach. This method trains a prediction model to minimize MSE and then treats the forecasts as fixed inputs for a subsequent Mean-SD optimization.

The second benchmark is the parametric portfolio policy (PPP) method proposed by Brandt et al. (2009). Unlike the two-stage approach, PPP models portfolio weights directly as a function of firm characteristics, bypassing the explicit estimation of expected returns. By maximizing the investor's utility function directly, PPP represents a class of decision-aware models that directly maps from the characteristic space to the decision space. However, it does not incorporate an optimization layer; consequently, the final portfolio is mechanically determined by the model inputs and parameter values. We implement PPP using the same neural network architecture to ensure a fair comparison, training it to directly output weights that maximize the same Mean-SD utility.[22]

Throughout this section's analysis, we maintain the long-only portfolio constraint to ensure practical relevance in our empirical application of the Chinese market. We first assume zero transaction costs with the aim of isolating the impact of heterogeneous risk tolerance. We then impose a transaction cost of 20 bps to evaluate the strategy's performance under realistic trading frictions. The performance of the E2E framework alongside the MSE-Opt and PPP benchmarks is reported in Table 4. The results show that the E2E method significantly outperforms MSE and PPP in the higher risk tolerance ranges.

The relative performance of the benchmarks exhibits a clear crossover depending on risk tolerance. In low risk-aversion regimes, PPP outperforms MSE-Opt. This advantage stems from PPP's decision-aware objective, which identifies high-return opportunities that a purely statistical MSE loss might overlook. However, as risk aversion increases, MSE-Opt regains dominance. This reversal highlights a structural limitation of PPP: by mapping characteristics directly to weights, it lacks an intermediate optimization step to strictly enforce variance minimization. In contrast, MSE-Opt incorporates an optimization problem that precisely manages portfolio structure. As a result, in the high risk aversion regime, MSE-Opt demonstrates superior performance and yields a higher return at any given level of risk, regardless of whether transaction costs are considered.

The E2E framework effectively synthesizes the strengths of these two paradigms. By in-

[22]While the original PPP framework in Brandt et al. (2009) assumes a linear mapping from characteristics to portfolio weights, recent literature has extended this approach to incorporate deep neural networks and demonstrated better performance (Feng et al., 2023; Simon et al., 2025). We adopt this idea using the same NN3 architecture as our E2E model. This ensures a fair comparison by controlling the prediction model, implying that performance differences arise from the structure (adding an optimization layer vs. direct mapping) rather than the complexity of the neural network.

tegrating the explicit optimization layer into the learning process, E2E retains the structural precision of MSE-Opt, which enables it to converge to the minimum-variance portfolio when η is high. Simultaneously, it leverages the decision-aware learning capability of PPP to maximize returns when η is low. This feature allows E2E to generate a superior efficient frontier that encompasses the strengths of both traditional and parametric approaches, dominating the benchmarks in both gross and net performance.

To assess the statistical significance of the E2E framework's outperformance, we conduct two multivariate tests spanning the full spectrum of risk aversion levels. We first implement a volatility-matched test to compare risk-adjusted returns. By scaling portfolios to a fixed annualized volatility target (20%), we isolate the excess return generated solely by asset selection, independent of overall risk exposure. Second, we test the pooled difference in realized utility, which serves as a direct, non-scalable measure of investor objective. The results are reported in Table 5. We find that the E2E framework generates statistically significant volatility-matched alphas against both the MSE-Opt and PPP benchmarks. Furthermore, the tests for realized utility are consistent with the hypothesis that the E2E framework delivers a significant gain across the risk spectrum.

Figure 2 provides a visual comparison of each method's realized performance frontiers.[23] The plot illustrates that the E2E frontier (blue line) dominates both the PPP frontier (green line) and the MSE-Opt frontier (red line). For any given level of realized risk, the E2E approach generates a portfolio with a higher realized return. Moreover, the visual gap between the gross (dashed lines) and net (solid lines) frontiers underscores the substantial impact of transaction costs, which erode a significant portion of profits. This marked reduction reinforces our earlier finding that high-turnover strategies face severe implementation hurdles. In addition, the PPP frontier lies above the MSE frontier in the high-return region but falls behind as volatility decreases, visually confirming the dynamic relationship discussed above. The E2E framework pushes the frontier even further outward, highlighting its unique ability to learn return-maximizing strategies for risk-tolerant investors.

[23]Since our study employs a rolling-window analysis, the ex-ante efficient frontier is time-varying and dynamic. Therefore, to evaluate the consistent efficacy of our approach, we present the ex-post (realized) performance frontier.

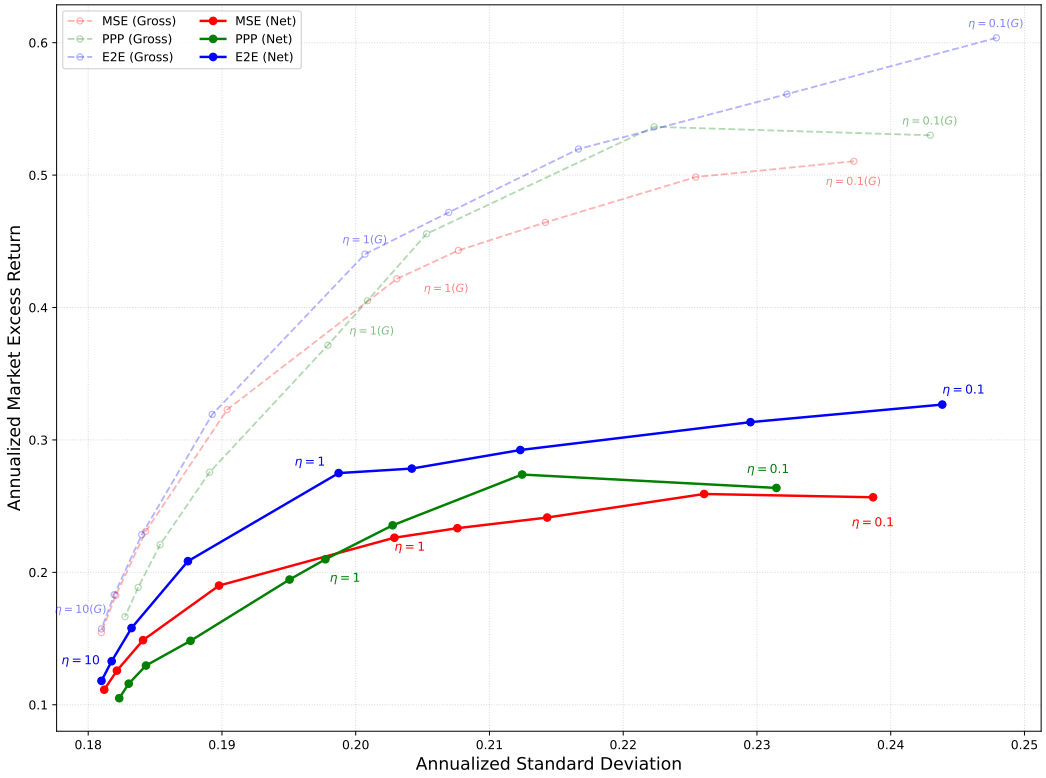


Figure 2: Realized Efficient Frontier for E2E, PPP and MSE frameworks

This figure plots the realized out-of-sample performance frontiers for the E2E, PPP, and MSE-Opt frameworks, all using the NN3 prediction model. This is similar to the efficient frontier except the curve represents ex post outcomes. Each point on the frontiers represents the annual market excess return and annual standard deviation pair for an optimal portfolio from Table 4, corresponding to a specific level of investor risk aversion (η). The blue dots and line represent the E2E framework, while the red dots and line represent the MSE-Opt framework. Solid lines denote net performance (assuming 20 bps transaction costs), while dashed lines denote gross performance (assuming zero transaction costs). All portfolios are subject to a long-only constraint.

Table 4: Portfolio performance with investors' different risk preferences

This table reports portfolio performance with the NN3 model across varying levels of investor risk aversion, denoted by η. The higher value of η indicates the greater risk aversion (i.e., more conservative investors). Daily Ret and Daily SD refer to the average daily return and standard deviation (in percent), respectively. Ann Ret, Ann SD, and Ann SR represent the annualized return, standard deviation, and Sharpe ratio. We rebalance at each market day from 9:30 AM to 3:00 PM. All portfolios are subject to a long-only constraint and a max participation constraint. The transaction costs are assumed to be either zero (Gross) and 20bps (Net).

		Daily Ret (%)		Daily Vol (%)		Utility (%)		Ann Ret (%)		Ann Vol (%)		Sharpe	
η	Method	Gross	Net	Gross	Net	Gross	Net	Gross	Net	Gross	Net	Gross	Net
0.1	MSE	0.21	0.11	1.52	1.53	0.06	-0.05	51.0	25.7	23.7	23.9	2.15	1.08
	PPP	0.22	0.11	1.56	1.48	0.06	-0.04	53.0	26.4	24.3	23.1	2.18	1.14
	E2E	0.25	0.13	1.59	1.56	0.09	-0.02	60.4	32.7	24.8	24.4	2.44	1.34
0.25	MSE	0.21	0.11	1.45	1.45	-0.16	-0.26	49.9	25.9	22.5	22.6	2.21	1.15
	PPP	0.22	0.11	1.43	1.36	-0.14	-0.23	53.6	27.4	22.2	21.2	2.41	1.29
	E2E	0.23	0.13	1.49	1.47	-0.14	-0.24	56.1	31.3	23.2	22.9	2.42	1.37
0.5	MSE	0.19	0.10	1.37	1.37	-0.50	-0.59	46.4	24.1	21.4	21.4	2.17	1.13
	PPP	0.19	0.10	1.32	1.30	-0.47	-0.55	45.6	23.6	20.5	20.3	2.22	1.16
	E2E	0.21	0.12	1.39	1.36	-0.48	-0.56	52.0	29.2	21.7	21.2	2.40	1.38
0.75	MSE	0.18	0.10	1.33	1.33	-0.82	-0.90	44.3	23.3	20.8	20.8	2.13	1.12
	PPP	0.17	0.09	1.29	1.27	-0.80	-0.86	40.5	21.0	20.1	19.8	2.02	1.06
	E2E	0.19	0.11	1.33	1.31	-0.80	-0.87	47.2	27.8	20.7	20.4	2.28	1.36
1.0	MSE	0.17	0.09	1.30	1.30	-1.13	-1.21	42.2	22.6	20.3	20.3	2.08	1.11
	PPP	0.15	0.08	1.27	1.25	-1.12	-1.17	37.1	19.5	19.8	19.5	1.88	1.00
	E2E	0.18	0.11	1.29	1.27	-1.11	-1.16	44.0	27.5	20.1	19.9	2.19	1.38
2.5	MSE	0.13	0.08	1.22	1.22	-2.92	-2.97	32.3	19.0	19.0	19.0	1.70	1.00
	PPP	0.11	0.06	1.21	1.20	-2.92	-2.95	27.6	14.8	18.9	18.8	1.46	0.79
	E2E	0.13	0.09	1.21	1.20	-2.90	-2.92	31.9	20.8	18.9	18.8	1.69	1.11
5.0	MSE	0.10	0.06	1.18	1.18	-5.82	-5.84	23.1	14.9	18.4	18.4	1.25	0.81
	PPP	0.09	0.05	1.19	1.18	-5.86	-5.86	22.1	13.0	18.5	18.4	1.19	0.70
	E2E	0.09	0.06	1.18	1.18	-5.81	-5.81	22.9	15.8	18.4	18.3	1.24	0.86
7.5	MSE	0.08	0.05	1.17	1.17	-8.68	-8.71	18.3	12.6	18.2	18.2	1.00	0.69
	PPP	0.08	0.05	1.18	1.17	-8.76	-8.76	18.9	11.6	18.4	18.3	1.03	0.63
	E2E	0.08	0.05	1.17	1.17	-8.68	-8.69	18.3	13.3	18.2	18.2	1.01	0.73
10.0	MSE	0.06	0.05	1.16	1.16	-11.55	-11.58	15.4	11.1	18.1	18.1	0.85	0.61
	PPP	0.07	0.04	1.17	1.17	-11.65	-11.65	16.7	10.5	18.3	18.2	0.91	0.58
	E2E	0.06	0.05	1.16	1.16	-11.55	-11.56	15.7	11.8	18.1	18.1	0.87	0.65

Table 5: Comparisons of the E2E, PPP and MSE frameworks: Statistical Tests

This table reports the results of two multivariate tests examining the performance of the E2E framework relative to the MSE-Opt and PPP benchmarks across nine risk aversion levels ($\eta \in \{0.1, 0.25, 0.5, 0.75, 1.0, 2.5, 5.0, 7.5, 10.0\}$). The left panel reports the daily pooled difference in returns and its t-statistic, obtained by calculating the difference between the volatility-scaled E2E and benchmark returns (scaled to a 20% annualized volatility). The right panel reports the daily pooled difference in realized utility and its t-statistic, obtained by testing the mean difference of the daily utility series ($u_t = r_t - \eta\sigma$). The results are reported for both Gross returns (assuming zero transaction costs) and Net returns (assuming 20 bps transaction costs).

		Volatility-Matched Return		**Realized Utility**	
Type	**Comparison**	**Difference (%)**	***t*-statistic**	**Difference (%)**	***t*-statistic**
Gross	E2E vs. MSE	0.01	2.43	0.02	3.60
	E2E vs. PPP	0.01	2.20	0.03	5.39
Net	E2E vs. MSE	0.01	3.80	0.03	7.56
	E2E vs. PPP	0.02	3.71	0.03	5.41

5 Conclusion

This paper proposes an end-to-end (E2E) learning framework for portfolio construction enabled by machine learning tools. Unlike the conventional two-stage approach that separates expected return estimation from portfolio implementation, the E2E framework directly aligns asset returns prediction with investor objectives and market frictions. The empirical work suggests that our E2E model consistently outperforms the traditional mean-squared error-based benchmark in a range of settings, including neural network and linear specifications, long-only portfolios, and environments with realistic transaction costs and structural constraints.

Our results highlight the flexibility and robustness of the E2E framework. It not only delivers higher out-of-sample returns and Sharpe ratios in the presence of trading frictions in our empirical tests but also adapts meaningfully to investor heterogeneity in risk preferences, structural portfolio constraints, and market microstructure features. The E2E model dynamically adjusts its signal composition in response to cost pressures and investor risk aversion, generating feasible and preference-aligned portfolios along a smooth and well-behaved efficient frontier. Taken together, these findings suggest that integrating learning and decision-making under a unified objective yields economically significant gains and offers a promising direction to bridge the gap between predictive modeling and portfolio implementation.

While our implementation is able to embrace investor risk heterogeneity as well as market frictions, there are at least two promising, unexplored applications of our E2E approach. First, E2E can take into account the specific tax profile of the investor as well as the tax status of each position in the portfolio. This status is important in deciding weight changes (e.g., tax loss harvesting). Second, most portfolio management research focuses on asset allocation and ignores the liability side. The giant world pension complex faces a joint asset-liability optimization given that they often have end-of-month obligations to pensioners. The E2E approach is ideally suited to handle this complexity. These important applications are part of our on-going research.

References

Agrawal, Akshay, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter, 2019, Differentiable convex optimization layers, *Advances in Neural Information Processing Systems* 32.

Ammann, Manuel, Guillaume Coqueret, and Jan-Philip Schade, 2016, Characteristics-based portfolio choice with leverage constraints, *Journal of Banking and Finance* 70, 23–37.

Amos, Brandon, and J. Zico Kolter, 2017, Optnet: Differentiable optimization as a layer in neural networks, *International Conference on Machine Learning* 136–145.

Arulkumaran, Kai, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath, 2017, Deep reinforcement learning: A brief survey, *IEEE Signal Processing Magazine* 34 (6), 26–38.

Avramov, Doron, Si Cheng, and Lior Metzker, 2023, Machine learning vs. economic restrictions: Evidence from stock return predictability, *Management Science* 69 (5), 2587–2619.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton, 2016, Layer normalization, *arXiv preprint arXiv:1607.06450*.

Bagnara, Matteo, 2024, Asset pricing and machine learning: A critical review, *Journal of Economic Surveys* 38 (1), 27–56.

Bishop, Christopher M., and Nasser M. Nasrabadi, 2006, *Pattern Recognition and Machine Learning* (Springer).

Blanchet, Jose, Lin Chen, and Xun Yu Zhou, 2022, Distributionally robust mean-variance portfolio selection with Wasserstein distances, *Management Science* 68 (9), 6382–6410.

Blondel, Mathieu, André FT Martins, and Vlad Niculae, 2020, Learning with Fenchel-young losses, *Journal of Machine Learning Research* 21 (35), 1–69.

Boyd, Stephen, and Lieven Vandenberghe, 2004, *Convex Optimization* (Cambridge University Press).

Brandt, Michael W., Pedro Santa-Clara, and Rossen Valkanov, 2009, Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns, *Review of Financial Studies* 22 (9), 3411–3447.

Butler, Andrew, and Roy H. Kwon, 2023, Integrating prediction in mean-variance portfolio optimization, *Quantitative Finance* 23 (3), 429–452.

Cao, Guanghe, Yitian Zhang, Qi Lou, and Gaike Wang, 2024, Optimization of high-frequency trading strategies using deep reinforcement learning, *Journal of Artificial Intelligence General Science* 6 (1), 230–257.

Chen, Andrew Y., and Jack McCoy, 2024, Missing values handling for machine learning portfolios, *Journal of Financial Economics* 155, 103815.

Chen, Luyang, Markus Pelger, and Jason Zhu, 2024, Deep learning in asset pricing, *Management Science* 70 (2), 714–750.

Chinco, Alex, Andreas Neuhierl, and Michael Weber, 2021, Estimating the anomaly base rate, *Journal of Financial Economics* 140 (1), 101–126.

Choi, Darwin, Wenxi Jiang, and Chao Zhang, 2024, Alpha go everywhere: Machine learning and international stock returns, *Working Paper, HKUST*.

Cochrane, John H., 2011, Presidential address: Discount rates, *Journal of Finance* 66 (4), 1047–1108.

Colson, Benoît, Patrice Marcotte, and Gilles Savard, 2005, Bilevel programming: A survey, *4or* 3 (2), 87–107.

Cong, Lin William, Ke Tang, Jingyuan Wang, and Yang Zhang, 2021, Alphaportfolio: Direct construction through deep reinforcement learning and interpretable AI, *Working Paper, Cornell University*.

Constantinides, George M., 1979, Multiperiod consumption and investment behavior with convex transactions costs, *Management Science* 25 (11), 1127–1137.

Constantinides, George M., 1982, Intertemporal asset pricing with heterogeneous consumers and without demand aggregation, *Journal of Business* 253–267.

Costa, Giorgio, and Garud N. Iyengar, 2023, Distributionally robust end-to-end portfolio construction, *Quantitative Finance* 23 (10), 1465–1482.

DeMiguel, Victor, Lorenzo Garlappi, Francisco J. Nogales, and Raman Uppal, 2009, A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms, *Management Science* 55 (5), 798–812.

DeMiguel, Victor, Alberto Martin-Utrera, Francisco J. Nogales, and Raman Uppal, 2020, A transaction-cost perspective on the multitude of firm characteristics, *Review of Financial Studies* 33 (5), 2180–2222.

Dempe, Stephan, 2002, *Foundations of Bilevel Programming* (Springer Science & Business Media).

Deng, Yue, Feng Bao, Youyong Kong, Zhiquan Ren Ren, and Qionghai Dai, 2016, Deep direct reinforcement learning for financial signal representation and trading, *IEEE Transactions on Neural Networks and Learning Systems* 28 (3), 653–664.

Detzel, Andrew, Robert Novy-Marx, and Mihail Velikov, 2023, Model comparison with transaction costs, *Journal of Finance* 78 (3), 1743–1775.

Didisheim, Antoine, Shikun Ke, Bryan Kelly, and Semyon Malamud, 2024, APT or "AIPT"? the surprising dominance of large factor models, *Working Paper, NBER*.

Dimopoulos, Yannis, Paul Bourret, and Sovan Lek, 1995, Use of some sensitivity criteria for choosing networks with good generalization ability, *Neural Processing Letters* 2 (6), 1–4.

Dominguez, Alejandro Rodriguez, Muhammad Shahzad, and Xia Hong, 2025, Multi-hypothesis prediction for portfolio optimization: A structured ensemble learning approach to risk diversification, *Expert Systems with Applications* 292, 128633.

Dong, Xi, Yan Li, David E. Rapach, and Guofu Zhou, 2022, Anomalies and the expected market return, *Journal of Finance* 77 (1), 639–681.

Donti, Priya, Brandon Amos, and J. Zico Kolter, 2017, Task-based end-to-end model learning in stochastic optimization, *Advances in Neural Information Processing Systems* 30.

Dou, Winston Wei, Itay Goldstein, and Yan Ji, 2025, AI-powered trading, algorithmic collusion, and price efficiency, *Jacobs Levy Equity Management Center for Quantitative Financial Research Paper*.

Drobetz, Wolfgang, and Tizian Otto, 2021, Empirical asset pricing via machine learning: evidence from the European stock market, *Journal of Asset Management* 22 (7), 507–538.

Du, Jiayi, Muyang Jin, Petter N. Kolm, Gordon Ritter, Yixuan Wang, and Bofei Zhang, 2020, Deep reinforcement learning for option replication and hedging, *Journal of Financial Data Science* 2 (4), 44–57.

Duchi, John, Elad Hazan, and Yoram Singer, 2011, Adaptive subgradient methods for online learning and stochastic optimization., *Journal of Machine Learning Research* 12 (7).

Elmachtoub, Adam N., and Paul Grigas, 2022, Smart "predict, then optimize", *Management Science* 68 (1), 9–26.

Feng, Guanhao, Liang Jiang, Junye Li, and Yizhi Song, 2023, Deep tangency portfolio, *Working Paper, City University of Hong Kong*.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber, 2020, Dissecting characteristics nonparametrically, *Review of Financial Studies* 33 (5), 2326–2377.

Frost, Peter A., and James E. Savarino, 1988, For better performance: Constrain portfolio weights, *Journal of Portfolio Management* 15 (1), 29.

Geng, Zhengyang, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin, 2021, On training implicit models, *Advances in Neural Information Processing Systems* 34, 24247–24260.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio, 2016, *Deep Learning* (MIT Press).

Goulart, Paul J, and Yuwen Chen, 2024, Clarabel: An interior-point solver for conic programs with quadratic objectives, *arXiv preprint arXiv:2405.12762*.

Gu, Shihao, Bryan Kelly, and Dacheng Xiu, 2020, Empirical asset pricing via machine learning, *Review of Financial Studies* 33 (5), 2223–2273.

Gu, Shihao, Bryan T. Kelly, and Dacheng Xiu, 2021, Autoencoder asset pricing models, *Journal of Econometrics* 222 (1), 429–450.

Hansen, Lars Peter, and Ravi Jagannathan, 1991, Implications of security market data for models of dynamic economies, *Journal of Political Economy* 99 (2), 225–262.

Hansen, Lars Peter, and Scott F. Richard, 1987, The role of conditioning information in deducing testable restrictions implied by dynamic asset pricing models, *Econometrica* 55, 587–613.

Harvey, Campbell R., Yan Liu, and Heqing Zhu, 2016, ... and the cross-section of expected returns, *Review of Financial Studies* 29 (1), 5–68.

Heaton, James B., Nick G. Polson Polson, and Jan Hendrik Witte, 2017, Deep learning for finance: Deep portfolios, *Applied Stochastic Models in Business and Industry* 33 (1), 3–12.

Hinton, Geoffrey, Nitish Srivastava, and Kevin Swersky, 2012a, Neural networks for machine learning: Lecture 6a overview of mini-batch gradient descent, *Coursera Lecture*.

Hinton, Geoffrey E., Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov, 2012b, Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580*.

Hjalmarsson, Erik, and Petar Manchev, 2012, Characteristic-based mean-variance portfolio choice, *Journal of Banking and Finance* 36 (5), 1392–1401.

Hoffmann, Laurence D., Gerald L. Bradley, and Kenneth H. Rosen, 1989, *Calculus for Business, Economics, and the Social and Life Sciences* (McGraw-Hill).

Hou, Kewei, Chen Xue, and Lu Zhang, 2015, Digesting anomalies: An investment approach, *Review of Financial Studies* 28 (3), 650–705.

Huang, Michael, and Vishal Gupta, 2024, Decision-focused learning with directional gradients, *Advances in Neural Information Processing Systems* 37, 79194–79220.

Huang, Zhichun, Shaojie Bai, and J Zico Kolter, 2021, (Implicit)^2: Implicit layers for implicit representations, *Advances in Neural Information Processing Systems* 34, 9639–9650.

Ioffe, Sergey, and Christian Szegedy, 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *International Conference on Machine Learning* 448–456.

Jensen, Theis Ingerslev, Bryan T. Kelly, Semyon Malamud, and Lasse Heje Pedersen, 2024, Machine learning and the implementable efficient frontier, *Swiss Finance Institute Research Paper*.

Jiang, Zhengyao, Dixing Xu, and Jinjun Liang, 2017, A deep reinforcement learning framework for the financial portfolio management problem, *arXiv preprint arXiv:1706.10059*.

Jorion, Philippe, 1986, Bayes-Stein estimation for portfolio analysis, *Journal of Financial and Quantitative Analysis* 21 (3), 279–292.

Jorion, Philippe, 1992, Portfolio optimization in practice, *Financial Analysts Journal* 48 (1), 68–74.

Kaniel, Ron, Zihan Lin, Markus Pelger, and Stijn Van Nieuwerburgh, 2023, Machine-learning the skill of mutual fund managers, *Journal of Financial Economics* 150 (1), 94–138.

Karush, William, 1939, Minima of functions of several variables with inequalities as side constraints, *Master's Dissertation, University of Chicago*.

Kelly, Bryan T., Diogo Palhares, and Seth Pruitt, 2023, Modeling corporate bond returns, *Journal of Finance* 78 (4), 1967–2008.

Kelly, Bryan T., Seth Pruitt, and Yinan Su, 2020, Instrumented principal component analysis, *Working Paper, Yale University*.

Kelly, Bryan T., and Dacheng Xiu, 2023, Financial machine learning, *Foundations and Trends in Finance* 13, 205–363.

Kingma, Diederik P., and Jimmy Ba, 2014, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

Kotary, James, Ferdinando Fioretto, Pascal Van Hentenryck, and Bryan Wilder, 2021, End-to-end constrained optimization learning: A survey, *arXiv preprint arXiv:2103.16378*.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh, 2020, Shrinking the cross-section, *Journal of Financial Economics* 135 (2), 271–292.

Kuhn, Harold W., and Albert W. Tucker, 2013, Nonlinear programming, in *Traces and Emergence of Nonlinear Programming*, 247–258 (Springer).

Le, Ngan, Vidhiwar Singh Rathour, Kashu Yamazaki, Khoa Luu, and Marios Savvides, 2022, Deep reinforcement learning in computer vision: A comprehensive survey, *Artificial Intelligence Review* 55 (4), 2733–2819.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton, 2015, Deep learning, *Nature* 521 (7553), 436–444.

Lee, Junhyeong, Haeun Jeon, Hyunglip Bae, and Yongjae Lee, 2025, Return prediction for mean-variance portfolio selection: How decision-focused learning shapes forecasting models.

Leippold, Markus, Qian Wang, and Wenyu Zhou, 2022, Machine learning in the Chinese stock market, *Journal of Financial Economics* 145 (2), 64–82.

Lettau, Martin, and Markus Pelger, 2020, Estimating latent asset-pricing factors, *Journal of Econometrics* 218 (1), 1–31.

Li, Bin, and Alberto G. Rossi, 2020, Selecting mutual funds from the stocks they hold: A machine learning approach, *Working Paper, Georgetown University*.

Liu, Jianan, Robert F. Stambaugh, and Yu Yuan, 2019, Size and value in China, *Journal of Financial Economics* 134 (1), 48–69.

Look, Andreas, Simona Doneva, Melih Kandemir, Rainer Gemulla, and Jan Peters, 2020, Differentiable implicit layers, *arXiv preprint arXiv:2010.07078*.

Luenberger, David G., 1997, *Optimization by Vector Space Methods* (John Wiley & Sons).

Magill, Michael J.P., and George M. Constantinides, 1976, Portfolio selection with transactions costs, *Journal of Economic Theory* 13 (2), 245–263.

Mandi, Jayanta, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto, 2024, Decision-focused learning: Foundations, state of the art, benchmark and future opportunities, *Journal of Artificial Intelligence Research* 80, 1623–1701.

Markowitz, Harry M., 1952, Portfolio selection, *Journal of Finance* 7 (1), 71–91.

Martin, Ian W. R., and Stefan Nagel, 2022, Market efficiency in the age of big data, *Journal of Financial Economics* 145 (1), 154–177.

Masters, Timothy, 1993, *Practical Neural Network Recipes in C++* (Morgan Kaufmann).

Muravyev, Dmitriy, Neil D. Pearson, and Joshua Matthew Pollet, 2025, Anomalies and their short-sale costs, *Working Paper, University of Illinois at Urbana-Champaign.*

Newey, Whitney K., and Kenneth D. West, 1987, A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix, *Econometrica* 55 (3), 703–708.

Novy-Marx, Robert, and Mihail Velikov, 2016, A taxonomy of anomalies and their trading costs, *Review of Financial Studies* 29 (1), 104–147.

Ostertagová, Eva, 2012, Modelling using polynomial regression, *Procedia Engineering* 48, 500–506.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, 2019, Pytorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems* 32, 8026–8037.

Patton, Andrew J., and Brian M. Weller, 2020, What you see is not what you get: The costs of trading market anomalies, *Journal of Financial Economics* 137 (2), 515–549.

Pogančić, Marin Vlastelica, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek, 2019, Differentiation of blackbox combinatorial solvers, *International Conference on Learning Representations* .

Prechelt, Lutz, 2002, Early stopping-but when?, in *Neural Networks: Tricks of the Trade*, 55–69 (Springer).

Pricope, Tidor-Vlad, 2021, Deep reinforcement learning in quantitative algorithmic trading: A review, *arXiv preprint arXiv:2106.00123*.

Sadana, Utsav, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal, 2025, A survey of contextual optimization methods for decision-making under uncertainty, *European Journal of Operational Research* 320 (2), 271–289.

Sharma, Akanksha Rai, and Pranav Kaushik, 2017, Literature survey of statistical, deep and reinforcement learning in natural language processing, *International Conference on Computing, Communication and Automation (ICCCA)* 350–354.

Silver, David, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis, 2016, Mastering the game of go with deep neural networks and tree search, *Nature* 529 (7587), 484–489.

Simon, Frederik, Sebastian Weibels, and Tom Zimmermann, 2025, Deep parametric portfolio policies, *Working Paper, Centre for Financial Research (CFR)*.

Sinha, Ankur, Pekka Malo, and Kalyanmoy Deb, 2017, A review on bilevel optimization: From classical to evolutionary approaches and applications, *IEEE Transactions on Evolutionary Computation* 22 (2), 276–295.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, 2014, Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research* 15 (1), 1929–1958.

Stambaugh, Robert F., Jianfeng Yu, and Yu Yuan, 2012, The short of it: Investor sentiment and anomalies, *Journal of Financial Economics* 104 (2), 288–302.

Sutton, Richard S., David McAllester, Satinder Singh, and Yishay Mansour, 1999, Policy gradient methods for reinforcement learning with function approximation, *Advances in Neural Information Processing Systems* 12.

Tang, Bo, and Elias B. Khalil, 2024, PyEPO: a PyTorch-based end-to-end predict-then-optimize library for linear and integer programming, *Mathematical Programming Computation* 16 (3), 297–335.

Uddin, Ajim, Xinyuan Tao, and Dantong Yu, 2024, The network factor of equity pricing: A signed graph laplacian approach, *Journal of Financial Econometrics* 22 (5), 1616–1655.

Von Stackelberg, Heinrich, 1934, *Marktform und Gleichgewicht* (Springer).

Von Stackelberg, Heinrich, 2010, *Market Structure and Equilibrium* (Springer Science & Business Media).

Wang, Rong, 2025, Factor investing and factor-neutral investing, *Working Paper, Duke University*.

Williams, A.C., 1970, Complementarity theorems for linear programming, *SIAM Review* 12 (1), 135–137.

Wu, Wenbo, Jiaqi Chen, Zhibin Yang, and Michael L. Tindall, 2021, A cross-sectional machine learning approach for hedge fund return prediction and selection, *Management Science* 67 (7), 4577–4601.

Yao, Yuan, Lorenzo Rosasco, and Andrea Caponnetto, 2007, On early stopping in gradient descent learning, *Constructive Approximation* 26 (2), 289–315.

Yu, Pengqian, Joon Sern Lee, Ilya Kulyatin, Zekun Shi, and Sakyasingha Dasgupta, 2019, Model-based deep reinforcement learning for dynamic portfolio optimization, *arXiv preprint arXiv:1901.08740*.

Zhao, Wenshuai, Jorge Peña Queralta, and Tomi Westerlund, 2020, Sim-to-real transfer in deep reinforcement learning for robotics: A survey, *2020 IEEE Symposium Series on Computational Intelligence* 737–744.

Zhou, Yang, Jianqing Fan, and Lirong Xue, 2024, How much can machines learn finance from Chinese text data?, *Management Science* 70 (12), 8962–8987.

Online Appendix to
"Machine Learning Meets Markowitz"

A.1 Details of Data

A.1.1 Data sample

Our final sample spans the period 2010 to 2023, consisting of 5,489 unique stocks over 14 years. The average number of stocks in each year is reported in Table A.1. The steady increase, from 1,841 stocks in 2010 to 5,196 in 2023, is mainly driven by the expansion of the Chinese A-share market during this period. As our dataset construction aims for broad coverage, encompassing the vast majority of listed A-share firms, its size expands in line with the market's growth.[24]

A.1.2 Predictors

In our analysis, we incorporate a comprehensive set of 415 real-time firm-level characteristics, which are widely documented in the asset pricing literature as predictors for stock returns. These characteristics span a broad set of firm attributes, including valuation ratios, momentum, risk measures, and profitability metrics. Detailed information on each characteristic, including its definition and the original study, is provided in Table A.2.

Table A.2: Details on stock characteristics

Acronym	**Original Study**	**Characteristic Description**
Abnormalaccruals	Xie (2001)	Abnormal Accruals
Abnormalaccrualspercent	Hafzalla, Lundholm, and van Winkle (2011)	Percent Abnormal Accruals
Accrualquality	Francis et al. (2005)	Accrual Quality
Accrualqualityjune	Francis et al. (2005)	Accrual Quality in June
Accruals	Sloan (1996)	Accruals
Accruals_3y	Sloan (1996)	Accruals 3 Year Mean
Accruals_5y	Sloan (1996)	Accruals 5 Year Mean
Accrualsbm	Bartov and Kim (2004)	Book-to-market and Accruals
Activism2	Cremers and Nair (2005)	Active Shareholders
Activisml	Cremers and Nair (2005)	Takeover Vulnerability

Continued on the next page

[24]To ensure the investability and tradability of our asset universe, we exclude stocks that are under trading suspension or designated as "Special Treatment" (ST). In the Chinese A-share market, the "ST" designation is applied to firms facing financial abnormalities, such as consecutive years of net losses, which signals a risk of delisting. Due to strict risk management mandates and regulatory constraints, many institutional investors are prohibited from holding these high-risk assets.

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Adexp	Chan, Lakonishokand, and Sougiannis (2001)	Advertising Expense
Ageipo	Ritter (1991)	Ipo and Age
Am	Fama and French (1992)	Total Assets to Market
Am_3y	Fama and French (1992)	Total Assets to Market 3 Year Mean
Am_5y	Fama and French (1992)	Total Assets to Market 5 Year Mean
Amq	Fama and French (1992)	Total Assets to Market(quarterly)
Analystrevision	Hawkins, Chamberlin, and Daniel (1984)	Eps Forecast Revision
Analystvalue	Frankel and Lee (1998)	Analyst Value
Announcementret	Chan, Jegadeesh, and Lakonishok (1996)	Earnings Announcement Return
Aop	Frankel and Lee (1998)	Analyst Optimism
Assetgrowth	Cooper, Gulen, and Schill (2008)	Asset Growth
Assetgrowth_q	Cooper, Gulen, and Schill (2008)	Asset Growth Quarterly
Assetliquiditybook	Ortiz-Molina and Phillips (2014)	Asset Liquidity Over Book Assets
Assetliquiditybookquart	Ortiz-Molina and Phillips (2014)	Asset Liquidity Over Book(qtrly)
Assetliquiditymarket	Ortiz-Molina and Phillips (2014)	Asset Liquidity Over Market
Assetliquiditymarketquart	Ortiz-Molina and Phillips (2014)	Asset Liquidity Over Market (qtrly)
Assetturnover	Soliman (2008)	Asset Turnover
Assetturnover_q	Soliman (2008)	Asset Turnover Quarterly
Beta	Fama and Macbeth (1973)	Capm Beta
Betabdleverage	Adrian, Etula, and Muir (2014)	Broker-dealer Leverage Beta
Betacc	Acharya and Pedersen (2005)	Iliquidity-illiquidity Beta(beta2i)
Betacr	Acharya and Pedersen (2005)	Iliquidity-market Return Beta(beta4i)
Betadimson	Dimson (1979)	Dimson Beta
Betafp	Frazzini and Pedersen (2014)	Frazzini-pedersen Beta
Betaliquidityps	Pastor and Stambaugh (2003)	Pastor-stambaugh Liquidity Beta
Betanet	Acharya and Pedersen (2005)	Net Liquidity Beta(betanet,p)
Betarc	Acharya and Pedersen (2005)	Return-market Illiquidity Beta
Betarr	Acharya and Pedersen (2005)	Return-market Returnilliquidity Beta
Betasquared	Fama and Macbeth (1973)	Capm Beta Squred
Betatailrisk	Kelly and Jiang (2014)	Tail Risk Beta
Betavix	Ang et al. (2006)	Systematic Volatility
Bidaskspread	Amihud and Mendelsohn (1986)	Bid-ask Spread
Bidasktaq	Hou and Loh (2016)	Bid-ask Spread(taq)
Bm	Rosenberg, Reid, and Lanstein (1985)	Book to Market Using Most Recent Me
Bmdec	Fama and French (1992)	Book to Market Using December Me
Bmq	Rosenberg, Reid, and Lanstein (1985)	Book to Market (quarterly)
Bookleverage	Fama and French (1992)	Bookleverage(annual)
Bookleveragequarterly	Fama and French (1992)	Bookleverage (quarterly)
Bpebm	Penman, Richardson, and Tuna (2007)	Leverage Component Of Bm
Brandcapital	Belo, Lin, and Vitorino (2014)	Brand Capital to Assets
Brandinvest	Belo, Lin, and Vitorino (2014)	Brand Capital Investment
Capturnover	Haugen and Baker (1996)	Capital Turnover
Capturnover_q	Haugen and Baker (1996)	Capital Turnover(quarterly)
Cash	Palazzo (2012)	Cash to Assets
Cash_3y	Palazzo (2012)	Cash to Assets 3 Year Mean

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Cash_5y	Palazzo (2012)	Cash to Assets 5 Year Mean
Cashdebt	Ou and Penman (1989)	Cf to Debt
Cashdebt_q	Ou and Penman (1989)	Cf to Debt
Cashprod	Chandrashekar and Rao (2009)	Cash Productivity
Cboperprof	Ball et al. (2016)	Cash-based Operating Profitability
Cboperproflagat	Ball et al. (2016)	Cash-based Oper Prof Lagged Assets
Cboperproflagat_q	Ball et al. (2016)	Cash-based Oper Prof Lagged Assets Qtrly
Cf	Lakonishok, Shleifer, and Vishny (1994)	Cash Flow to Market
Cfp	Desai, Rajgopal, and Venkatachalam (2004)	Operating Cash Flows to Price
Cfpq	Desai, Rajgopal, and Venkatachalam (2004)	Operating Cash Flows to Price Quarterly
Cfq	Lakonishok, Shleifer, and Vishny (1994)	Cash Flow to Market Quarterly
Changeinrecommendation	Jegadeesh et al. (2004)	Change in Recommendation
Changeinrecommendation_3y	Jegadeesh et al. (2004)	Change in Recommendation
Changeinrecommendation_5y	Jegadeesh et al. (2004)	Change in Recommendation
Changeroa	Balakrishnan, Bartov, and Faurel (2010)	Change in Return on Assets
Changeroa_3y	Balakrishnan, Bartov, and Faurel (2010)	Change in Return on Assets
Changeroa_5y	Balakrishnan, Bartov, and Faurel (2010)	Change in Return on Assets
Changeroe	Balakrishnan, Bartov, and Faurel (2010)	Change in Return on Equity
Changeroe_3y	Balakrishnan, Bartov, and Faurel (2010)	Change in Return on Equity
Changeroe_5y	Balakrishnan, Bartov, and Faurel (2010)	Change in Return on Equity
Chassetturnover	Soliman (2008)	Change in Asset Turnover
Cheq	Lockwood and Prombutr (2010)	Growth in Book Equity
Chforecastaccrual	Barth and Hutton (2004)	Change in Forecast and Accrual
Chinv	Thomas and Zhang (2002)	Inventory Growth
Chinvia	Abarbanell and Bushee (1998)	Change in Capitalinv(ind Adj)
Chnanalyst	Scherbina (2008)	Decline in Analyst Coverage
Chncoa	Soliman (2008)	Change in Noncurrent Operating Assets
Chncoa_3y	Soliman (2008)	Change in Noncurrent Operating Assets
Chncoa_5y	Soliman (2008)	Change in Noncurrent Operating Assets
Chncol	Soliman (2008)	Change in Noncurrent Operating Liab
Chncol_3y	Soliman (2008)	Change in Noncurrent Operating Liab
Chncol_5y	Soliman (2008)	Change in Noncurrent Operating Liab
Chnncoa	Soliman (2008)	Change in Net Noncurrent Op Assets
Chnncoa_3y	Soliman (2008)	Change in Net Noncurrent Op Assets
Chnncoa_5y	Soliman (2008)	Change in Net Noncurrent Op Assets
Chnwc	Soliman (2008)	Change in Net Working Capital
Chnwc_3y	Soliman (2008)	Change in Net Working Capital
Chnwc_5y	Soliman (2008)	Change in Net Working Capital
Chpm	Soliman (2008)	Change in Profit Margin
Chpm_3y	Soliman (2008)	Change in Profit Margin
Chpm_5y	Soliman (2008)	Change in Profit Margin
Chtax	Thomas and Zhang (2011)	Change in Taxes
Chtax_3y	Thomas and Zhang (2011)	Change in Taxes
Chtax_5y	Thomas and Zhang (2011)	Change in Taxes
Citationsrd	Hirshleifer, Hsu, and Li (2013)	Citations to Rd Expenses

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Compequlss	Daniel and Titman (2006)	Composite Equity Issuance
Compositedebtissuance	Lyandres, Sun, and Zhang (2008)	Composite Debt Issuance
Consrecomm	Barber et al. (2002)	Consensus Recommendation
Convdebt	Valta (2016)	Convertible Debt Indicator
Coskewacx	Ang, Chen, and Xing (2006)	Coskewness Using Daily Returns
Coskewness	Harvey and Siddique (2000)	Coskewness
Credratdg	Dichev and Piotroski (2001)	Credit Rating Downgrade
Currat	Ou and Penman (1989)	Current Ratio
Customermom	Cohen and Frazzini (2008)	Customer Momentum
Debtissuance	Spiess and Affleck-Graves (1999)	Debt Issuance
Delayacct	Callen, Khan, and Lu (2013)	Accounting Component Of Price Delay
Delaynonacct	Callen, Khan, and Lu (2013)	Non-accounting Component Of Price Delay
Delbreadth	Chen, Hong, and Stein (2002)	Breadth Of Ownership
Delcoa	Richardson et al. (2005)	Change in Current Operating Assets
Delcoa_3y	Richardson et al. (2005)	Change in Current Operating Assets
Delcoa_5y	Richardson et al. (2005)	Change in Current Operating Assets
Delcol	Richardson et al. (2005)	Change in Current Operating Liabilities
Delcol_3y	Richardson et al. (2005)	Change in Current Operating Liabilities
Delcol_5y	Richardson et al. (2005)	Change in Current Operating Liabilities
Deldrc	Prakash and Sinha (2012)	Deferred Revenue
Deldrc_3y	Prakash and Sinha (2012)	Deferred Revenue
Deldrc_5y	Prakash and Sinha (2012)	Deferred Revenue
Delequ	Richardson et al. (2005)	Change in Equity to Assets
Delequ_3y	Richardson et al. (2005)	Change in Equity to Assets
Delequ_5y	Richardson et al. (2005)	Change in Equity to Assets
Delfinl	Richardson et al. (2005)	Change in Financial Liabilities
Delfinl_3y	Richardson et al. (2005)	Change in Financial Liabilities
Delfinl_5y	Richardson et al. (2005)	Change in Financial Liabilities
Dellti	Richardson et al. (2005)	Change in Long-term Investment
Dellti_3y	Richardson et al. (2005)	Change in Long-term Investment
Dellti_5y	Richardson et al. (2005)	Change in Long-term Investment
Delnetfin	Richardson et al. (2005)	Change in Net Financial Assets
Delnetfin_3y	Richardson et al. (2005)	Change in Net Financial Assets
Delnetfin_5y	Richardson et al. (2005)	Change in Net Financial Assets
Delsti	Richardson et al. (2005)	Change in Short-term Investment
Delsti_3y	Richardson et al. (2005)	Change in Short-term Investment
Delsti_5y	Richardson et al. (2005)	Change in Short-term Investment
Depr	Holthausen and Larcker (1992)	Depreciation to Ppe
Depr_q	Holthausen and Larcker (1992)	Depreciation to Ppe
Divinit	Michaely, Thaler, and Womack (1995)	Dividend Initiation
Divomit	Michaely, Thaler, and Womack (1995)	Dividend Omission
Divseason	Hartzmark and Salomon (2013)	Dividend Seasonality
Divyield	Naranjo, Nimalendran, and Ryngaert (1998)	Dividend Yield For Small Stocks
Divyieldann	Naranjo, Nimalendran, and Ryngaert (1998)	Last Year's Dividends Over Price
Divyieldst	Litzenberger and Ramaswamy (1979)	Predicted Divyield Next Month

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Dnoa	Hirshleifer et al. (2004)	Change in Net Operating Assets
Dnoa_3y	Hirshleifer et al. (2004)	Change in Net Operating Assets
Dnoa_5y	Hirshleifer et al. (2004)	Change in Net Operating Assets
Dolvol	Brennan, chordia, and Subra (1998)	Past Trading Volume
Downrecomm	Barber et al. (2002)	Down Forecast Eps
Downsidebeta	Ang, Chen, and Xing (2006)	Downside Beta
Earningsconservatism	Francis et al. (2004)	Earnings Conservatism
Earningsconsistency	Alwathainani (2009)	Earnings Consistency
Earningsforecastdisparity	Da and Warachka (2011)	Long-vs-short Eps Forecasts
Earningspersistence	Francis et al. (2004)	Earnings Persistence
Earningspredictability	Francis et al. (2004)	Earnings Predictability
Earningssmoothness	Francis et al. (2004)	Earnings Smoothness
Earningsstreak	Loh and Warachka (2012)	Earnings Surprise Streak
Earningssurprise	Foster, Olsen, and Shevlin (1984)	Earnings Surprise
Earningstimeliness	Francis et al. (2004)	Earnings Timeliness
Earningsvaluerelevance	Francis et al. (2004)	Value Relevance Of Earnings
Earnsupbig	Hou (2007)	Earnings Surprise Of Big Firms
Ebm	Penman, Richardson, and Tuna (2007)	Enterprise Component Of Bm
Ebm_q	Penman, Richardson, and Tuna (2007)	Enterprise Component Of Bm
Entmult	Loughran and Wellman (2011)	Enterprise Multiple
Entmult_q	Loughran and Wellman (2011)	Enterprise Multiple Quarterly
Ep	Basu (1977)	Earnings-to-price Ratio
Epq	Basu (1977)	Earnings-to-price Ratio
Equityduration	Dechow, Sloan, and Soliman (2004)	Equity Duration
Etr	Abarbanell and Bushee (1998)	Effective Tax Rate
Etr_3y	Abarbanell and Bushee (1998)	Effective Tax Rate 3 Year
Etr_5y	Abarbanell and Bushee (1998)	Effective Tax Rate 5 Year
Exchswitch	Dharan and Ikenberry (1995)	Exchange Switch
Exciexp	Doyle, Lundholm, and Soliman (2003)	Excluded Expenses
Failureprobability	Campbell, Hilscher, and Szilagyi (2008)	Failure Probability
Failureprobabilityjune	Campbell, Hilscher, and Szilagyi (2008)	Failure Probability
Feps	Cen, Wei, and Zhang (2006)	Analyst Earnings Per Share
Fgr5yrlag	La Porta (1996)	Long-term Eps Forecast
Fgr5yrnolag	La Porta (1996)	Long-term Eps Forecast(monthly)
Firmage	Barry and Brown (1984)	Firm Age Based on Crsp
Firmagemom	Zhang (2004)	Firm Age-momentum
Forecastdispersion	Diether, Malloy, and Scherbina (2002)	Eps Forecast Dispersion
Forecastdispersionlt	Anderson, Ghysels, and Juergens (2005)	Long-term Forecast Dispersion
Fr	Franzoni and Marin (2006)	Pension Funding Status
Frbook	Franzoni and Marin (2006)	Pension Funding Status
Frontier	Nguyen and Swanson (2009)	Efficient Frontier Index
Governance	Gompers, Ishii, and Metrick (2003)	Governance Index
Gp	Novy-Marx (2013)	Gross Profits /total Assets
Gp_q	Novy-Marx (2013)	Gross Profits /total Assets
Gplag	Novy-Marx (2013)	Gross Profits /total Assets

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Gplag_q	Novy-Marx (2013)	Gross Profits /total Assets
Gradexp	Lou (2014)	Growth in Advertising Expenses
Grcapx	Anderson and Garcia-Feijoo (2006)	Change in Capex(2 Years)
Grcapx3y	Anderson and Garcia-Feijoo (2006)	Change in Capex(3 Years)
Grcapx5y	Anderson and Garcia-Feijoo (2006)	Change in Capex(5 Years)
Grinvest1y	Anderson and Garcia-Feijoo (2006)	Investment Growth(1 Years)
Grinvest3y	Anderson and Garcia-Feijoo (2006)	Investment Growth(3 Years)
Grinvest5y	Anderson and Garcia-Feijoo (2006)	Investment Growth(5 Years)
Grgmtogrsales	Abarbanell and Bushee (1998)	Gross Margin Growth to Sales Growth
Grgmtogrsales_3y	Abarbanell and Bushee (1998)	Gross Margin Growth to Sales Growth
Grgmtogrsales_5y	Abarbanell and Bushee (1998)	Gross Margin Growth to Sales Growth
Grltnoa	Fairfield, Whisenant, and Yohn (2003)	Growth in Long Term Operating Assets
Grltnoa_3y	Fairfield, Whisenant, and Yohn (2003)	Growth in Long Term Operating Assets
Grltnoa_5y	Fairfield, Whisenant, and Yohn (2003)	Growth in Long Term Operating Assets
Grsaletogrinv	Abarbanell and Bushee (1998)	Sales Growth Over Inventory Growth
Grsaletogrinv_3y	Abarbanell and Bushee (1998)	Sales Growth Over Inventory Growth
Grsaletogrinv_5y	Abarbanell and Bushee (1998)	Sales Growth Over Inventory Growth
Grsaletogroverhead	Abarbanell and Bushee (1998)	Sales Growth Over Overhead Growth
Grsaletogroverhead_3y	Abarbanell and Bushee (1998)	Sales Growth Over Overhead Growth
Grsaletogroverhead_5y	Abarbanell and Bushee (1998)	Sales Growth Over Overhead Growth
Grsaletogrreceivables	Abarbanell and Bushee (1998)	Change in Sales Vs Change in Receiv
Grsaletogrreceivables_3y	Abarbanell and Bushee (1998)	Change in Sales Vs Change in Receiv
Grsaletogrreceivables_5y	Abarbanell and Bushee (1998)	Change in Sales Vs Change in Receiv
Herf	Hou and Robinson (2006)	Industry Concentration(sales)
Herfasset	Hou and Robinson (2006)	Industry Concentration(assets)
Herfbe	Hou and Robinson (2006)	Industry Concentration(equity)
High52	George and Hwang (2004)	52 Week High
Hire	Bazdresch, Belo, and Lin (2014)	Employment Growth
Idiorisk	Ang et al. (2006)	Idiosyncratic Risk
Idiovol3f	Ang et al. (2006)	Idiosyncratic Risk (3 Factor)
Idiovolaht	Ali, Hwang, and Trombley (2003)	Idiosyncratic Risk (aht)
Idiovolcapm	Ang et al. (2006)	Idiosyncratic Risk(capm)
Idiovolqf	Ang et al. (2006)	Idiosyncratic Risk(q Factor)
Illiquidity	Amihud (2002)	Amihud's Illiquidity
Indipo	Ritter (1991)	Initial Public Offerings
Indmom	Grinblatt and Moskowitz (1999)	Industry Momentum
Indretbig	Hou (2007)	Industry Return Ofbig Firms
Intanbm	Daniel and Titman (2006)	Intangible Return Using Bm
Intancfp	Daniel and Titman (2006)	Intangible Return Using Cftop
Intanep	Daniel and Titman (2006)	Intangible Return Using Ep
Intansp	Daniel and Titman (2006)	Intangible Return Using Sale2p
Intmom	Novy-Marx (2012)	Intermediate Momentum
Intrinsicvalue	Frankel and Lee (1998)	Intrinsic Or Historical Value
Investment	Titman, Wei, and Xie (2004)	Investment to Revenue
Investppeinv	Lyandres, Sun, and Zhang (2008)	Change in Ppe and Inv/assets

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Invgrowth	Belo and Lin (2012)	Inventory Growth
Io_shortinterest	Asquith, Pathak, and Ritter (2005)	Inst Own Among High Short Interest
Iomom_cust	Menzly and Ozbas (2010)	Customers Momentum
Iomom_supp	Menzly and Ozbas (2010)	Suppliers Momentum
Kz	Lamont, Polk, and Saa-Requejo (2001)	Kaplan Zingales Index
Kz_q	Lamont, Polk, and Saa-Requejo (2001)	Kaplan Zingales Index Quarterly
Laborforceefficiency	Abarbanell and Bushee (1998)	Laborforce Efficiency
Leverage	Bhandari (1988)	Market Leverage
Leverage_q	Bhandari (1988)	Market Leverage Quarterly
Lrreversal	De Bondt and Thaler (1985)	Long-run Reversal
Maxret	Bali, Cakici, and Whitelaw (2010)	Maximum Return Over Month
Maxret_2m	Bali, Cakici, and Whitelaw (2010)	Maximum Return Over 2 Months
Maxret_3m	Bali, Cakici, and Whitelaw (2010)	Maximum Return Over 3 Months
Maxret_6m	Bali, Cakici, and Whitelaw (2010)	Maximum Return Over 6 Months
Maxret_9m	Bali, Cakici, and Whitelaw (2010)	Maximum Return Over 9 Months
Maxret_12m	Bali, Cakici, and Whitelaw (2010)	Maximum Return Over 12 Months
Meanrankrevgrowth	Lakonishok, Shleifer, and Vishny (1994)	Revenue Growth Rank
Mom1m	Jegadeesh and Titman (1993)	Momentum(1month)
Mom2m	Jegadeesh and Titman (1993)	Momentum(2month)
Mom3m	Jegadeesh and Titman (1993)	Momentum(3month)
Mom6m	Jegadeesh and Titman (1993)	Momentum(6month)
Mom12m	Jegadeesh and Titman (1993)	Momentum(12 Month)
Mom12moffseason	Heston and Sadka (2008)	Momentum Without The Seasonalpart
Mom1mjunk	Avramovet et al. (2007)	Junk Stock Momentum
Mom2mjunk	Avramovet et al. (2007)	Junk Stock Momentum
Mom3mjunk	Avramovet et al. (2007)	Junk Stock Momentum
Mom6mjunk	Avramovet et al. (2007)	Junk Stock Momentum
Mom12mjunk	Avramovet et al. (2007)	Junk Stock Momentum
Momoffseason	Heston and Sadka (2008)	Off Season Long-term Reversal
Momoffseason06yrplus	Heston and Sadka (2008)	Off Season Reversal Years6 to 10
Momoffseason16yrplus	Heston and Sadka (2008)	Off Season Reversal Years 16 to 20
Momoffseasonllyrplus	Heston and Sadka (2008)	Off Season Reversal Years 1l to 15
Momrev	Chan and Ko (2006)	Momentum and Lt Reversal
Momseason	Heston and Sadka (2008)	Return Seasonality Years 2 to 5
Momseason06yrplus	Heston and Sadka (2008)	Return Seasonality Years 6to 10
Momseason16yrplus	Heston and Sadka (2008)	Return Seasonality Years 16 to 20
Momseasonl1yrplus	Heston and Sadka (2008)	Return Seasonality Years Ll to 15
Momseasonshort	Heston and Sadka (2008)	Return Seasonality Last Year
Momvol	Lee and Swaminathan (2000)	Momentum in High Volume Stocks
Momvol_1m	Lee and Swaminathan (2000)	Momentum in High Volume Stocks
Momvol_3m	Lee and Swaminathan (2000)	Momentum in High Volume Stocks
Momvol_6m	Lee and Swaminathan (2000)	Momentum in High Volume Stocks
Momvol_9m	Lee and Swaminathan (2000)	Momentum in High Volume Stocks
Momvol_12m	Lee and Swaminathan (2000)	Momentum in High Volume Stocks
Mrreversal	De Bondt and Thaler (1985)	Medium-run Reversal

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Ms	Mohanram (2005)	Mohanram G-score
Nanalyst	Elgers, Lo, and Pfeiffer (2001)	Number Of Analysts
Netdebtfinance	Bradshaw, Richardson, and Sloan (2006)	Net Debt Financing
Netdebtprice	Penman, Richardson, and Tuna (2007)	Net Debt to Price
Netdebtprice_g	Penman, Richardson, and Tuna (2007)	Net Debt to Price
Netequityfinance	Bradshaw, Richardson, and Sloan (2006)	Net Equity Financing
Netpayoutyield	Boudoukh et al. (2007)	Net Payout Yield
Netpayoutyield_q	Boudoukh et al. (2007)	Net Payout Yield Quarterly
Noa	Hirshleifer et al. (2004)	Net Operating Assets
Numearnlncrease	Loh and Warachka (2012)	Earnings Streak Length
Operprof	Fama and French (2006)	Operating Profits /book Equity
Operproflag	Fama and French (2006)	Operating Profits /book Equity
Operproflag_9	Fama and French (2006)	Operating Profits /book Equity
Operprofrd	Ball et al. (2016)	Operating Profitability R&d Adjusted
Operprofrdlagat	Ball et al. (2016)	Oper Prof R&d Adjlagged Assets
Operprofrdlagat_q	Ball et al. (2016)	Oper Prof R&d Adjlagged Assets (qtrly)
Sp_q	Barbee, Mukherji, and Raines (1996)	Sales-to-price Quarterly
Opleverage	Novy-Marx (2010)	Operating Leverage
Opleverage_q	Novy-Marx (2010)	Operating Leverage (qtrly)
Optionvolume2	Johnson and So (2012)	Option Volume to Average
Optionvolumel	Johnson and So (2012)	Option to Stock Volume
Orderbacklog	Rajgopal, Shevlin, and Venkatachalam (2003)	Order Backlog
Orderbacklogchg	Baik and Ahn (2007)	Change in Order Backlog
Orgcap	Eisfeldt and Papanikolaou (2013)	Organizational Capital
Orgcapnoadj	Eisfeldt and Papanikolaou (2013)	Org Cap W/o Industry Adjustment
Oscore	Dichev (1998)	O Score
Oscore_q	Dichev (1998)	O Score Quarterly
Patentsrd	Hirshleifer, Hsu, and Li (2013)	Patents to Rd Expenses
Payoutyield	Boudoukh et al. (2007)	Payout Yield
Payoutyield_q	Boudoukh et al. (2007)	Payout Yield Quarterly
Pchcurrat	Ou and Penman (1989)	Change in Current Ratio
Pchdepr	Holthausen and Larcker (1992)	Change in Depreciation to Ppe
Pchgm_pchsale	Abarbanell and Bushee (1998)	Change in Gross Margin Vs Sales
Pchquick	Ou and Penman (1989)	Change in Quick Ratio
Pchsaleinv	Ou and Penman (1989)	Change in Sales to Inventory
Pctacc	Hafzalla, Lundholm, and van Winkle (2011)	Percent Operating Accruals
Pcttotacc	Hafzalla, Lundholm, and van Winkle (2011)	Percent Totalaccruals
Pm	Soliman (2008)	Profit Margin
Pm_q	Soliman (2008)	Profit Margin
Predictedfe	Frankel and Lee (1998)	Predicted Analyst Forecast Error
Price	Blume and Husic (1972)	Price
Pricedelayrsq	Hou and Moskowitz (2005)	Price Delay R Square
Pricedelayslope	Hou and Moskowitz (2005)	Price Delay Coeff
Pricedelaytstat	Hou and Moskowitz (2005)	Price Delay Se Adjusted
Probinformedtrading	Easley, Hvidkjaer, and O'hara (2002)	Probability Of Informed Trading

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Ps	Piotroski (2000)	Piotroski F-score
Ps_q	Piotroski (2000)	Piotroski F-score
Quick	Ou and Penman (1989)	Quick Ratio
Rd	Chan, Lakonishok, and Sougiannis (2001)	R&d Over Market Cap
Rd_q	Chan, Lakonishok, and Sougiannis (2001)	R&d Over Market Cap Quarterly
Rd_sale	Chan, Lakonishok, and Sougiannis (2001)	R&d to Sales
Rd_sale_q	Chan, Lakonishok, and Sougiannis (2001)	R&d to Sales
Rdability	Cohen, Diether, and Malloy (2013)	R&d Ability
Rdcap	Li (2011)	R&d Capital-to-assets
Rdipo	Gou, Lev, and Shi (2006)	Ipo and No R&d Spending
Rds	Landsman et al. (2011)	Real Dirty Surplus
Realestate	Tuzel (2010)	Real Estate Holdings
Residualmom	Blitz, Huij, and Martens (2011)	Momentum Based on Ff3 Residuals
Residualmom6m	Blitz, Huij, and Martens (2011)	6month Residual Momentum
Retconglomerate	Cohen and Lou (2012)	Conglomerate Return
Retnoa	Soliman (2008)	Return on Net Operating Assets
Retnoa_q	Soliman (2008)	Return on Net Operating Assets
Retskew	Bali, Engle, and Murray (2015)	Return Skewness
Retskew3f	Bali, Engle, and Murray (2015)	Idiosyncratic Skewness(3fmodel)
Retskewcapm	Bali, Engle, and Murray (2015)	Idiosyncratic Skewness(capm)
Retskewqf	Bali, Engle, and Murray (2015)	Idiosyncratic Skewness(qmodel)
Rev6	Chan, Jegadeesh, and Lakonishok (1996)	Earnings Forecast Revisions
Revenuesurprise	Jegadeesh and Livnat (2006)	Revenue Surprise
Rio_disp	Nagel (2005)	Inst Own and Forecast Dispersion
Rio_mb	Nagel (2005)	Inst Own and Market to Book
Rio_turnover	Nagel (2005)	Inst Own and Turnover
Rio_volatility	Nagel (2005)	Inst Own and Idio Vol
Roaq	Balakrishnan, Bartov, and Faurel (2010)	Return on Assets (qtrly)
Roavol	Francis et al. (2004)	Roa Volatility
Roe	Haugen and Baker (1996)	Net Income /bookequity
Roic	Brown and Rowe (2007)	Return on Invested Capital
Salecash	Ou and Penman (1989)	Sales to Cash Ratio
Salecash_2y	Ou and Penman (1989)	Sales to Cash Ratio
Salecash_3y	Ou and Penman (1989)	Sales to Cash Ratio
Saleinv	Ou and Penman (1989)	Sales to Inventory
Saleinv_2y	Ou and Penman (1989)	Sales to Inventory
Saleinv_3y	Ou and Penman (1989)	Sales to Inventory
Salerec	Ou and Penman (1989)	Sales to Receivables
Salerec_2y	Ou and Penman (1989)	Sales to Receivables
Salerec_3y	Ou and Penman (1989)	Sales to Receivables
Secured	Valta (2016)	Secured Debt
Securedind	Valta (2016)	Secured Debt Indicator
Sfe	Elgers, Lo, and Pfeiffer (2001)	Earnings Forecast to Price
Sgr	Lakonishok, Shleifer, and Vishny (1994)	Annual Sales Growth
Sgr_q	Lakonishok, Shleifer, and Vishny (1994)	Annual Sales Growth Quarterly

Continued on the next page

Table A.2: Details on stock characteristics (continued).

Acronym	Original Study	Characteristic Description
Sharelss1y	Pontiff and Woodgate (2008)	Share Issuance (1 Year)
Sharelss2y	Pontiff and Woodgate (2008)	Share Issuance (2 Year)
Sharelss3y	Pontiff and Woodgate (2008)	Share Issuance (3 Year)
Sharelss5y	Daniel and Titman (2006)	Share Issuance (5 Year)
Sharerepurchase	Ikenberry, Lakonishok, and Vermaelen (1995)	Share Repurchases
Sharevol	Datar, Naik, and Radcliffe (1998)	Share Volume
Shortinterest	Dechow et al. (2001)	Short Interest
Sinalgo	Hong and Kacperczyk (2009)	Sin Stock(selection Criteria)
Size	Banz (1981)	Size
Skewl	Xing, Zhang, and Zhao (2010)	Volatility Smirk Near The Money
Smileslope	Yan (2011)	Put Volatility Minus Call Volatility
Sp	Barbee, Mukherji, and Raines (1996)	Sales-to-price
Spinoff	Cusatis, Miles, and Woolridge (1993)	Spinoffs
Std_turn	Chordia, Subra, and Anshuman (2001)	Share Turnover Volatility
Std_turn_3y	Chordia, Subra, and Anshuman (2001)	Share Turnover Volatility
Streversal	Jegadeesh (1989)	Short Term Reversal
Surpriserd	Eberhart, Maxwell, and Siddique (2004)	Unexpected R&d Increase
Tang	Hahn and Lee (2009)	Tangibility
Tang_q	Hahn and Lee (2009)	Tangibility Quarterly
Tax	Lev and Nissim (2004)	Taxable Income to Income
Tax_q	Lev and Nissim (2004)	Taxable Income to Income (qtrly)
Totalaccruals	Richardson et al. (2005)	Total Accruals
Uprecomm	Barber et al. (2002)	Up Forecast
Varcf	Haugen and Baker (1996)	Cash-flow to Price Variance
Volmkt	Haugen and Baker (1996)	Volume to Market Equity
Volsd	Chordia, Subra, and Anshuman (2001)	Volume Variance
Volsd_1m	Chordia, Subra, and Anshuman (2001)	Volume Variance
Volsd_3m	Chordia, Subra, and Anshuman (2001)	Volume Variance
Volsd_6m	Chordia, Subra, and Anshuman (2001)	Volume Variance
Volsd_9m	Chordia, Subra, and Anshuman (2001)	Volume Variance
Volsd_12m	Chordia, Subra, and Anshuman (2001)	Volume Variance
Volumetrend	Haugen and Baker (1996)	Volume Trend
Volumetrend_1m	Haugen and Baker (1996)	Volume Trend
Volumetrend_3m	Haugen and Baker (1996)	Volume Trend
Volumetrend_6m	Haugen and Baker (1996)	Volume Trend
Volumetrend_9m	Haugen and Baker (1996)	Volume Trend
Volumetrend_12m	Haugen and Baker (1996)	Volume Trend
Ww	Whited and Wu (2006)	Whited-wuindex
Ww_q	Whited and Wu (2006)	Whited-wuindex
Xfin	Bradshaw, Richardson, and Sloan (2006)	Net External Financing
Zerotrade	Liu (2006)	Days With Zero Trades
Zerotradealt1	Liu (2006)	Days With Zero Trades
Zerotradealt12	Liu (2006)	Days With Zero Trades
Zscore	Dichev (1998)	Altman Z-score
Zscore Q	Dichev (1998)	Altman Z-score Quarterly

Table A.1: The average number of stocks by year

This table reports the average daily number of stocks included in our final sample for each year from 2010 to 2023.

Year	Avg Stocks No.
2010	1,841
2011	2,169
2012	2,393
2013	2,462
2014	2,527
2015	2,736
2016	2,932
2017	3,345
2018	3,612
2019	3,741
2020	4,015
2021	4,485
2022	4,869
2023	5,196

A.2 Details of Model and Training Procedure

A.2.1 Prediction layer

In our empirical analysis, we evaluate a suite of feedforward neural network (NN) architectures, drawing inspiration from Gu et al. (2020). We implement five distinct network configurations, varying in depth from one to five hidden layers, to assess the impact of model complexity on the performance of our proposed end-to-end framework. Our simplest neural network, denoted by NN1, comprises a single hidden layer with 32 neurons. We then progressively increase the depth following the geometric pyramid rule (Masters, 1993). NN2 contains two hidden layers, with 32 neurons in the first and 16 neurons in the second. NN3 employs three hidden layers, configured with 32, 16, and 8 neurons, respectively. NN4 utilizes four hidden layers, with 32, 16, 8, and 4 neurons. Finally, NN5 consists of five hidden layers, with 32, 16, 8, 4, and 2 neurons.

All NNs involve an initial Layer Normalization (Ba et al., 2016) applied directly to the

input features before they are fed into the first hidden layer. This initial normalization helps ensure that the first hidden layer receives inputs that are more consistently scaled, improving training stability and accelerating convergence. For all hidden layers, we utilize the Rectified Linear Unit (ReLU) activation function. To further enhance training dynamics, we incorporate a Batch Normalization layer (Ioffe and Szegedy, 2015) after each ReLU activation in the hidden layers. This layer normalizes the output of a hidden layer by re-centering and re-scaling the activations within each mini-batch during training. This process alleviates the "internal covariate shift" issue, where the distribution of a layer's inputs changes during training as the parameters of the preceding layers evolve. After the Batch Normalization layer, we apply Dropout with a rate of 5%, following the design of Chen et al. (2024). Dropout is a powerful regularization technique that randomly "drops" (sets to zero) a fraction of neurons and their connections during each training iteration. This prevents neurons from co-adapting too much and encourages them to learn more robust features that are useful in conjunction with different random subsets of other neurons. In effect, Dropout can be interpreted as training a large ensemble of thinned networks that share weights (Srivastava et al., 2014; Hinton et al., 2012b). Previous studies suggest that Dropout provides superior generalization performance compared to traditional weight decay methods such as L_1 or L_2 regularization, particularly for deep and complex neural networks (Chen et al., 2024; Goodfellow et al., 2016; Drobetz and Otto, 2021).

We note that the specific architectural choices and the associated hyperparameters are directly adopted from established practices and successful implementations in the machine learning asset pricing literature (Chen et al., 2024; Drobetz and Otto, 2021; Gu et al., 2020). In particular, we deliberately refrain from exhaustive hyperparameter searches in this study for several key reasons. First, our empirical study employs a rolling window scheme, under which models are retrained quarterly by shifting the training data window forward. Given the large number of retraining instances required, incorporating extensive cross-validation is computationally intensive. Additionally, our primary objective is to compare the performance of our proposed E2E framework with the traditional two-stage approach. To ensure a fair comparison, it is essential to maintain consistency in the prediction layer.

A.2.2 Training procedure

For both the traditional MSE models and our proposed E2E models, we employ a consistent training methodology incorporating best practices for model development and evaluation. Our approach utilizes a rolling-window scheme, where models are retrained at the beginning of each quarter using a shifting window of past data. Within each retraining period, the available historical data are further divided into a training set and a validation set. Typically, we allocate a fixed 9-month period of the historical data preceding the current quarter to the validation set, with the remainder forming the training set. During the training process, after each epoch, we evaluate the model's performance based on the validation set. Monitoring validation performance throughout training is widely adopted in the machine learning literature (Goodfellow et al., 2016; Bishop and Nasrabadi, 2006) as it provides a good estimate of the model's ability to generalize to unseen data.

To control the training duration and mitigate overfitting, we implement an early stopping mechanism with a fixed patience criterion. Specifically, if the model's performance on the validation set does not improve for 10 consecutive epochs, the training process is terminated, and the parameters corresponding to the best observed validation performance are retained. Early stopping is a highly effective regularization tool (Prechelt, 2002; Yao et al., 2007). By halting training when generalization performance begins to degrade or ceases to improve, it implicitly restricts the model complexity and prevents it from overfitting to the noise in the training data.

For training the deep neural networks, we employ the Adam (Adaptive Moment Estimation) optimizer (Kingma and Ba, 2014) implemented in PyTorch across all models. Adam is an adaptive learning-rate optimization algorithm that computes parameter-specific learning rates based on estimates of first and second moments of the gradients. It combines the advantages of two widely used extensions to stochastic gradient descent: AdaGrad (Duchi et al., 2011), which performs well with sparse gradients, and RMSProp (Hinton et al., 2012a), which is effective in online and non-stationary settings. Adam has demonstrated superior performance relative to traditional stochastic gradient descent (SGD) in many deep learning applications and has been widely adopted in the field of machine learning for asset pricing (Chen et al., 2024; Gu et al., 2020, 2021; Leippold et al., 2022). Following the setup in Chen et al. (2024), we set the learning rate for the Adam optimizer to 1×10^{-3} for most

of our experiments except for the risk preference examination in Section 4.4. The reason is that when the risk aversion parameter η is small, the E2E objective function becomes highly sensitive to small changes in predicted returns, leading to potentially unstable training dynamics. To ensure convergence and stable learning in these specific high-sensitivity scenarios, we reduce the learning rate to 1×10^{-4} when η is less than one.

Training E2E models requires substantially higher computational costs than classical MSE models. This increased demand arises from the E2E architecture. During each forward propagation step, the model involves solving a portfolio optimization problem for every cross-section under the predicted returns (Butler and Kwon, 2023). Additionally, during backpropagation, differentiating through this optimization layer requires computing the Jacobian of the optimal portfolio weights with respect to the predicted returns, which could be computationally intensive for large-scale problems (Amos and Kolter, 2017). To mitigate these computational demands and accelerate training, we employ a warm-start strategy. The rationale is that while a model trained purely on MSE may not align with specific downstream investment objectives, its learned parameters often capture significant underlying patterns in the data and can serve as a good initial point for the more complex E2E optimization. Therefore, in our training pipeline, we first train a standard MSE-based model. The parameters obtained from this initial MSE are then used to initialize the corresponding E2E model. This warm-start procedure effectively provides the E2E model with a more informed starting point in the parameter space, which reduces the computational time required for convergence.

For the E2E models, the upper-level objective function and its gradients are computed based on portfolio performance across all cross-sections. To manage computational resources effectively, particularly GPU memory, while still obtaining stable gradient updates, we process cross-sections in mini-batches. Specifically, we accumulate gradients over 8 cross-sections before performing a parameter update. This approach balances the need for a representative sample of portfolio decisions for gradient estimation with the practical limits of available hardware.

A.2.3 Implementation environment

All empirical tests and model training are conducted on a Linux server equipped with four NVIDIA GeForce RTX 4090 GPUs, providing substantial parallel processing capabilities

for computationally intensive tasks. The prediction models, including both the MSE-based benchmarks and our proposed end-to-end (E2E) frameworks, are implemented using the PyTorch library (Paszke et al., 2019). PyTorch is selected for its flexibility in building custom neural network architectures and its robust support for automatic differentiation, which is fundamental to our gradient-based training.

A key component of our E2E model implementation is the integration of the portfolio optimization layer into the neural network's computational graph. To achieve this, we leverage cvxpylayers Agrawal et al. (2019), a Python library that allows the differentiation of convex optimization problems. This package effectively creates a differentiable layer that can be seamlessly incorporated into standard deep learning frameworks like PyTorch.

In the forward pass, we specify Clarabel (Goulart and Chen, 2024) as the underlying solver. Clarabel is a high-performance interior-point solver for conic optimization, which supports Rust and Python. The choice of Clarabel is particularly advantageous for our E2E framework. During the forward pass, Clarabel efficiently returns not only the primal optimal solution but also the dual optimal solution (the Lagrange multipliers associated with the constraints). This enables cvxpylayers to efficiently compute the Jacobian during the backward pass.

A.3 Example of Linear Program.

In this section, we present an illustrative example for the gradient vanish issue of linear programs and the effect of the regularization approach.

Example 1. *We consider a simple portfolio selection problem with only two assets:*

$$\begin{aligned} \min \quad & -w_1\hat{r}_1 - w_2\hat{r}_2 \\ \text{s.t.} \quad & w_1, w_2 \in \mathbb{R} \\ & 0 \le w_1 \le 1,\ 0 \le w_2 \le 1, \\ & w_1 + w_2 = 1. \end{aligned} \tag{22}$$

The above linear program aims to find the asset with larger return and assign all weights to it. Due to its simplicity, one can easily observe:

1. *When $\hat{r}_1 > \hat{r}_2$, the linear program chooses the largest possible value for w_1, yielding $w_1^\star = 1$. Consequently, $w_2^\star = 0$.*

2. *When $\hat{r}_1 < \hat{r}_2$, the linear program chooses the largest possible value for w_2, yielding $w_2^\star = 1$. Consequently, $w_1^\star = 0$.*

3. *When $\hat{r}_1 = \hat{r}_2$, any $w_1 \in [0, 1]$ (with $w_2 = 1 - w_1$) is an optimal solution.*

Therefore, the optimal weights $\boldsymbol{w}^\star(\hat{\boldsymbol{r}})$ is

$$\boldsymbol{w}^\star(\hat{\boldsymbol{r}}) = \begin{cases} 1 & \text{if } \hat{r}_1 > \hat{r}_2 \\ 0 & \text{if } \hat{r}_1 < \hat{r}_2 \\ \text{any value in } [0,1] & \text{if } \hat{r}_1 = \hat{r}_2. \end{cases} \tag{23}$$

This function is piecewise constant where $\hat{r}_1 \neq \hat{r}_2$, resulting in a zero gradient in these regions. At the breakpoint where $\hat{r}_1 = \hat{r}_2$, multiple optimal solutions exist, and the gradient is undefined. A visualization of this function is presented in Figure 3(A).

On the other hand, Figure 3(B) presents the smoothed solution function obtained by solving the L_2 regularized program

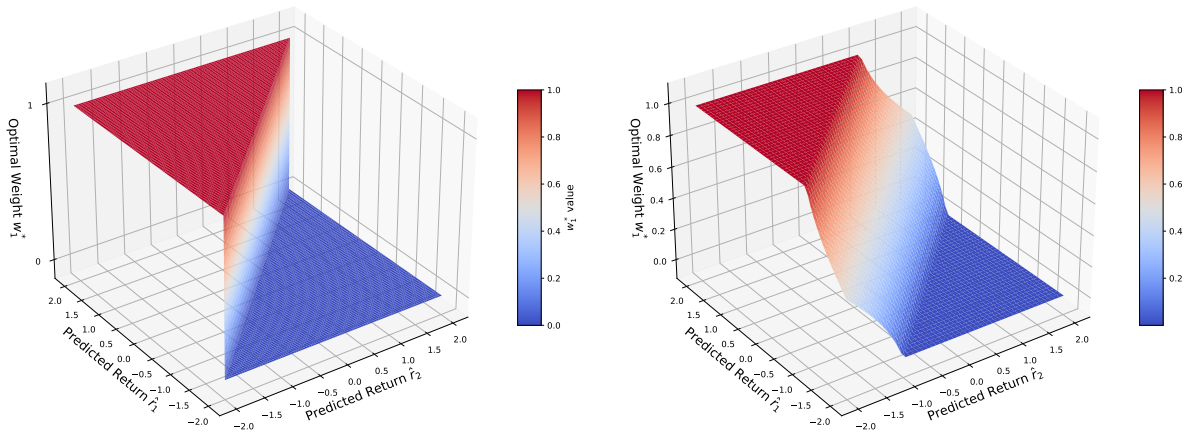
$$\begin{aligned} \min \quad & -w_1\hat{r}_1 - w_2\hat{r}_2 + \|[w_1, w_2]\|_2 \\ \text{s.t.} \quad & w_1, w_2 \in \mathbb{R} \\ & 0 \leq w_1 \leq 1,\ 0 \leq w_2 \leq 1, \\ & w_1 + w_2 = 1. \end{aligned} \tag{24}$$

Figure 3(B) shows that the solution function is a Lipschitz continuous function with well-defined gradients on its domain. With this solution function, we can easily trace the influence of the predicted return $\hat{\boldsymbol{r}}$ on the optimal portfolio allocation decision.

A.4 Additional Experimental Results

A.4.1 Additional metrics of the equal-weighted long-only portfolio

Table A.3 provides supplemental performance metrics for the long-only (High) portfolios, encompassing the linear specification as well as all five neural network architectures. These metrics include portfolio turnover, maximum single-day drawdowns (MDD-1D), and the magnitude and duration of the maximum cumulative drawdowns observed throughout the entire sample period. By reporting these additional statistics, we offer a more granular view



(a) Solution function of the linear program

(b) Solution function of the regularized linear program

Figure 3: Visualization of the solution functions.

This figure illustrates the solution functions for two cases: the left panel A shows the linear program, and the right panel B shows the regularized linear program.

of the risk characteristics and operational efficiency of the strategies developed under both the E2E and MSE-based frameworks.

The empirical evidence demonstrates that the E2E framework systematically delivers higher daily average returns compared to the traditional MSE-based approach across every model specification. This outperformance is statistically significant at the 5% level or better in all six cases. Coupled with the generally lower turnover rates observed in E2E portfolios, these results reinforce the framework's superior ability to internalize investment objectives and produce more robust, implementable long-only strategies.

A.4.2 Value-weighted portfolios

While our main results are based on equal-weighted portfolios, concerns may arise regarding the practical relevance of this approach, particularly for institutional investors. In real-world settings, especially for large asset managers, portfolio allocations are often scaled by firm size, and value-weighted strategies may better reflect implementation capacity and market impact. To address this concern and assess the robustness of our findings, we recon-

Table A.3: Additional performance metrics for the long-only (High) portfolios

This table reports additional performance metrics for the long-only (High) portfolios. Turnover is the average percentage change in portfolio holdings. MDD-1D refers to the maximum single-day drawdowns, while MDD denotes the maximum drawdowns over the entire sample period. We also report the corresponding start and end dates (MDD Begin and MDD End) as well as the duration (MDD Length) of the maximum drawdowns. t-Stats represents mean difference tests for daily average portfolio returns between E2E and MSE, where $**$ indicates that E2E outperforms MSE at the 5% significance level or better.

		Turnover %	MDD-1D %	MDD %	MDD Begin	MDD End	Length	t-stats
Linear	MSE-L	53.2	7.95	22.54	2015-06-19	2015-07-06	18	4.27**
	E2E	49.2	8.24	23.43	2015-06-10	2015-07-06	27	
NN1	MSE	58.0	8.51	23.59	2015-06-19	2015-07-06	18	3.51**
	E2E	56.0	8.25	22.44	2015-06-19	2015-07-06	18	
NN2	MSE	60.0	8.37	23.10	2015-06-19	2015-07-06	18	4.11**
	E2E	57.4	8.55	22.66	2015-06-19	2015-07-06	18	
NN3	MSE	60.6	8.27	22.42	2015-06-19	2015-07-06	18	1.74**
	E2E	58.6	8.31	23.59	2015-06-19	2015-07-06	18	
NN4	MSE	57.5	8.27	21.23	2015-06-19	2015-07-06	18	2.68**
	E2E	54.9	8.62	23.94	2015-06-19	2015-07-06	18	
NN5	MSE	58.0	8.27	23.19	2015-06-19	2015-07-06	18	3.49**
	E2E	54.8	8.10	23.20	2015-06-19	2015-07-06	18	

duct the main tests using value-weighted portfolios, where weights are proportional to firm size.

Results are summarized in Tables A.4 and A.5, which are consistent with the main findings reported in Tables 2 and A.3. We compare the E2E framework with the two-stage MSE approach across six model specifications, including one linear model and five neural network models (NN1 through NN5). In all cases, the E2E framework continues to outperform MSE within the long-only (H) portfolio group, delivering higher annualized returns and Sharpe ratios. The improvements in portfolio returns and Sharpe ratios are approximately 12% and 10%, respectively, closely matching the results based on equal-weighted portfolios. This consistency reinforces the robustness of our findings and suggests that the performance advantage of the E2E framework is not sensitive to the choice of weighting scheme.

However, we do observe that compared to the equal-weighted portfolios, overall performance tends to be lower, regardless of whether the model is MSE or E2E, or whether it is linear or nonlinear. The return differences between E2E and MSE also become less statistically significant. This finding is consistent with the literature suggesting that abnormal returns are more likely to be concentrated in small firms. When using value-weighted portfolios, the weights assigned to small-cap stocks are reduced, which in turn dampens the overall portfolio performance.

A.4.3 Robustness check: excluding microcaps

In this section, following the setup of Leippold et al. (2022), we exclude the bottom 30% of stocks by market capitalization and re-examine the performance of the E2E model and the MSE model. Microcap stocks warrant special attention in asset management research for several reasons. First, due to their lower trading volumes, investing in microcaps can incur higher transaction costs and price impact in practice. In addition, small-cap stocks in the Chinese market may be influenced by a shell-value premium (Liu et al., 2019). To assess whether the superior performance of the E2E framework relative to the MSE model is robust, we focus our analysis on the universe of stocks in the top 70% of market capitalization.

Table A.6 and Table A.7 details the portfolio performance across all six predictive models. The results consistently mirror our main findings from Section 4.2: the E2E framework systematically outperforms the traditional MSE approach. Across every model specification, from the linear model to the most complex neural network (NN5), the E2E framework delivers

higher annualized returns and superior Sharpe ratios for the long-only (High) portfolio. Furthermore, the t-statistics for the return difference between the E2E and MSE models are statistically significant across all six model specifications, consistently meeting the 5% significance level. These results demonstrate that the E2E model's advantage is not merely an artifact of exploiting hard-to-trade anomalies concentrated in the microcap segment. Even within a universe of larger and more liquid stocks, the decision-aware learning process remains more effective.

A.4.4 Factor importance with varying transaction cost rates

To further explore how the E2E framework adapts to rising transaction cost rates, we examine the changes in factor importance[25] across different cost levels. Given the large number of pricing characteristics (415 in total), for clarity, Figure 4 displays the 100 characteristics that exhibit the highest average importance across all cost levels tested. Each column in the figure corresponds to a specific TCR. The color gradient reflects the rank of relative importance of each factor at that cost level, where darker blue signifies higher importance, and white indicates minimal influence. The key insight from Figure 4 is the dynamic nature of factor importance within the E2E framework. As the TCRs increase from left to right, the model adjusts its reliance on different signals. Some characteristics that are highly important at low or zero transaction costs become less influential as trading frictions increase, as indicated by their lighter shading. In contrast, other factors gain prominence, suggesting that they are associated with more stable, lower-turnover signals.

This behavior highlights an important advantage of our approach and stands in contrast to the traditional MSE framework. An MSE model, trained solely to minimize prediction errors without awareness of downstream costs, would produce a single static ranking of factor importance. In contrast, the E2E model demonstrates its ability to adapt the pricing function itself in response to changes in the optimization objective. By learning to prioritize different signals under varying cost assumptions, the E2E framework produces strategies that are not only predictive, but also economically robust and implementable.

[25]Following Gu et al. (2020), we measure factor importance using the sum of squared partial derivatives (SSD). This metric, proposed by Dimopoulos et al. (1995), is calculated as the sum of the squared partial derivatives of the model's output with respect to each input feature. It effectively quantifies the sensitivity of the model's predictions to changes in a particular factor, with a higher value indicating greater importance.

Overall, these results highlight the importance of explicitly modeling transaction costs within the learning process. Compared to the two-stage MSE approach, the E2E framework delivers higher net returns, reduces trading intensity, mitigates drawdown risk, and produces more robust and implementable strategies. In addition, we shed light on how E2E achieves this outperformance by adaptively adjusting factor importance in response to transaction costs, effectively shifting weight toward lower-turnover signals as trading frictions increase.

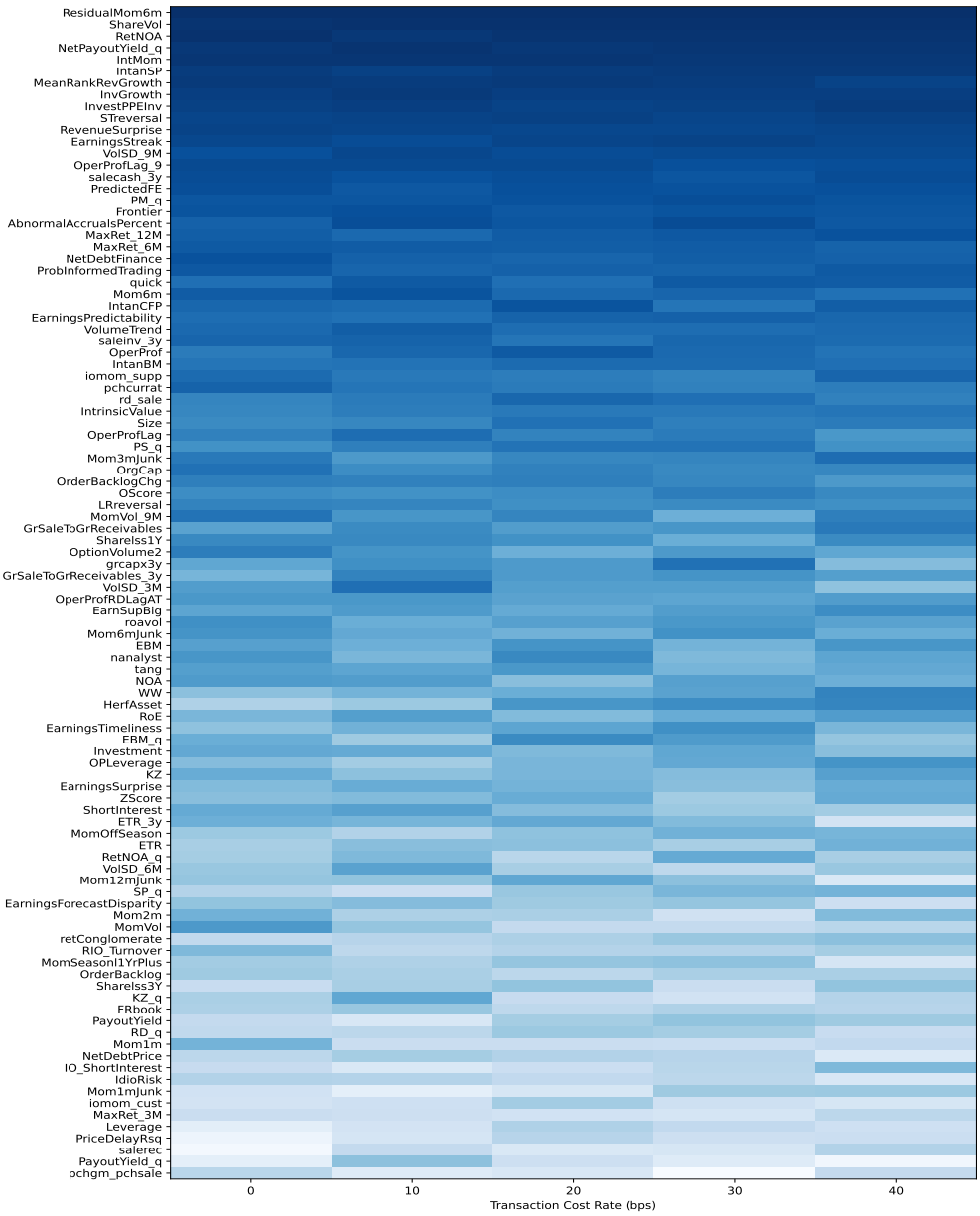


Figure 4: Changing signal importance with transaction costs

This figure shows how factor importance changes as transaction cost rates (TCRs) increase from 0 to 40 basic points. Out of 415 characteristics, we select and report the top 100 characteristics with the highest average importance across all tested TCR levels. Columns correspond to different levels of transaction costs. The color gradient within each column reflects the relative importance of each factor at a given cost level, with darker blue indicating higher importance and white indicating minimal influence.

Table A.4: Value-weighted portfolio performance of both MSE and E2E

This table reports value-weighted portfolio performance under the MSE and E2E frameworks across a linear model and five neural network architectures (NN1–NN5), where weights are proportional to firm size. Avg and SD denote the annualized daily average return and standard deviation (percent), and SR is the annualized Sharpe ratio. Models are trained on daily stock returns using a rolling-window procedure that advances quarterly. Each window spans four years, with 39 months for training, 9 months for validation, and the subsequent quarter for out-of-sample prediction. At the start of each test quarter, stocks are sorted into deciles based on predicted one-quarter-ahead returns, from highest (H) to lowest (L), with H–L representing the corresponding long–short portfolio.

		MSE			E2E					MSE			E2E		
	Groups	Avg %	SD %	SR	Avg %	SD %	SR		Groups	Avg %	SD %	SR	Avg %	SD %	SR
Linear	High(H)	29.1	20.6	1.42	33.1	20.9	1.59	NN3	High(H)	34.7	20.7	1.68	36.6	20.8	1.76
	2	16.1	19.7	0.82	24.1	19.7	1.23		2	21.4	19.7	1.09	22.3	19.9	1.11
	3	11.7	19.1	0.61	15.5	19.3	0.80		3	14.4	19.3	0.75	14.5	19.2	0.75
	4	7.7	18.6	0.41	10.9	19.2	0.57		4	9.5	18.9	0.50	10.1	18.9	0.53
	5	5.5	18.7	0.29	7.7	19.0	0.40		5	6.3	19.1	0.33	5.9	18.9	0.31
	6	1.0	18.6	0.05	1.0	18.7	0.05		6	3.9	18.8	0.20	0.0	18.9	0.00
	7	-4.8	18.9	-0.25	-2.8	18.8	-0.15		7	-4.7	18.7	-0.25	-2.8	18.9	-0.15
	8	-7.0	18.8	-0.37	-9.3	18.9	-0.49		8	-10.8	18.9	-0.57	-8.5	19.0	-0.44
	9	-15.8	19.1	-0.83	-16.2	19.2	-0.85		9	-20.5	19.1	-1.07	-13.8	19.1	-0.72
	Low(L)	-39.8	20.6	-1.93	-31.7	21.0	-1.51		Low(L)	-56.1	20.6	-2.73	-42.7	20.8	-2.05
	H-L	34.4	9.5	3.62	32.4	10.3	3.14		H-L	45.4	9.1	4.97	39.5	10.0	3.94
NN1	High(H)	35.0	20.6	1.70	39.2	20.8	1.88	NN4	High(H)	32.6	20.8	1.57	37.5	20.9	1.80
	2	18.8	19.5	0.97	25.3	20.0	1.27		2	22.0	19.8	1.11	21.9	19.9	1.10
	3	14.5	19.1	0.76	19.0	19.3	0.98		3	12.9	19.2	0.67	16.3	19.3	0.84
	4	10.6	18.9	0.56	12.3	18.8	0.65		4	11.0	19.3	0.57	6.7	19.2	0.34
	5	5.5	18.8	0.29	8.3	18.5	0.45		5	8.6	18.9	0.45	4.6	19.2	0.24
	6	4.9	18.5	0.27	3.0	18.8	0.16		6	0.4	18.9	0.02	2.6	19.3	0.13
	7	-2.7	18.7	-0.14	-1.2	18.7	-0.07		7	-2.6	19.0	-0.13	-2.3	19.0	-0.12
	8	-10.3	18.7	-0.55	-8.0	18.7	-0.43		8	-6.1	18.9	-0.32	-4.6	18.8	-0.24
	9	-19.5	19.1	-1.02	-13.9	19.4	-0.72		9	-19.4	19.4	-0.99	-12.6	19.7	-0.64
	Low(L)	-54.5	21.0	-2.59	-38.8	20.8	-1.87		Low(L)	-58.8	21.0	-2.80	-44.0	20.6	-2.13
	H-L	44.8	9.4	4.78	39.0	9.9	3.95		H-L	45.7	9.4	4.88	40.8	9.7	4.19
NN2	High(H)	33.3	20.6	1.61	37.5	21.1	1.77	NN5	High(H)	35.0	20.3	1.73	37.1	20.7	1.80
	2	22.8	19.5	1.17	26.1	20.0	1.31		2	23.5	19.8	1.19	24.4	19.7	1.24
	3	15.4	19.3	0.80	15.4	19.3	0.80		3	15.0	19.6	0.77	10.1	19.3	0.52
	4	10.7	19.3	0.55	12.7	19.2	0.66		4	7.7	18.9	0.40	9.7	19.1	0.51
	5	3.3	19.1	0.17	7.1	18.8	0.37		5	5.4	18.9	0.28	8.1	18.9	0.43
	6	2.8	18.8	0.15	3.0	18.8	0.16		6	3.3	19.2	0.17	-1.2	19.2	-0.06
	7	-4.2	19.1	-0.22	-3.0	19.0	-0.16		7	-4.1	18.9	-0.22	-1.2	19.0	-0.06
	8	-12.5	18.9	-0.66	-7.5	19.2	-0.39		8	-9.6	18.9	-0.51	-7.9	19.0	-0.42
	9	-18.6	19.0	-0.98	-17.2	19.4	-0.89		9	-21.4	19.2	-1.12	-19.1	19.3	-0.99
	Low(L)	-55.8	20.9	-2.68	-40.1	20.5	-1.95		Low(L)	-52.9	21.3	-2.49	-40.7	21.2	-1.92
	H-L	44.5	9.2	4.86	38.8	9.7	3.99		H-L	43.9	9.7	4.54	38.9	9.7	4.03

Table A.5: Additional performance metrics for the value-weighted long-only portfolios

This table reports additional performance metrics for the value-weighted long-only portfolios. Turnover is the average percentage change in portfolio holdings. MDD-1D refers to the maximum single-day drawdowns, while MDD denotes the maximum drawdowns over the entire sample period. We also report the corresponding start and end dates (MDD Begin and MDD End) as well as the duration (MDD Length) of the maximum drawdowns. t-Stats represents mean difference tests for daily average portfolio returns between E2E and MSE, where $**$ and $*$ indicates that E2E outperforms MSE at the 5% and 10% significance level, respectively.

		Turnover %	MDD-1D %	MDD %	MDD Begin	MDD End	Length	t-stats
Linear	MSE	58.9	7.69	31.80	2015-06-19	2015-08-24	67	1.52^{*}
	E2E	56.1	7.91	23.85	2015-06-08	2015-07-30	53	
NN1	MSE	64.5	9.08	27.95	2015-06-19	2015-07-30	42	1.43^{*}
	E2E	62.0	8.78	23.25	2015-06-19	2015-07-30	42	
NN2	MSE	66.8	8.14	31.10	2015-05-28	2015-07-30	64	1.66^{**}
	E2E	64.1	8.28	26.63	2015-05-28	2015-07-30	64	
NN3	MSE	67.1	8.47	26.01	2015-06-08	2015-07-30	53	0.88
	E2E	63.9	8.53	21.06	2015-06-08	2015-07-06	29	
NN4	MSE	64.4	8.41	18.57	2021-08-02	2021-12-31	152	1.67^{**}
	E2E	62.0	8.32	21.90	2015-06-10	2015-07-06	27	
NN5	MSE	65.2	8.34	21.39	2015-06-19	2015-07-06	18	0.81
	E2E	61.7	8.39	21.09	2015-06-19	2015-07-06	18	

Table A.6: Equal-weighted portfolio performance of both MSE and E2E

This table reports equal-weighted portfolio performance under the MSE and E2E frameworks across a linear model and five neural network architectures (NN1–NN5), after excluding the bottom 30% of stocks by market capitalization. Avg and SD denote the annualized daily average return and standard deviation (percent), and SR is the annualized Sharpe ratio. Models are trained on daily stock returns using a rolling-window procedure that advances quarterly. Each window spans four years, with 39 months for training, 9 months for validation, and the subsequent quarter for out-of-sample prediction. At the start of each test quarter, stocks are sorted into deciles based on predicted one-quarter-ahead returns, from highest (H) to lowest (L), with H–L representing the corresponding long–short portfolio.

		MSE			E2E					MSE			E2E		
	Groups	Avg %	SD %	SR	Avg %	SD %	SR		Groups	Avg %	SD %	SR	Avg %	SD %	SR
Linear	High(H)	28.0	19.5	1.44	34.2	20.2	1.69	NN3	High(H)	33.6	20.3	1.65	37.8	20.2	1.87
	2	19.8	18.9	1.05	20.8	19.1	1.09		2	19.6	19.0	1.03	21.4	18.8	1.14
	3	13.8	18.5	0.75	15.1	18.7	0.81		3	15.2	18.4	0.82	14.1	18.4	0.77
	4	9.4	18.3	0.51	7.8	18.4	0.42		4	10.3	18.0	0.57	8.6	18.1	0.47
	5	5.2	18.1	0.28	2.5	18.2	0.13		5	5.5	18.0	0.31	4.6	18.0	0.26
	6	0.0	18.0	0.00	-2.6	18.0	-0.14		6	1.5	17.8	0.08	0.5	18.0	0.03
	7	-4.8	18.0	-0.27	-7.0	17.9	-0.39		7	-3.1	17.8	-0.18	-5.7	17.8	-0.32
	8	-11.4	17.8	-0.64	-14.9	17.7	-0.84		8	-10.6	17.9	-0.59	-12.1	17.9	-0.68
	9	-21.9	17.9	-1.22	-24.8	17.7	-1.41		9	-25.4	18.0	-1.41	-25.7	18.0	-1.43
	Low(L)	-60.2	18.9	-3.19	-54.4	18.4	-2.90		Low(L)	-68.4	19.1	-3.58	-65.5	19.0	-3.45
	H-L	44.1	6.6	6.67	43.8	7.1	6.21		H-L	51.0	6.4	8.02	51.7	6.3	8.19
NN1	High(H)	33.1	20.1	1.65	36.9	20.1	1.83	NN4	High(H)	31.2	20.5	1.52	34.7	20.2	1.71
	2	19.1	18.9	1.01	21.3	18.8	1.13		2	21.4	18.9	1.13	20.0	18.9	1.06
	3	15.7	18.3	0.85	14.3	18.3	0.78		3	14.3	18.4	0.78	11.7	18.4	0.64
	4	10.7	18.1	0.59	9.3	18.2	0.51		4	9.8	18.3	0.53	6.9	18.1	0.38
	5	6.0	18.0	0.33	4.1	18.0	0.23		5	6.7	18.1	0.37	3.6	18.0	0.20
	6	1.0	17.8	0.05	0.0	17.8	0.00		6	2.3	17.8	0.13	1.6	17.9	0.09
	7	-3.1	17.8	-0.17	-7.1	17.9	-0.40		7	-3.5	17.8	-0.20	-5.1	17.8	-0.29
	8	-11.4	17.9	-0.64	-14.4	17.8	-0.81		8	-9.7	17.8	-0.54	-11.7	17.8	-0.66
	9	-22.6	18.2	-1.25	-24.4	18.0	-1.36		9	-23.9	18.0	-1.33	-23.2	18.1	-1.28
	Low(L)	-70.4	19.3	-3.64	-62.1	19.0	-3.27		Low(L)	-70.4	19.4	-3.63	-60.7	19.3	-3.15
	H-L	51.8	6.5	7.95	49.5	6.3	7.81		H-L	50.8	6.6	7.72	47.7	6.1	7.87
NN2	High(H)	31.2	20.2	1.54	35.3	20.2	1.75	NN5	High(H)	32.3	20.1	1.61	36.4	20.3	1.79
	2	21.1	19.0	1.11	21.3	18.9	1.13		2	20.5	18.9	1.08	19.1	18.8	1.02
	3	15.8	18.5	0.86	14.7	18.4	0.80		3	14.9	18.4	0.81	11.5	18.3	0.63
	4	10.4	18.2	0.57	10.0	18.1	0.55		4	8.9	18.2	0.49	7.7	18.1	0.42
	5	5.1	17.9	0.29	4.7	17.9	0.26		5	5.9	17.9	0.33	2.9	17.9	0.16
	6	0.8	17.9	0.04	-0.4	17.8	-0.02		6	2.7	17.9	0.15	-0.2	17.9	-0.01
	7	-3.8	17.8	-0.22	-5.4	17.8	-0.30		7	-3.5	18.0	-0.19	-2.5	18.0	-0.14
	8	-12.4	17.8	-0.70	-12.4	17.7	-0.70		8	-10.7	17.8	-0.60	-11.8	18.1	-0.66
	9	-22.4	18.0	-1.24	-25.4	18.0	-1.41		9	-26.2	18.0	-1.45	-24.7	18.1	-1.36
	Low(L)	-67.8	19.3	-3.52	-64.5	19.1	-3.37		Low(L)	-66.7	19.6	-3.40	-60.3	19.5	-3.10
	H-L	49.5	6.4	7.73	49.9	6.3	7.99		H-L	49.5	6.5	7.62	48.3	6.3	7.72

Table A.7: Additional performance metrics for the long-only (High) portfolios

This table reports additional performance metrics for the equal-weighted long-only portfolios, after excluding the bottom 30% of stocks by market capitalization. Turnover is the average percentage change in portfolio holdings. MDD-1D refers to the maximum single-day drawdowns, while MDD denotes the maximum drawdowns over the entire sample period. We also report the corresponding start and end dates (MDD Begin and MDD End) as well as the duration (MDD Length) of the maximum drawdowns. t-Stats represents mean difference tests for daily average portfolio returns between E2E and MSE, where $**$ indicates that E2E outperforms MSE at the 5% significance level or better.

		Turnover %	MDD-1D %	MDD %	MDD Begin	MDD End	Length	t-stats
Linear	MSE	53.9	8.13	20.57	2015-06-19	2015-07-06	18	3.44^{**}
	E2E	49.8	8.30	21.90	2015-06-19	2015-07-06	18	
NN1	MSE	58.4	9.01	22.92	2015-06-19	2015-07-06	18	3.07^{**}
	E2E	56.4	8.64	21.19	2015-06-19	2015-07-06	18	
NN2	MSE	60.3	8.50	21.61	2015-06-19	2015-07-06	18	3.48^{**}
	E2E	57.7	8.73	21.50	2015-06-19	2015-07-06	18	
NN3	MSE	60.9	8.57	21.68	2015-06-19	2015-07-06	18	3.59^{**}
	E2E	58.8	8.55	22.74	2015-06-19	2015-07-06	18	
NN4	MSE	57.2	9.03	21.21	2015-06-19	2015-07-06	18	2.72^{**}
	E2E	54.8	8.77	22.90	2015-06-19	2015-07-06	18	
NN5	MSE	58.5	8.80	22.37	2015-06-19	2015-07-06	18	2.84^{**}
	E2E	55.4	8.68	21.86	2015-06-19	2015-07-06	18	