Trading Volume Alpha[*]

Ruslan Goyenko
McGill

Bryan Kelly
Yale, AQR, and NBER

Tobias Moskowitz
Yale, AQR, and NBER

Yinan Su
Johns Hopkins

Chao Zhang
HKUST (GZ)

June 12, 2025

Abstract

Rather than focusing on predicting asset return moments, we model the economic benefits of predicting individual stock trading volume. We translate volume forecasts into a component of expected trading costs and analyze their value through a portfolio framework. By recasting the volume prediction problem into a portfolio optimization problem that trades off tracking error versus net-of-cost performance, we quantify volume predictions into economic outcomes. Incorporating the economic loss function directly into a machine learning algorithm yields better out-of-sample performance than commonly used statistical loss functions. While volume is only one component of what drives trading costs, it is highly predictable, readily available, and its economic benefits are as large as those from stock return predictability.

Keywords: machine learning, AI, neural networks, portfolio optimization, trading volume, trading costs, investments
JEL Codes: C45, C53, C55, G00, G11, G12

[*] We are grateful to the conference and seminar participants at NBER LTAM, AFE "AI and Finance" Session at ASSA meetings, Columbia-RFS AI in Finance Conference, Cornell, Syracuse, CityU Hong Kong, and George Mason; discussants Andrew Chen, Dmitriy Muravyev, and Gustavo Schwenkler; and Hui Chen, Martin Lettau, Lu Lu, Andrew Patton, and Annette Vissing-Jorgensen; for valuable comments and suggestions. We thank Zhongji Wei and Andy Yang for their excellent research assistance. AQR Capital Management is a global investment management firm, which may or may not apply similar investment techniques or methods of analysis as described herein. The views expressed here are those of the authors and not necessarily those of AQR. Send correspondence to Bryan Kelly, bryan.kelly@yale.edu.

1 Introduction

Research in asset pricing primarily focuses on predicting moments of asset returns.[1] However, other non-return forecastable variables may be valuable for asset pricing and investment decisions. We focus on one such variable – asset-specific trading volume. While we are hardly the first to look at trading volume in an asset pricing context,[2] we take a unique approach by examining the economic consequences of *predicting* volume.

Quantifying the economic benefits of non-return variable prediction is challenging and requires connecting – through theory or empirical work – the stock characteristic to an economic outcome, which has typically been a moment of returns. However, real-world implementation costs also play a critical role in the efficacy of portfolios. While the benefits and pitfalls of return moment forecasts are well-studied in the literature, trading costs have received relatively little attention, and *forecasting* costs almost no attention.[3] Given the literature arguing that return predictability decays over time (McLean and Pontiff, 2016; Linnainmaa and Roberts, 2018; Chen and Zimmermann, 2020), and that the search for return prediction has been oversaturated (Harvey, Liu, and Zhu, 2016; Jensen, Kelly, and Pedersen, 2022), focusing on other, non-return predictors valuable for asset pricing may be a productive exercise.

We connect trading volume prediction to expected trading costs, and recast the volume prediction problem into a portfolio problem. This recasting does several things: First, it allows us to quantify volume prediction directly into an economic outcome (e.g., trading costs or net-of-cost Sharpe ratio of an investment strategy). Second, by transforming the prediction problem in terms of trading costs, we obtain better/more valuable forecasts than we do with standard statistical objectives in mind. Incorporating an economic objective provides "transfer learning" to the prediction problem that we find generates significant improvement in out-of-sample portfolio performance. We term this improvement "trading volume alpha," and show it can be sizeable.

[1] A long literature in asset pricing focuses on mean return prediction, with summaries on the state of this literature, including some of its criticisms, found in Harvey, Liu, and Zhu (2016); McLean and Pontiff (2016); Jensen, Kelly, and Pedersen (2022). A good summary of the literature on return volatility prediction is Engle (2004).

[2] See literature review in Section 2.

[3] Several papers focus on trading costs from a theoretical perspective: Kyle (1985); Gârleanu and Pedersen (2016). Even less work has provided empirical estimates of trading costs for use in a portfolio optimization context: Frazzini, Israel, and Moskowitz (2012, 2018).

We model a portfolio problem that seeks to maximize net-of-cost performance using a mean-variance utility function, but where we model the cost of transacting as a simple function of trading volume. The optimization trades off the cost of trading versus the (opportunity) cost of not trading – minimizing trading costs versus minimizing tracking error to the before-cost optimal portfolio – where trading costs and tracking error are endogenously negatively related.

We impose a functional form for trading costs due to price impact motivated by theory (Kyle, 1985) and empirical evidence (Frazzini, Israel, and Moskowitz, 2018), where price impact (Kyle's Lambda) is an increasing linear function of the trader's size of trade relative to the volume of trade in the market, termed the "participation rate." All else equal, a higher predicted volume allows the trader to trade more aggressively (larger size) because price impact per dollar traded will be lower. Conversely, a lower predicted volume causes the trader to scale back the trade (even perhaps to zero and not trade) because the price impact per dollar will be higher. We model trading costs and benefits as a tracking error problem in order to abstract from return or variance prediction. We take the moments of security returns as given, which greatly simplifies the problem, but, more importantly, allows us to focus exclusively on volume prediction and its economic impact. Reframing the optimization as a tracking error problem avoids the well-trodden ground and usual pitfalls of return prediction, which we do not want confounded with volume prediction.

Importantly, our goal is *not* to provide the best or most reliable trading cost model or forecast. Rather, our more modest objective is to simply translate volume prediction into an economic primitive for portfolio prediction. Our aim is to provide a general forecast of costs for trading an individual stock in order to quantify the value of predicting a stock's volume. For this task, we purposely choose a simple, but generic, model for expected costs.

Of course, predicting trading costs, particularly at the individual stock level, is both complicated and challenging. The biggest component of costs for a large investor is price impact, which depends on the size of the trade, the amount traded by other traders in that security, and the urgency and identity of the trader (Frazzini, Israel, and Moskowitz, 2018).[4] Each trader may face their own

[4] While many different trading cost models exist, they universally contain these three elements: trade size, market size, and trader specifics (identify, information, motive, patience, etc.). Theoretical foundations of trading costs in the classic market microstructure literature include asymmetric information (Kyle, 1985; Glosten and Milgrom, 1985)

cost function and the size of the trade is endogenously a function of both expected risk and return contribution to the portfolio and expected trading costs, all of which make estimating expected costs (enormously) challenging. Our simple model, while limited, has some modeling benefits by circumventing these challenging issues. First, we avoid the return prediction problem by focusing on tracking error. Second, we focus only on the component of costs that is neither trader-specific nor endogenous – the level of total trading volume in the stock. This allows us to write a generic expected trading cost model. Holding trade size constant, the less trading volume in the stock the greater the price impact from a trade. Thus, by only forecasting trading volume for each security, we can translate volume into a component of expected trading cost. Acknowledging, and despite, the shortcomings of this simple trading cost model, we demonstrate the power of this translation for quantifying the importance of volume prediction. A more sophisticated trading cost model may yield even greater portfolio benefits, but our simple approach illustrates how powerful volume prediction can be when couched in economic terms. Moreover, this translation enables after-cost portfolio modeling using only widely available volume data without having to find trader-specific and limited-access trading cost data (Frazzini, Israel, and Moskowitz, 2018).

An interesting feature that emerges from the model is that the economic value of volume prediction is asymmetric. Price impact costs are linear in participation rate, but non-linear in trading volume. Very low trading volume implies exponentially high impact costs, whereas very high volume implies negligible costs. As volume tends to zero, price impact costs approach infinity, whereas when volume becomes large, impact costs are bounded by zero. Hence, predicted changes in volume have much more economic impact when volume is low versus high, thus creating asymmetric costs of volume forecast errors. Conversely, tracking error, or the opportunity cost of not trading, is independent of volume. The combination of these two effects implies the optimization will penalize overestimating volume more than underestimating volume. Trading too much when you overestimate volume is more costly than trading too little when you underestimate it. The cost of trading with respect to volume is very steep at low volume and very flat at high volume. Intuitively, an illiq-

and inventory costs (Stoll, 1978; Ho and Stoll, 1983; Grossman and Miller, 1988). In either case, volume is the major determinant of liquidity (Benston and Hagerman, 1974; Glosten and Harris, 1988; Brennan and Subrahmanyam, 1995) and is often used to proxy for trading costs or liquidity (Campbell, Grossman, and Wang, 1993; Datar, Naik, and Radcliffe, 1998; Amihud, 2002).

uid stock's price impact is very sensitive to small changes in volume, and in a highly non-linear way because participation rates move by orders of magnitude for small changes in volume. Conversely, a highly liquid security's price is fairly inelastic to changes in volume. As a result, the optimization seeks to be conservative, rather than aggressive.[5] These features lead to "inaction regions" where volume may be low enough that trading is not optimal for a range of values. Conversely, there are regions where volume is high enough that price impact is negligible for a wide range of values.

Since participation rate drives trading costs, it is not only trading volume that matters, but also the size of the trade. Trade size is determined endogenously and is a function of expected trading volume and aversion to tracking error. In the model, the size (assets under management, AUM) and volatility of the fund (and risk aversion of the investor) also affect the costs and benefits of trading. Because price impact is an increasing function of participation rate, trading costs increase with AUM endogenously, and the relative penalty for tracking error decreases with AUM. The optimal tradeoff between trading costs and tracking error will therefore vary with the size of the portfolio, and so will the economic impact of volume prediction. For small AUM, tracking error considerations likely dominate trading cost considerations, hence the economic benefit to predicting volume is relatively less valuable. For large AUM, trading cost considerations dominate.[6]

Applying the model to data, we run a series of trading experiments for optimally designed portfolios that take into account trading costs using stock-level volume prediction as the sole input. Since liquidity is an unknown quantity to the portfolio manager, she uses volume prediction as an input to alter the expected cost and benefit of trading, endogenously responding to her forecast of volume by altering her portfolio. We assess the out-of-sample performance of this optimal portfolio, net of trading costs. More accurate volume predictions provide more efficient implementation and hence more efficient portfolios net of costs.

We experiment with target positions to mimic realistic trading tasks. We start by simulating a

[5] Our framework, and its implication that portfolio optimization will seek to trade conservatively rather than aggressively because of the asymmetric cost of volume forecast errors, implies that arbitrage activity may be limited as a consequence. This implication provides a novel and additional source of limits to arbitrage activity in the spirit of Shleifer and Vishny (1997).

[6] From this analysis, it is also possible to assess the optimal dollar size of the portfolio, including the break-even fund size where trading costs exactly offset portfolio returns or the fund size that maximizes after-cost dollars rather than returns (Frazzini, Israel, and Moskowitz, 2018).

hypothetical, extremely profitable (*before-cost*) daily quantitative trading strategy. This portfolio, which requires high turnover and aggressive trading, is much less profitable after accounting for trading costs. We then consider expected trading costs in the optimization to maximize the after-cost performance of the fund (at various fund sizes). Expected trading costs dictate how frequently to trade, which stocks to trade, and how much to trade, with each of these choices simultaneously affecting tracking error. We also target a host of factor portfolios from Jensen, Kelly, and Pedersen (2022) based on ex-ante return-predicting characteristics from the literature.

We find that volume prediction has a measurable and economically significant effect on after-cost portfolio performance. To predict volume, we use technical signals, such as lagged returns and lagged trading volume, as well as firm characteristics that the literature finds capture return anomalies (though not necessarily trading volume). We then add indicators for various market-wide or firm-level events associated with volume movements, including upcoming and past earnings releases. We analyze both linear and non-linear prediction methods using various neural networks designed to maximize out-of-sample predictability. Finally, we alter the objective/loss function of the neural network to take into account the portfolio problem's economic objective when predicting volume.[7]

We find that volume prediction improves significantly over moving averages of lagged trading volume when using technical signals. Adding firm characteristics (such as BE/ME) further improves volume predictability, even though these variables are primarily used for return forecasting. Information on events such as earnings releases further enhances volume predictability. Non-linear functions from neural network searches provide even better predictability over simple linear models, controlling for the same set of variables/information. Finally, recurrent neural networks, which learn dynamic predictive relationships, yield even further improvements.

Perhaps most interestingly, we find that imposing an economic loss function consistent with the portfolio problem greatly improves the value of volume predictability over statistical loss func-

[7] Once again, our aim is not to conduct an exhaustive search to find the best prediction model for stock trading volume, but rather to couch the prediction problem into economic outcomes and measure its costs and benefits accordingly. For example, we leave out many potential variables that relate to trading volume, such as other microstructure variables, cross-security lead-lag effects, etc. To that end, we assess a number of different models and variables for predicting volume in order to highlight how different prediction methods lead to different economic consequences.

tions. When fine-tuning the neural networks on the economic objective (derived from the portfolio problem) rather than on a statistical objective, such as mean squared errors used to pre-train the volume prediction model, we find marked out-of-sample portfolio improvement. This improvement occurs because the portfolio problem recognizes that trading costs are not a linear function of trading volume. The neural network places greater weight on observations that impact trading costs more and worries less about predicting volume where trading costs are less affected, such as recognizing the asymmetric costs of over- versus underestimating volume, as well as regions where volume prediction error has negligible impact (i.e., very low or very high volume levels). While an MSE objective criterion may maximize the out-of-sample R^2 of volume prediction, an economic loss function directly tied to the portfolio problem provides *more valuable* volume forecasts that results in better out-of-sample after-cost portfolio performance.

Since portfolio size affects the tradeoff between the cost and benefit of trading, solutions change with different levels of AUM. In addition, while volume prediction has benefits across all factors, some factors have greater "trading volume alpha" than others due to the varying tradeoff between the cost of trading and the opportunity cost of not trading across factors. Intuitively, factors with higher turnover (e.g., momentum, short-term reversals) benefit more from after-cost portfolio optimization that has better volume forecasts.

In general, we find that "trading volume alpha" is substantial, and as large as finding return alpha. For example, for a $1 billion fund, the after-cost improvement in portfolio performance due solely to trading volume prediction beyond using lagged volume, can be as much as double in terms of expected returns or Sharpe ratio after trading costs. Among popular asset pricing factors, the improvement in after-cost returns ranges from 20 to 100 bps above using a moving average of lagged volume to predict future volume. Refining the prediction methods and deepening the prediction models could add substantially to these improvements.

Going a step further, modeling and forecasting other variables that predict trading costs in addition to volume, such volatility, spreads, and other microstructure variables, in a richer trading cost model could yield even larger benefits. Our aim is simply to showcase the power of forecasting non-return variables for asset pricing and portfolio choice, illustrating the benefits with a single variable and simple model. Our framework illustrates the importance of constructing volume prediction models and we hope sparks future studies to integrate market microstructure frictions into asset pricing forecasting tools (Goldstein, Spatt, and Ye, 2021). More generally, the idea of forecasting non-return variables, and translating them into economic inputs, may help us understand their relevance and impact on asset prices and investment decisions.

The rest of the paper is organized as follows. Section 2 covers some preliminaries of the analysis: a motivation, literature review, and description of our data. Section 3 examines volume prediction from a statistical out-of-sample perspective. Section 4 presents a theoretical portfolio framework for quantifying the economic value of volume prediction. Section 5 discusses the empirical results of predicting volume through the lens of our framework using a variety of machine learning methods. Section 6 applies these insights and methods to trading experiments that characterize the net-of-cost performance improvement of simulated real-world portfolios. Section 7 discusses further steps our framework could take. Section 8 concludes.

2 Preliminaries: Motivation, Literature Review, and Data

We motivate interest in predicting volume, review the literature, and describe the data.

2.1 Motivation

Real-world portfolios face execution costs, which are critical to realizing investment performance. However, empirical challenges to modeling trading costs and finding available and generalizable data limits the attention it has received. One of the main challenges in modeling and forecasting trading costs is that the largest component of these costs for a large investor is price impact, which depends on the size of the trade, the amount traded by other traders (in the same and in opposite directions), and the identity of the trader. Different traders face different price impact. These features frustrate a generic portfolio solution that incorporates trading costs.

Following Kyle (1985) and subsequent empirical work (Frazzini, Israel, and Moskowitz, 2018), price impact depends on the trader's participation rate, defined as the dollar amount traded by investor n in security i relative to total dollar volume in stock i (the amount traded in the market by everybody in security i) at the same time t,

$$ParticipationRate_{n,i,t} = \frac{\$Traded_{n,i,t}}{\$Volume_{i,t}}.$$

Price impact is an increasing function of participation rate (modeled linearly in Kyle, 1985 and empirically verified in Frazzini, Israel, and Moskowitz, 2018), whose elasticity varies by investor n. The numerator is also endogenous to expected price impact, which itself is a function of the participation rate. The circular nature of trade size, and its variation across investors, makes modeling trading costs particularly challenging. However, the denominator of the participation rate is independent of n and exogenous to the trader's desired trade size (assuming a single investor's trade is too small to materially affect total dollar volume). Thus, trading volume is exogenous to each trader and is universal to all investors.

From an empirical standpoint, total dollar volume is easier to forecast because high frequency data on total volume is readily available, while data on individual traders is not. For simplicity, and to illustrate how we can convert a non-return characteristic such as volume into an economic input, we model expected trading costs solely using forecasted total dollar volume for a stock. This overly simple model is only a partial solution to the trading cost problem, but it is a general one, and one that allows us to showcase the economic value of predicting trading volume, abstracting from all of the usual complications and challenges of predicting trading costs. We will find significant after-cost investment improvements just from volume forecasts alone. A more sophisticated trading cost model may well yield even larger performance benefits.

2.2 Literature Review

We are not the first to examine trading volume in an asset pricing context, though our examination is unique in several respects.

Gallant, Rossi, and Tauchen (1992); Brock and LeBaron (1996); Darrat, Rahman, and Zhong (2003) study volume dynamics and predictability, focusing on lead-lag relationships between volume and return volatility for the S&P 500 stock index and individual stocks. Chordia, Huh, and Subrahmanyam (2007) and Chordia, Roll, and Subrahmanyam (2011) examine the determinants of trading volume activity, using proxies for firms' visibility, uncertainty, and dispersion of opinion. Lo and Wang (2000) argue that trading volume has received far less attention in asset pricing research compared to prices and returns, and advocate for greater research attention to volume. Two and a half decades later, volume research still receives relatively little attention.

We forecast volume using machine learning techniques and a rich set of predictors.[8] More importantly, we translate volume prediction into an economic input for the portfolio problem and show that this further improves the value of prediction and quantifies its impact. We follow the literature that shows trading volume is related to liquidity and price impact. Market microstructure theories posit that trading costs arise to compensate liquidity providers for adverse selection risk (Kyle, 1985; Glosten and Milgrom, 1985), inventory risk (Stoll, 1978; Ho and Stoll, 1983), and the uncertainty of offsetting order flow arrivals while holding inventory (Grossman and Miller, 1988). Trading activity, or volume, is associated with lower asymmetric information, resulting in higher liquidity and lower trading costs (Kyle, 1985). Empirically, Campbell, Grossman, and Wang (1993) link volume to price impact. Amihud (2002) constructs a measure of illiquidity as the ratio of absolute daily return to daily trading volume and examines its affect on asset prices.[9]

Our work also contributes to research on transaction costs and portfolio implementation. Korajczyk and Sadka (2004); Frazzini, Israel, and Moskowitz (2012); DeMiguel et al. (2020); Chen and Velikov (2023); Detzel, Novy-Marx, and Velikov (2023); Avramov, Cheng, and Metzker (2023); Azevedo, Hoegner, and Velikov (2023); Jensen et al. (2024) examine the performance of anomaly portfolios and (or) machine learning-driven portfolios after accounting for transaction costs. The main difference between these papers and ours is that all of these previous papers take transaction costs as given and known to the investor. In our setting, trading costs are unknown and must be forecasted and the error in those forecasts plays an important role in portfolio optimization, something previous studies have ignored.

[8] Kaastra and Boyd (1995) is an early precursor that applies neural networks as a prototypical artificial intelligence method to forecast futures volume for the Winnipeg Commodity Exchange (WCE).

[9] Additionally, Benston and Hagerman (1974) draw the connection between volume and liquidity in OTC markets.

To better understand our contribution and how it differs from the literature, consider the most recent paper by Jensen et al. (2024), who evaluate the performance of ML models (for return prediction and for portfolio choice) given assumed (and known) transaction costs. They take transaction costs as given and ask are ML models that seek to predict returns still useful after adjusting for known transactions costs and can ML models be trained to optimize in light of these given costs.[10] Their objective is about after-cost evaluation of signals and models for given trading costs, which is very different from our objective, which is to evaluate the economic content of non-return predictors, in this case trading volume. The fact that we use an expected trading cost model to give trading volume forecasts economic content connects these papers, but fundamentally our aim is different.

Our paper studies a related problem, but comes at it from the opposite direction. We emphasize that transaction costs are *not* given or known when forming the portfolio, and that the investor must make her allocation decisions based on *expected* transactions costs, which are estimated with error. We ask what is the economic value of having more accurate trading cost forecasts using a portfolio framework? Or, conversely what is the cost of trading cost forecast error? By translating volume into expected trading costs, we summarize the value of volume forecast improvement in terms of portfolio economics.[11] This translation allows us to evaluate forecast improvements in terms of meaningful economic consequences rather than statistical metrics such as R^2 or MSE. In addition, and interestingly, we find that incorporating that economic objective into the forecasting problem directly yields significantly better OOS results, offering insight into ways to improve forecasting models, including from machine learning, by merging economic modeling with computer science techniques.[12]

[10] This work follows Brandt, Santa-Clara, and Valkanov (2009) who directly model portfolio weights and optimize after-cost portfolio performance. Simon, Weibels, and Zimmermann (2025) upgrade their framework to deep neural networks.

[11] Related to this idea, Białkowski, Darolles, and Le Fol (2008) model intraday volume dynamics to improve daily VWAP execution. Balduzzi and Lynch (1999) and Çetin, Jarrow, and Protter (2004) model transaction costs in portfolio settings and study the economic value from various transaction cost models.

[12] From a methodology standpoint, our application of transfer learning is closely related to Chen et al. (2023). We present an "economic learning" method that fine-tunes statistical forecasts on an economic model-implied objective in order to optimize economic value. Chen et al. (2023) applies transfer learning from a source domain of simulated data to "teach economics" to the model. In our case, the transfer is not between different datasets, but between different objective functions. Other recent machine learning applications in portfolio optimization also use the portfolio outcome as the direct objective function, for example Chen, Pelger, and Zhu (2023), Cong et al. (2021), following the

2.3 Data

We compile a data panel of daily stock-level dollar trading volume ($\widetilde{V}_{i,t}$) and 175 predictors ($X_{i,t}$). The unit of observation is stock-day (i, t). We adopt the convention that $X_{i,t}$ is observed by day $t-1$, whereas the associated prediction target $\widetilde{V}_{i,t}$ is observed until the end of day t. We use a tilde to denote a random variable conditional on the information prior to day t.

The sample period is 2018 to 2022 (1,258 days). The cross-section covers around 4,700 stocks, with an average of 3,500 stocks per day (4.4 million observations). We split the data into a 3-year training sample and a 2-year testing sample. All models are trained once in the training sample and evaluated out of sample. We avoid re-sampling methods such as cross-validation and rolling-window re-estimations.

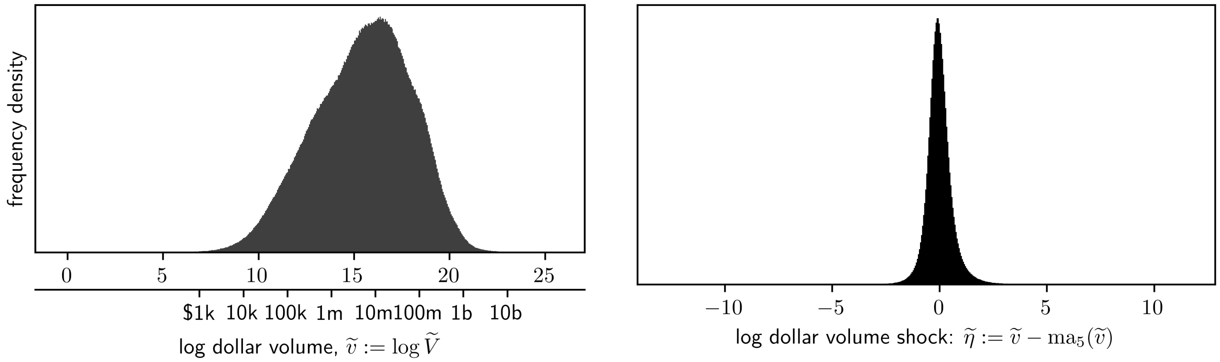
Our analysis focuses on predicting out-of-sample trading volume. Reasonable in-sample fit, often evaluated in the literature (Chordia, Huh, and Subrahmanyam, 2007; Chordia, Roll, and Subrahmanyam, 2011), does not often lead to good out-of-sample (OOS) performance, which is of primary interest to evaluate the robustness and economic impact of volume predictability.

While it is common in the stock return predictability literature to use large sets of variables (e.g., "factor zoo") to identify the best predictors, it is less common to use large data sets to predict market microstructure variables such as trading volume (Easley et al., 2020). We show that using big data improves the precision of volume forecasts significantly.

The main variable we aim to predict is dollar trading volume, which we measure as the natural logarithm of end-of-day transacted total dollar trading volume for each stock. This variable is highly persistent. We focus on predicting innovations in trading volume as well, which we show have significant impact on trading costs. When volume is suddenly much lower than expected, an investor will "overpay" in transaction costs. If volume is higher than expected, then there is more liquidity, and an investor incurs opportunity costs from not trading aggressively enough. We examine the predictive content of several sets of predictors, including technical, fundamental, and event-based variables, which we describe below.

seminal paper of Brandt, Santa-Clara, and Valkanov (2009).

Figure 1: Distributions of daily stock-level log dollar volume ($\widetilde{v}$) and its shock ($\widetilde{\eta}$), panel pooled



Histograms of $\widetilde{v}_{i,t}$ and $\widetilde{\eta}_{i,t}$ in the full sample of around 4,400,000 stock-day observations. The second horizontal axis in the left panel is dollar volume $\widetilde{V}_{i,t}$ in the log scale. Log dollar volume shock $\widetilde{\eta}$ is daily log dollar volume minus the moving average in the past five days: $\widetilde{\eta}_{i,t} := \widetilde{v}_{i,t} - \frac{1}{5}\left(\widetilde{v}_{i,t-1} + \cdots + \widetilde{v}_{i,t-5}\right)$.

2.4 Prediction objects: daily stock trading volume

Daily dollar trading volume $\widetilde{V}_{i,t}$ ranges widely (from thousands to billions of dollars) across stocks and is highly skewed. We take the log of dollar volume, $\widetilde{v} = \log \widetilde{V}$, whose distribution, shown in Figure 1, is relatively well-behaved, being close to normal and symmetrical.

Log dollar volume shows strong temporal persistence and can be easily predicted by lagged moving averages of various frequencies. The five-day moving average predicts log dollar volume with an R^2 of 93.68%, higher than the one-day lag (92.53%), moving average of 22 days (92.60%), or 252 days (86.12%).

We focus on predicting the log dollar volume *shock* defined as daily log dollar volume minus the moving average over the previous five days, $\widetilde{\eta}_{i,t} := \widetilde{v}_{i,t} - \frac{1}{5}\left(\widetilde{v}_{i,t-1} + \cdots + \widetilde{v}_{i,t-5}\right) := \widetilde{v}_{i,t} - [\mathrm{ma}_5]_{i,t}$. Predicting the shock is equivalent to predicting the level as one can always add the baseline back to the shock for the level forecast. Taking out the baseline is comparable to predicting asset returns (change in log price) instead of price levels. We select the five-day frequency due to its highest R^2. Figure 1 (right panel) shows the pooled distribution of $\widetilde{\eta}$, which is relatively symmetric, centered around zero, and has long tails.

2.5 Predictors

We use a total of 175 predictors from various sources, including technical signals, firm fundamentals, and market and corporate events. We show that the virtue of complexity approach in return prediction (Kelly, Malamud, and Zhou, 2024) is also useful for volume prediction. Each subset of variables provides incremental improvement to predicting volume, while using all variables has the greatest out of sample predictability. The sets of predictors that are cumulatively added to the prediction model are:

1. Technical signals ("tech"): lagged moving averages of returns and log dollar volume over the past 1, 5, 22, and 252 days. (8 predictors.)
2. A small set of commonly used fundamental firm characteristics ("fund-1"): market equity, standardized earnings surprise, book leverage, book-to-market equity, Dimson beta, and firm age. (6 predictors.)[13]
3. The remaining firm characteristics from the JKP dataset ("fund-2"), which are merged and transformed in the same way as fund-1 variables. (147 predictors.)
4. Calendar dates with large effects on trading volume ("calendar"). We hard code four binary features based on the dates of the following four types of events.
    - Early closing days for the exchanges (July 3rd, Black Friday, Christmas Eve, and New Year's Eve).
    - Triple witching days (four times a year when the index futures contract expires at the same time as the index option contract and single stock options).
    - Double witching days (eight times a year when two of the three events above, the single stock options and index options expiration, coincide).
    - Russell index re-balancing (once a year, the fourth Friday in June).[14]

[13] These predictors are from the JKP dataset (Jensen, Kelly, and Pedersen, 2022). We forward fill the monthly firm characteristics in time when merging to the daily panel. Hence, the characteristics are still always ex-ante available. On each day, we rank standardize in the cross-section each characteristic to a uniform distribution from -1 to +1.

[14] Triple witching happens on the third Friday in March, June, September, and December. Double witching is on the third Friday in the other eight months. Russell index re-balancing is on the fourth Friday of June when the Russell 1000, Russell 2000, Russell 3000, and other Russell indexes are reconstituted, and has "often been one of the highest-volume trading days of the year" for the exchange, due to indexes tracking funds adjusting their holdings to reflect the updates. Rerferenes: https://www.nasdaq.com/articles/the-powerful-impact-of-triple-witching-2021-06-10., and https://www.nasdaq.com/articles/2023-russell-rebalancing:-what-you-need-to-know.

Early closing days have substantially less trading volume, while the other three are associated with positive spikes in trading volume.

5. Earnings release schedule ("earnings"): We construct 10 categorical dummy variables (one-hot encoding) indicating whether the firm has an upcoming earnings release or just had one in the past few days. We first construct the number of days until the next known scheduled earnings release event. For example, a value of zero implies the current day is previously known to have a scheduled release. A negative value means there are no known scheduled events in the future and indicates how many days since the last event. We convert this number into 10 dummy variables of categorical bins: $\leq -4, -3, -2, -1, 0, 1, 2, 3, 4, \geq 5$. The data source is the Capital IQ Key Developments dataset. (10 predictors.)

This list of variables for predicting volume is not exhaustive. Other variables that could add predictive power are microstructure variables, intraday observations, and lead-lag relations across stocks in terms of trading (e.g., large to small stocks, within industry, etc.). As stated previously, we do not attempt to provide the best volume prediction model. Rather, we translate the volume prediction problem into an economic problem whose objective is after-cost portolio performance. Using our framework, future work can add further predictors for volume that may provide even larger economic benefits than we show here. However, our framework provides a way to assess those predictive contributions in economic terms.

3 Volume prediction from a statistical perspective

We start with a statistical prediction of daily trading volume using various subsets of predictors and a variety of methods, including machine learning techniques.

3.1 Prediction methods

We run predictive regressions of $\widetilde{\eta}$ (changes in daily dollar volume) on a set of predictors, X, in the training sample panel to estimate the models.[15] We compare linear models (ols), with

[15] As explained above, predicting $\widetilde{\eta}$ or $\widetilde{v}$ are essentially the same: predicting $\widetilde{\eta}$ as $nn(X)$ is just predicting $\widetilde{v}$ as $nn(X) + \text{ma}_5$, where ma_5 is one of the predictors. From the machine learning perspective, this is implementing a

neural networks (nn) that allow for non-linear transformations and complex interactions, as well as recurrent neural networks (rnn) that, in addition, allow for state variables to incorporate time series dynamics. The simplest baseline is predicting $\widehat{v}_{i,t} = [\text{ma}_5]_{i,t}$, or in other words, $\widehat{\eta}_{i,t} = 0$. (The "hat" denotes predicted values.) Linear regression is also a simple benchmark comparison.

The neural network implementation is kept simple, standard, and fixed throughout the paper in order to facilitate transparency. The network architecture has three fully-connected hidden layers of 32, 16, and 8 ReLU nodes, respectively, and one linear output node. The size of the input layer is the number of predictors supplied.

Recurrent neural network architecture is particularly appealing for this application as it is designed to capture time-series dynamics. An rnn is analogous to state space models like GARCH, in which the forecast $\widehat{\eta}_{i,t}$ is not only a (non-linear) function of the concurrent predictors $X_{i,t}$, but also of a set of state variables that is the output of the network applied to the previous data point $\{i, t-1\}$. That is, $(\widehat{\eta}_{i,t}, state_{i,t}) = rnn(X_{i,t}, state_{i,t-1})$, where *rnn* represents the neural network function and *state* are the state variables. The *recurrent* neural network processes data sequentially, and recursively passes the state variables to the next time period. Essentially, rnn extracts predictive information from concurrent and lagged predictors $X_{i,t}, X_{i,t-1}, X_{i,t-2}, \ldots$, in contrast to a nn that uses only the concurrent predictors but nothing from the past. Although $X_{i,t}$ contains moving averages of $\widetilde{v}_{i,t-1}, \widetilde{v}_{i,t-2}, \widetilde{v}_{i,t-3} \ldots$, for example, the way such lagged information enters the model without an rnn architecture is highly restrictive. With rnn, the model can "learn" flexible dynamics, where time-series dependencies are parameterized by trainable network weights. We implement the rnn with the popular and standard lstm (long short-term memory) architecture. The number of layers and neurons are kept the same as nn, but the total number of parameters increases by four times (due to the flow of lagged information).[16]

Appendix A.1 contains other details on implementing the machine learning methods, including the optimizer, training scheme, and infrastructure. We do not tune or optimize the hyper-

simple residual connection (ResNet, He et al. 2016) as illustrated in Figure 4.

[16] Specifically, the bottom hidden layer in the aforementioned 3-layer network is upgraded to an lstm layer with 32 hidden states and cell states, with the rest of the two layers unchanged. Lstm is a standard and popular type of rnn with four specific internal mechanisms, or gates, that control the flow of information from both the short- and long-term past (Hochreiter and Schmidhuber, 1997). See Kelly and Xiu (2023) for a general reference and Appendix A.1 for our specifications.

Table 1: Prediction accuracy

cumulatively adding predictor sets total number of predictors	tech 8	fund-1 14	fund-2 161	calendar 165	earnings 175
A: R^2 relative to $\widetilde{\eta}$ ($\widetilde{v} - \text{ma}_5$)					
ma_5	0				
ols	12.09	12.26	12.27	14.85	15.99
nn	14.31	14.90	14.42	17.13	18.45
rnn	15.80	16.25	15.47	18.12	19.86
B: number of parameters					
ma_5	0				
ols	9	15	162	166	176
nn	961	1,153	5,857	5,985	6,305
rnn	6,049	6,817	25,633	26,145	27,425

Each row represents a prediction model, and each column cumulatively adds to the set of predictors. The R^2 is calculated with ma_5 as the benchmark: $R^2 = 1 - \text{MSE}/\text{avg}(\widetilde{v} - \text{ma}_5)^2$, where $\text{MSE} := \text{avg}(\widetilde{v} - \widehat{v})^2 = \text{avg}(\widetilde{\eta} - \widehat{\eta})^2$ and $\text{avg} := \frac{1}{|\text{OOS}|} \sum_{i,t \in \text{OOS}}$ is the OOS average.[18] Each reported R^2 value is the average across five independent runs initialized with different random seeds to ensure the results' robustness and reproducibility. Panel B reports the number of parameters, for which rnn is about four times of nn due to the four gates in lstm, see exact formulas in Appendix A.1.

parameters, the architecture, or the training scheme to improve the results.

3.2 Prediction results

Table 1 reports the OOS prediction accuracy of each method. We cumulatively add sets of predictors in the columns from left to right to highlight the prediction improvement from using larger data sets. The rows correspond to different prediction methods. Panel A reports OOS prediction accuracy in terms of the R^2, where the benchmark is the five-day moving average of volume.[17] Panel B reports the number of parameters estimated in each configuration.

Volume changes are highly predictable. The most sophisticated model using all predictors can predict nearly 20% of future variation in daily trading volume changes. In comparison, daily stock returns are hardly predictable with barely positive OOS R^2, even with state of the art models and

[17] In addition to controlling for the trailing mean in the prediction target, Appendix B.2 considers an alternative predictive specification that standardizes $\eta_{i,t}$ by a rolling standard deviation. The rationale is that the standardized volume shock may be more conducive to forecasting. We find the predictability remains largely similar but somewhat underperforms the benchmark result reported in the main text.

[18] Appendix Table B.2 recast the R^2's in Panel in terms of the explained percentage of the total variation of $\widetilde{v}$.

predictors, and monthly returns are only slightly predictable with low single digit R^2's (Gu, Kelly, and Xiu, 2020).

Adding more predictors improves accuracy in general. All 175 predictors can increase the R^2 by more than three percentage points compared to just the eight technical signals. The exception is that adding the large set of fundamental signals (fund-2) makes the methods (even ML methods) perform a little worse. This may be due to overfitting when the number of features increases and because we do not use regularization techniques to try to mitigate overfitting. The predictors associated with expected returns do not necessarily work for predicting volume. Market-wide calendar events are quite effective in capturing volume changes, however. Scheduled earning announcements also add a sizable gain in prediction accuracy.

The results show that machine learning is useful for prediction, and that complexity has its virtue in the context of predicting volume. The prediction accuracy of the rnn is better than the nn, which in turn is better than ols, uniformly across each configuration of included predictors. Even simple ols models can deliver double-digit R^2, especially when using the largest set of predictors.[19] Panel B shows the improved prediction accuracy is achieved through a significant increase in the number of parameters, a measure of model complexity. Appendix A.1 shows the computational costs of the complex models are higher but manageable.

Appendix B.3 considers volume predictability in different firm size groups and reports that larger firms have higher prediction accuracy than smaller firms, while the overall patterns of predictability across methods are robust in each size sub-sample. The R^2's evaluated among the mega firms are roughly twice those of the nano firms. Smaller firms have a greater magnitude of unexpected trading volume shocks that are harder to predict. This result makes sense since small firms are volatile and have low trading volume, hence unexpected events that give rise to volume spikes are more likely for these firms. This finding indicates that in addition to small firms being less liquid on average, their liquidity is also less predictable and more volatile. In other words, their trading costs are less predictable. We examine whether firms of different size groups should be modeled differently by implementing a mixture of experts (moe) method, which is shown to be beneficial

[19] We experimented with regularization on the linear model (lasso and ridge regressions), and did not find significant improvements.

for the linear model but not for the neural networks.

In the next section, we form an economic objective to assess the value of volume prediction.

4 Trading Volume alpha: the economic value of volume forecasting

To quantify the economic value of predicting volume, we set up a portfolio problem that features a tradeoff between tracking a target portfolio versus minimizing trading costs.

Portfolio optimization targets the ex post performance (e.g. Sharpe ratio) of the portfolio profit and loss (P&L). In a general and comprehensive form, P&L can be expressed as

$$\widetilde{\text{P\&L}}_t = \sum_i x_{i,t}\widetilde{r}_{i,t} - \sum_i TradingCost(x_{i,t}, x^0_{i,t}, \widetilde{\Omega}_{i,t}) + Ar_{\text{f},t}, \tag{1}$$

where $\widetilde{r}_{i,t}$ is the return of stock i on day t in excess of the risk-free rate $r_{\text{f},t}$, A is the portfolio size (AUM), and $\widetilde{\Omega}_{i,t}$ is a vector of attributes, whose values are unknown at the time of trading, that are related to the trade itself, the characteristics of the asset traded, and the market conditions under which the asset will be traded. Importantly, the portfolio manager chooses the dollar positions $x_{i,t}$ before trading, asset, and market conditions (represented by $\widetilde{\Omega}_{i,t}$) are realized, given the starting positions $x^0_{i,t}$ inherited from the previous day as well as the available information set $\{X_{i,t}\}$, for asset i at time t.[20]

Much of the asset pricing literature focuses on forecasting $\widetilde{r}_{i,t}$, with all of the documented challenges of doing so. Here, we shift the focus to "trading costs" as a potential new avenue for building more efficient portfolios, which has its own set of (new) challenges. To simplify matters and as a starting point to quantify the economic benefits of predicting non-return variables, we examine trading volume as the key element in $\widetilde{\Omega}_{i,t}$ that affects trading costs. Of course, there are many attributes that affect trading costs, but our goal here is to focus singularly on trading volume to illustrate how we can translate a non-return prediction problem into portfolio economics.

[20] When implementing the portfolio, the starting position is inherit from the position on $t-1$ and mechanically adjusted by the raw return: $x^0_{i,t} := x_{i,t-1}\left(1 + r_{\text{f},t-1} + \widetilde{r}_{i,t-t}\right)$.

In addition, we fix and set aside the return prediction problem to focus on the improvement afforded by the volume prediction problem. The incentive to trade is modeled with an objective that penalizes the tracking error toward a target portfolio. The tracking target (potentially informed by return forecasting signals) is taken as given since the target itself plays a tangential role in the core tradeoff analysis. In other words, pre-cost or gross expected returns are exogenous and taken as given. The key choice is whether to trade aggressively toward the target or passively to avoid trading costs. The optimal balancing point depends on the volume forecast and the trading costs associated with it, and is general to any specific tracking target. We evaluate whether more accurate volume forecasts translate to better trading execution.

We solve this problem generally, and then apply its solutions to a range of trading experiments with pre-specified targets to evaluate the economic benefit achieved in different tracking tasks, such as implementing high versus low turnover portfolios and other quantitative trading strategies (Section 6). The evaluation uses the original P&L accounting stated above.

4.1 Tracking error optimization and its portfolio microfoundation

The tracking error objective can be modeled with a simple mean-variance portfolio optimization framework. In this framework, the tracking target is given, coming from return forecasts that are outside of this framework. Alternatively, the objective can be from a desire to track a benchmark index.

Suppose the portfolio manager maximizes a mean-variance certainty equivalence of the portfolio P&L, adjusted for trading costs:

$$\sum_i x_{i,t} m_{i,t} - \frac{\gamma}{2A} \sum_{i,j} x_{i,t} x_{j,t} \sigma^2_{ij,t} - \sum_i TradingCost_{i,t} + A r_{\mathrm{f},t}, \tag{2}$$

where the risk aversion coefficient γ is explicitly adjusted by the AUM A.[21] To improve the investment outcome, much empirical work has been devoted to modeling and estimating returns

[21] The risk aversion coefficient is explicitly adjusted by A such that the before-cost Markowitz optimal dollar position, x^*, scales up with A. Eq. 1 assumes a simple situation where the AUM is constant assuming immediate P&L payout.

moments (mean $m_{i,t}$ and the variance-covariances $\sigma^2_{ij,t}$). Instead, we assume a simple form for the return moments and take them as given in order to illustrate the economic value solely of the trading cost term. Assuming $\mathbb{V}\text{ar}_{t-1}\widetilde{r}_{i,t} = \sigma^2$, with zero covariances, the objective function is

$$-\frac{\gamma\sigma^2}{2A}\sum_i \left(x_{i,t} - \frac{A}{\gamma\sigma^2}m_{i,t}\right)^2 - \sum_i TradingCost_{i,t} + \left(Ar_{\text{f},t} + \frac{A}{2\gamma\sigma^2}\sum_i m^2_{i,t}\right). \tag{3}$$

The task is to balance the tradeoff between the first term, which is the tracking error penalty as the result of the mean-variance optimization, and the second term, the transaction cost, to be detailed below. The third term can be ignored in the optimization since it is irrelevant to the x choices.

The tracking error penalty (first term) is quadratic,

$$TrackingError_{i,t} := \frac{1}{2}\mu(x_{i,t} - x^*_{i,t})^2, \tag{4}$$

where the target x^* is the before-cost mean-variance efficient portfolio position, which increases in the asset's return expectations as well as the total portfolio size A. In implementation, the trading target is formed in a separate process without immediate trading cost consideration. We analyze a general strategy that optimizes the trading rate toward the target based on volume predictions. Parameter $\mu := \frac{\gamma\sigma^2}{A}$ controls the strength of the tracking error penalty. In later empirical analyses, we do not calibrate μ from risk coefficients but instead treat it as a hyperparameter and tune the optimal μ under various AUM levels according to investment performance. Still, the qualitative relationship is preserved – a larger investor penalizes tracking error (measured in dollars) less, meaning they trade less aggressively toward the target in general.

Trading costs are modeled as

$$TradingCost_{i,t} := \frac{1}{2}\widetilde{\lambda}_{i,t}(x_{i,t} - x^0_{i,t})^2, \tag{5}$$

where x^0 is the starting position. We specify $\widetilde{\lambda}$ as a function of volume: $\widetilde{\lambda} = 0.2/\widetilde{V} = 0.2\exp(-\widetilde{v})$, following Frazzini, Israel, and Moskowitz (2018). Underlying the quadratic functional form, it is assumed that the price impact is linear in the trade's size relative to the volume of the day (a.k.a.

participation rate): $PriceImpact = 0.1\frac{x-x^0}{\widetilde{V}}$ (Kyle, 1985). For example, buying (or selling) 10% of the daily volume would move the price by 1% (or −1%). And the trading cost is the price impact multiplied by the dollar trade size: $TradingCost = PriceImpact \cdot (x - x^0)$.[22]

In summary, the aggregate tracking error optimization problem is,

$$\min_{\{x_{i,t}\}} \sum_{i,t} \left(TrackingError_{i,t} + TradingCost_{i,t}\right). \tag{6}$$

Central to our paper, volume $\widetilde{v}_{i,t}$ or price impact are not known when choosing $x_{i,t}$ (emphasized by the tilde). One must predict them based on available conditioning information represented by predictors $X_{i,t}$. Taking $x^0_{i,t}$ and $x^*_{i,t}$ as given, the problem becomes $\{i,t\}$-separable.[23] Then, the problem is (in the panel population)

$$\min_{x\in\sigma(\mathcal{X})} \mathbb{E}\left[\frac{1}{2}\widetilde{\lambda}(x-x^0)^2 + \frac{1}{2}\mu(x-x^*)^2\right].^{24}$$

4.2 Normalized tracking error and trading rate (z)

To implement this problem empirically, we first normalize the problem by the target trade size $x^* - x^0$ so that we analyze the loss for a one-dollar target trade. The problem scales quadratically, for example, an $x^* - x^0 = \$1{,}000$-dollar trade task will incur 10^6 times the loss of a \$1-trade. In detail, let the choice variable be trading rate $z := \frac{x-x^0}{x^*-x^0}$, then the minimization objective becomes

$$\frac{1}{2}\widetilde{\lambda}(x-x^0)^2 + \frac{1}{2}\mu(x^*-x)^2 = \frac{1}{2}\left(x^*-x^0\right)^2\left(\widetilde{\lambda}z^2 + \mu(1-z)^2\right). \tag{7}$$

[22] For simplicity, we assume away cross-impact on $\widetilde{\lambda}$ from related stocks as well as other determinants of $\widetilde{\lambda}$. Also, other functional forms for $\widetilde{\lambda}$, such as the quadratic form often shown empirically, $\widetilde{\lambda} = \frac{0.2}{\sqrt{\widetilde{V}}} = 0.2\exp(-\frac{1}{2}\widetilde{v})$, (Frazzini, Israel, and Moskowitz, 2018) can also be used. In this case, all the analyses carries through but $\widetilde{v}$ will be twice as large. We stick with the linear specification consistent with theory (Kyle, 1985) and empirical evidence on the unexpected component of trading volume (see Frazzini, Israel, and Moskowitz 2018).

[23] Ideally, $x^0_{i,t}$ should not be taken as given, as it is affected by the choice on the previous day, but we are not considering the dynamics of the problem here. By taking $x^0_{i,t}$ and $x^*_{i,t}$ as given, the problem is $\{i,t\}$-separable and easier to solve. In the trading experiments (Section 6), however, we consider the dynamics by evaluating the recursively traded portfolios.

[24] Here "$x \in \sigma(\mathcal{X})$" restricts x as a random variable that depends only on predictive information available at the time of the choice, with $\mathcal{X}_{i,t} := [X_{i,t}, X_{i,t-1}, X_{i,t-2}, \ldots]$. This unconditional expectation minimization is equivalent to solving $\min_{x\in\mathbb{R}} \mathbb{E}\left[\frac{1}{2}\widetilde{\lambda}(x-x^0)^2 + \frac{1}{2}\mu(x-x^*)^2 \middle| \mathcal{X}\right]$ for each $\mathcal{X}$ realization.

Since the factor, $\left(x^* - x^0\right)^2$, does not matter for the choice of z, define the economic loss as

$$loss^{\text{econ}}(\widetilde{v}, z; \mu) := \widetilde{\lambda} z^2 + \mu(1-z)^2, \tag{8}$$

and the normalized problem as

$$\min_{z \in \sigma(\mathcal{X})} \mathbb{E}\left[loss^{\text{econ}}(\widetilde{v}, z; \mu)\right], \tag{9}$$

which is the main focus of economic machine learning. Being able to separate x^*, x^0 affords many conveniences. It means the core problem is independent of the target strategy or fund size. It allows us to look at each i, t observation independently in a volume prediction setting. After the prediction task is done, we evaluate the investment performance under various pre-specified target strategies (x^*) with different AUM levels.

4.3 The optimal policy ignoring forecast error

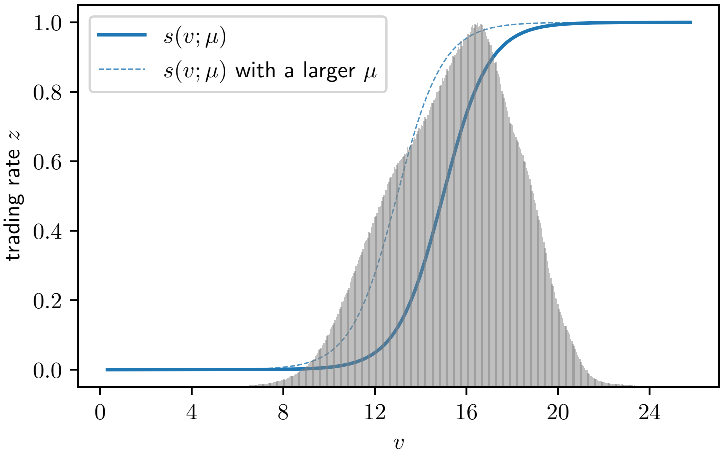
Suppose $\widetilde{v} = v$ and ignoring the inaccuracy in the prediction, the optimal policy is then,

$$s(v; \mu) := \arg\min_z loss^{\text{econ}}(v, z; \mu) = \frac{\mu}{\mu + \lambda} = \frac{1}{1 + \exp\left(-v + \log 0.2 - \log \mu\right)}. \tag{10}$$

We plot this function in Figure 2. It is a sigmoid function with a horizontal offset determined by μ. The optimal trading rate z ranges from 0 to 1 as v increases. Then, the optimal dollar position choice is $x = x^0 + s(v; \mu)(x^* - x^0)$. This is the finding of Gârleanu and Pedersen (2013) that the optimal strategy should "trade partially toward the aim."[25] In their setting, the trading rate is a fixed constant. Here, it is still irrelevant to either the trading target or the starting point (x^*, x^0), but importantly, it depends on the volume prediction, v. Instead of assuming liquidity as a constant known by the agent, the innovation and emphasis of this paper is that *volume prediction (and prediction error) alters the tradeoff between the cost and benefit of trading,*

[25] The other finding in Gârleanu and Pedersen (2013) is that the optimal strategy should also "aim in front of the target." This dynamic effect is abstracted away in our problem since we are considering the $\{i, t\}$-separable optimization. One interpretation is that x^* is already the aim that is in front of the target that implicitly embeds the dynamic effect.

Figure 2: The policy function (s) that maps log volume (v) to trading rate (z)



The solid curve uses a μ value relevant for \$1b AUM. The dashed curve changes to a greater μ (AUM = \$100m), increasing z across the spectrum. The background is the histogram of $\widetilde{v}$ data repeated from Figure 1, to show that a typical $\widetilde{v}$ corresponds to a $\widetilde{z}$ somewhere between 0 and 1 given the chosen μ's.

and hence more accurate predictions lead to more efficient portfolio implementation with a z that varies with forecasted volume. Additionally, the optimal z also does not explicitly depend on the scale of the fund. If the AUM doubles, and both x^* and x^0 double, the optimal z remains the same, while the dollar position choice x doubles. However, a smaller fund will find a larger μ more relevant for their investment performance optimization. In that case, they will trade more aggressively uniformly across v, as illustrated by the upward shift of the dashed curve in Figure 2, although the magnitude of the shift is not constant. For example, at very low (high) trading volume, v, both large and small managers will trade very little (aggressively) implying z close to zero (one). However, over the majority of the distribution of volume, small managers with larger μ will trade more aggressively (higher z), with differences in z being quite significant.

5 Machine learning for the economic value of volume prediction

We provide empirical methods to construct the policy of choosing z given $\mathcal{X}$.

5.1 The statistical and economic tasks of volume prediction

We consider two ways of approaching the portfolio optimization problem. The first conducts a statistical prediction of volume and then plugs the volume forecasts into the optimal trading policy $s(v;\mu)$ to form a trading plan. This indirect approach we call "statistical" learning. The second is an economic learning approach, which instead learns trading rate z as a function of conditioning information directly to minimize the economic loss. We argue directly choosing z is also "predicting volume," because there is a direct mapping from z to v, but with a different optimization objective rather than the least squares loss commonly used in statistical predictions. We term this approach "economic" learning. We show their theoretical differences: the economic loss penalizes inaccuracies in volume forecasting asymmetrically, where overestimating volume is more costly under some regions of the volume distribution. In this case, the model optimized for the economic goal will be more conservative, at the expense of compromising on minimizing squared errors.

- Approach 1, statistical learning:

  Step 1: run machine learning regressions of $\widetilde{v}$ onto $\mathcal{X}$ in the training sample as in Section 3

  $$v^*(\cdot) = \arg\min_{v(\cdot)} \sum_{i,t\in\text{train}} (\widetilde{v}_{i,t} - v(\mathcal{X}_{i,t}))^2. \tag{11}$$

  Step 2: plug the OOS predictions $\widehat{v}_{i,t} := v^*(\mathcal{X}_{i,t})$ into policy equation (10) to trade $\widehat{z}_{i,t} = s(\widehat{v}_{i,t};\mu)$.

- Approach 2, economic learning:

  parameterize z as a neural network, optimize the economic objective in the training sample

  $$z^*(\cdot) = \arg\min_{z(\cdot)} \sum_{i,t\in\text{train}} loss^{\text{econ}}(\widetilde{v}_{i,t}, z(\mathcal{X}_{i,t});\mu) \tag{12}$$

  and trade $\widehat{z}_{i,t} = z^*(\mathcal{X}_{i,t})$ in the testing sample.

The difference between the two approaches boils down to the different loss functions deployed in penalizing volume prediction errors. For example, a trading action $\widehat{z} = z(\mathcal{X})$ implies an underlying volume forecast $\widehat{v} = s^{-1}(\widehat{z};\mu)$, and equivalently the economic loss can be written as a function of the z-implied v instead of z itself: $loss^{\text{econ}}_{\text{vv}}(\widetilde{v}, v; \mu) := loss^{\text{econ}}(\widetilde{v}, s(v; \mu); \mu)$. Hence, the economic learning approach is equivalent to first solving

$$\min_{v(\cdot)} \sum_{i,t \in \text{train}} loss^{\text{econ}}_{\text{vv}}(\widetilde{v}_{i,t}, v(\mathcal{X}_{i,t}); \mu) \tag{13}$$

followed by the $s(\,\cdot\,; \mu)$ transformation, which is also required in the first approach.

Given that the two approaches are only different in the loss functions, we compare $loss^{\text{econ}}_{\text{vv}}$ with the least squares loss function, $loss^{\text{ls}}(\widetilde{v}, v) := \frac{1}{2}(v - \widetilde{v})^2$, used in statistical predictions. To understand how the two approaches behave differently, first note that the two functions are the same for the smallest possible loss being attained, which is when the forecast, v exactly equals the target, $\widetilde{v}$. In empirical experiments, we label the strategy made with perfect foresight “oracle” as the unattainable ideal ($\widehat{v}_{i,t} = \widetilde{v}_{i,t}$), which yields the smallest mean squared error (MSE) and smallest mean economic loss (MEL).

Second, both loss functions monotonically increase as the forecast v deviates away from the true $\widetilde{v}$. Therefore, it is intuitive to think that making forecasts that are close to $\widetilde{v}$ in the least squares sense, will translate to better portfolio implementation as evaluated by the economic loss.[26] However, forecast errors will not guarantee this outcome because of the differences between the two loss functions. From a theoretical perspective, it is well known that the conditional expectation, $\mathbb{E}[\widetilde{v}|\mathcal{X}]$, is the minimizer of the problem $\min_{v \in \sigma(\mathcal{X})} \mathbb{E}\left[loss^{\text{ls}}(\widetilde{v}, v)\right]$, so that the statistical learning method recovers the conditional expectation with the neural network tools. However, given a different economic loss function,

Proposition 1. *The least squares minimizer,* $\mathbb{E}[\widetilde{v}|\mathcal{X}] = \arg\min_{v \in \sigma(\mathcal{X})} \mathbb{E}\left[loss^{\text{ls}}(\widetilde{v}, v)\right]$*, does not optimize the economic loss minimization problem* $\min_{v \in \sigma(\mathcal{X})} \mathbb{E}\left[loss^{\text{econ}}_{\text{vv}}(\widetilde{v}, v; \mu)\right]$.

Even with unlimited data, the “perfect” statistical learning would not optimize the economic problem. The theoretical foundation of this claim is that $\mathbb{E}[\widetilde{v}|\mathcal{X}] = \arg\min_{v \in \sigma(\mathcal{X})} \mathbb{E}\left[\phi(\widetilde{v}, v)\right]$ *if and only if* the generic loss function ϕ is in the Bregman class (Banerjee, Guo, and Wang, 2005; Patton, 2020).

[26] The above mentioned properties expressed mathematically are $\arg\min_v loss(v, \widetilde{v}) = \widetilde{v}, \forall \widetilde{v}$; and $loss(v, \widetilde{v})$ is increasing in $|v - \widetilde{v}|, \forall v, \widetilde{v}$. Both $loss^{\text{econ}}_{\text{vv}}$ and $loss^{\text{ls}}$ satisfy these properties.

As is well known, the least squares loss belongs to the Bregman class. However, the economic loss function does not (proofs in Appendix C.3).

We plot and compare the two loss functions in Figure 3. We pick five different true values for $\widetilde{v} = 4, 8, 12, 16, 20$, and respectively plot the loss curves for a range of forecasts $v \in [0, 24]$. The left graph plots the statistical loss relative to v. The dots mark the minimum loss where the forecast $\hat{v}$ $= \tilde{v}$ ("the oracle"). Given the quadratic statistical loss function, forecast errors on either side of $\tilde{v}$ generate symmetric losses. The right graph in Figure 3 plots the economic loss relative to volume forecasts.[27] The distinguishing feature of the economic loss is its asymmetric penalty for over or underestimating volume over various regions of the volume distribution. For low actual volume, overestimating volume is extremely costly as it causes much larger price impact. For very high volume, the opportunity cost of not trading aggressively enough is larger than the cost of trading and hence underestimating volume when volume is high is more costly. Comparing the values of v over the actual distribution of realized trading volume, it is clear that overestimating volume is on average more costly than underestimating it.

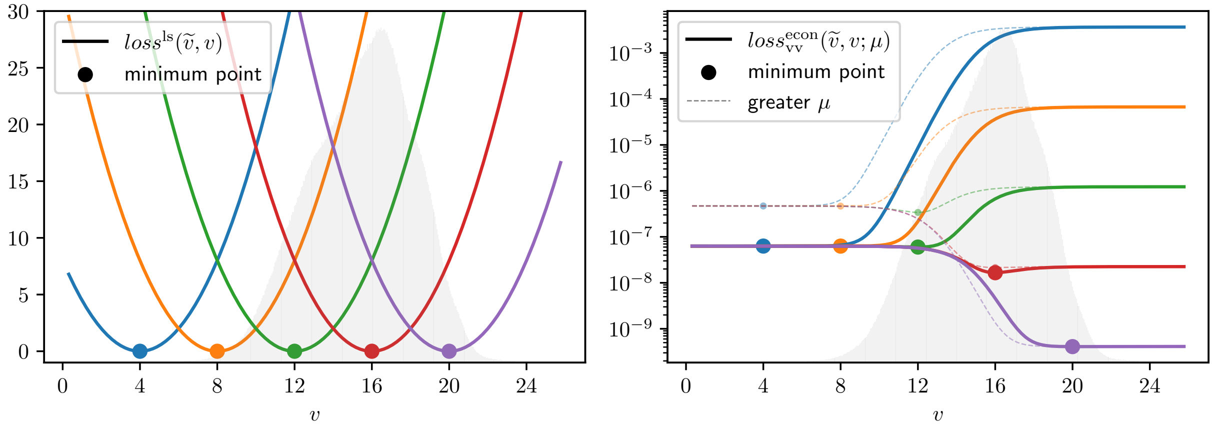
For example, the blue loss curve (actual $\widetilde{v} = 4$) is very high if the forecast is mistakenly large ($v > 12$). The economic intuition for why this particular forecasting error is so costly is that it implies trading aggressively (z close to 1) when actual liquidity is low. Analytically, $\lim_{z \uparrow 1} loss^{\text{econ}}(\widetilde{v}, z; \mu) = \widetilde{\lambda}$, which increases exponentially if $\widetilde{v}$ is small. In contrast, errors in the other direction are much more forgiving (e.g., the purple curve with $\widetilde{v} = 20$).[28] Trading too little when actual liquidity turns out to be ample delivers a loss equal to the opportunity cost of not tracking the target portfolio, which is fixed at μ ($\lim_{z \downarrow 0} loss^{\text{econ}}(\widetilde{v}, z; \mu) = \mu, \forall \widetilde{v}$). Thus, for low levels of true volume, overestimating volume is very costly, but for high levels of volume, the economic cost of volume forecast error is relatively small.

We state this asymmetric property formally in Proposition 4 in Appendix C.4, and provide further analysis. The analytical results rely on the quadratic functional forms assumed in equations

[27] In the right panel, the dips around the minimums (indicated by the dots) are too shallow to be noticeable, though analytically they are indeed the minimums. See Appendix C.4 for more details on the local curvature around the minimum points.

[28] Notice the vertical axis is in log scale, meaning the purple is much more flat compared to blue in terms of the difference of its right and left ends. The same plot in linear scale is in Appendix Figure C.4.

Figure 3: The statistical (least squares) and economic loss functions



The two panels visualize $loss^{\text{ls}}(\widetilde{v}, v)$ and $loss^{\text{econ}}_{\text{vv}}(\widetilde{v}, v; \mu)$, respectively. We pick five different true values $\widetilde{v} = 4, 8, 12, 16, 20$ (in five colors), and respectively plot the loss curves for $v \in [0, 24]$. The dots mark the minimums of the loss curves, attained at $v = \widetilde{v}$. The right panel is in the log scale. The solid and dash curves use the μ values in columns 2 and 3 in Table 2 (corresponding to \$1b and \$100m AUM), respectively.

(4) and (5). However, the qualitative points can carry over to more general settings.

These findings have important implications. Off-the-shelf machine learning tools are not the most suitable for specific portfolio problems because they minimize statistical error rather than economic error. Hence, an altered financial machine learning design that accounts for economic loss can be more effective. The ranking of forecasts evaluated by the two loss functions can be reversed – a set of forecasts with a smaller sample MSE might have a greater economic loss – something we will see empirically in the next section. The asymmetry in the economic loss has important implications for implementing a model optimized for the statistical criteria to the trading task. The model should “learn” to be more careful about the potential of a liquidity dry-up and be conservative in avoiding overestimating volume – compromising statistical accuracy in favor of minimizing economic losses.

Both the statistical and economic learning approaches are implemented with neural networks, given the many benefits of deep learning such as the ability to handle high-dimensional data and non-linear relations. The network architecture are kept the same for both approaches, with the only difference being the loss function used in training. We emphasize this one aspect of the implementation that is particularly relevant for transferring statistical learning results to finance applications.

5.2 Transfer learning via pre-training and fine-tuning

We implement a transfer learning paradigm in which the statistical and economic learning models are trained sequentially, as illustrated in Figure 4.

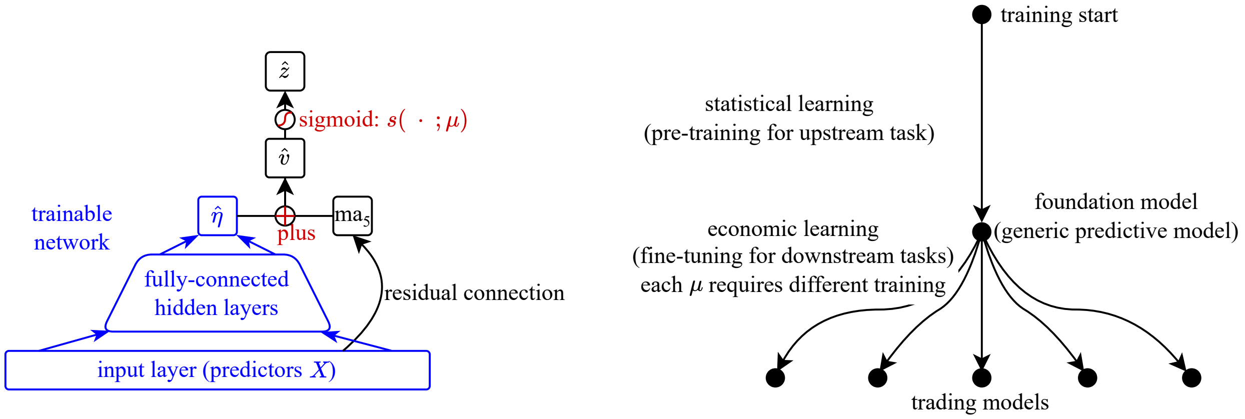
The statistical volume prediction is an upstream task. It learns valuable but generic information on the predictive relationship and serves as the foundation model (center node in Figure 4). It uses off-the-shelf machine learning programs optimized for this typical task. The background knowledge can be transferred to more specific downstream tasks such as portfolio optimization in our case. The finance-motivated tasks benefit from a good "foundation." Fine-tuning the pre-trained foundation model per the economic loss objective further improves the economic performance. The different specific downstream tasks require separate economic training routines. (The tasks are different because the economic loss function is modulated by μ.) The same pre-trained foundation model serves as the common starting point for different downstream fine-tuning.[29]

Pre-training optimizes a nn or rnn from X to $\widehat{\eta}$ (the blue part in Figure 4 left panel) as described in the previous section. Using the pre-trained network as is, flowing the output $\widehat{\eta}$ through additional transformations, "+ma$_5$" and "sigmoid $s(\,\cdot\,;\mu)$" shown in black, the resulting $\widehat{z}$ implements the plug-in step. Taking this foundation model as the initialization, fine-tuning conducts further stochastic gradient descent in the economic loss function evaluated at the same training sample. The resulting fine-tuned network implements the economic learning approach. Only the trainable part (in blue) is updated in fine-tuning. The economic approach only innovates on the loss function, not the network architecture or data. Neural networks tackle the non-linearity not only in the predictive relationship but also in the (marginal) economic loss function. Experiments show fine-tuning requires only a small number of epochs of training to significantly improve OOS economic performance compared to the pre-trained foundation model.[30]

[29] An even lower-level downstream task is to make decisions given target position x^*. We do not directly train for that, but do evaluate the performance in such trading experiments further below.

[30] Alternatively, side-stepping pre-training but directly training for the economic loss from scratch is generally less robust and takes more time (epochs over the sample) to train, probably because the machine learning program is not

Figure 4: Network architecture and transfer learning procedure


$\hat{z}$
sigmoid: $s(\,\cdot\,;\mu)$
$\hat{v}$
trainable network
$\hat{\eta}$
plus
ma_5
fully-connected hidden layers
residual connection
input layer (predictors X)
training start
statistical learning
(pre-training for upstream task)
foundation model
(generic predictive model)
economic learning
(fine-tuning for downstream tasks)
each μ requires different training
trading models

The left illustration shows the network architecture. The blue part "trainable network" is a standard feed-forward neural network, specified with three fully connected hidden layers with 32, 16, and 8 neurons, respectively. The black parts are non-trainable transformations from $\widehat{\eta}$ to log volume prediction $\widehat{v}$ and ultimately trading rate $\widehat{z}$. The recurrent connections in rnn are omitted in this illustration. The right figure illustrates the training procedure for transfer learning. Each dot represents a trained model, i.e., a parameterization of the network. The arrows show economic learning as specific fine-tuning steps based on the common foundation model pre-trained statistically.

Many other finance problems share a similar structure, in which statistical results are transferred into actionable strategies that are applied towards an economically motivated problem. Markowitz portfolio optimization is a classic example. Another example is financial risk management, which relies on volatility forecasts. The transfer learning procedure adopted here provides a unified framework for applying machine learning techniques in these scenarios.[31] The procedure is also similar to how GPT models are "P"re-trained on language representations and applications like ChatGPT are fine-tuned for generating conversational responses or other tasks.

5.3 Economic and statistical prediction results

We present a systematic comparison of how different predictor sets and models perform in terms of both statistical and economic performance. The results show the economic performance is indeed optimized by fine-tuning the training process on the same objective, and that more predictors and

optimized for such a loss. Not to mention that each μ would require a separate training routine as they do not share the common pre-trained baseline.

[31] Some financial machine learning studies, including Jensen et al. (2024), Chen et al. (2023), Cong et al. (2021) and Chen, Pelger, and Zhu (2023) also involve directly training for the economic target.

the network model lead to improved performance.

Table 2 presents results of hypothetically trading \$1 in every stock in our sample every day, using its forecasted daily trading volume. We evaluate in Panel A the OOS mean economic loss (MEL), and Panel B the mean squared error (MSE). For ease of comparison, Panels A$'$ and B$'$ normalize these metrics as a percentage loss reduction such that the "oracle" (which represents the perfect prediction $\widehat{v} = \widetilde{v}$) is at 100% accuracy and the baseline ma_5 at 0%. (The percentage reduction in MSE is then the R^2 predicting $\widetilde{\eta}$ as in Table 1 Panel A.) The four columns correspond to different μ values that modulate the economic loss function, representing four different downstream economic tasks relevant to different AUM magnitudes. As the AUM decreases, the relevant economic objective is parameterized by a greater μ, such that overall trading becomes more aggressive. The average trading rate ranges from trading conservatively near the starting position (13% for AUM = \$10b) to trading almost all the way to the target position (95% for AUM = \$10m).

The rows include the statistical prediction methods described in Section 3 (ma_5, ols, nn, and rnn). Additionally, the economic learning approaches (labeled ".econ") conduct fine-tuning on top of the statistical forecasts to optimize the economic loss function under the four different μ's. The lines spanning the columns indicate that the statistical forecasts do not vary with μ, whereas economic learning generates different (z-implied) volume forecasts as μ varies, because μ varies the economic loss from the trade off of trading cost versus tracking error, based endogenously on the model. The predictor sets include the 8 "tech" signals or "all" of the 175 signals.

Looking first at the statistical performances of the forecasts, model complexity and feature richness help reduce MSE. Configuration rnn_{all} yields the highest OOS $R^2 \approx 20\%$. However, smaller MSEs do not necessarily lead to better economic performance. For example, nn_{tech} accrues greater MELs than ols_{tech}, albeit nn is more accurate in terms of R^2 (comparing Panels A and B). For the low AUM task in particular, the various volume forecasts are even worse than the baseline ma_5, likely due to the need to trade aggressively in this task and the disproportional penalty when aggressive trades are the result of overestimating volume. This result is intuitive given the theoretical analysis, and argues why economic fine-tuning produces better OOS portfolio performance.

Table 2: Economic and statistical performance of different methods

	1	2	3	4	1	2	3	4
μ	1.2e-9	6.3e-8	4.7e-7	9.4e-6	1.2e-9	6.3e-8	4.7e-7	9.4e-6
avg $\widetilde{z}$	0.13	0.57	0.78	0.95	0.13	0.57	0.78	0.95
relevant AUM	\$10b	\$1b	\$100m	\$10m	\$10b	\$1b	\$100m	\$10m
A. Mean economic loss (MEL) ($\times 10^{-8}$)					A′. % reduction in mean economic loss			
ma$_5$	0.1046	3.163	15.41	93.0	0.0	0.0	0.0	0.0
ols$_{tech}$	0.1041	3.011	14.81	93.2	27.9	29.6	11.1	-0.3
nn$_{tech}$	0.1043	3.100	14.97	99.7	19.2	12.3	8.0	-12.8
rnn$_{tech}$	0.1040	2.955	14.37	102.2	33.6	40.8	19.2	-17.7
nn.econ$_{tech}$	0.1041	2.991	12.35	66.8	31.3	33.6	56.5	50.2
rnn.econ$_{tech}$	0.1039	2.855	11.78	64.2	39.7	60.4	67.0	55.3
ols$_{all}$	0.1040	3.024	14.97	94.7	32.4	27.2	8.1	-3.3
nn$_{all}$	0.1040	3.019	15.07	106.5	33.3	28.2	6.2	-25.9
rnn$_{all}$	0.1040	3.012	14.78	109.8	34.8	29.5	11.7	-32.1
nn.econ$_{all}$	0.1039	2.810	11.56	61.9	39.6	69.2	70.9	59.6
rnn.econ$_{all}$	0.1038	2.812	11.60	66.4	43.7	68.8	70.3	51.0
oracle	0.1029	2.653	9.99	40.8	100	100	100	100
B. Mean squared error (MSE)					B′. R^2 (% reduction in MSE)			
ma$_5$	0.437				0.0			
ols$_{tech}$	0.385				12.1			
nn$_{tech}$	0.375				14.3			
rnn$_{tech}$	0.368				15.8			
nn.econ$_{tech}$	0.389	0.449	0.457	0.630	11.2	-2.6	-4.5	-44.1
rnn.econ$_{tech}$	0.392	0.492	0.487	0.481	10.3	-12.4	-11.3	-10.0
ols$_{all}$	0.367				16.0			
nn$_{all}$	0.357				18.4			
rnn$_{all}$	0.350				19.9			
nn.econ$_{all}$	0.394	0.555	0.590	1.979	10.0	-26.8	-34.9	-352.5
rnn.econ$_{all}$	0.377	0.440	0.477	0.785	13.9	-0.6	-9.0	-79.5
oracle	0.00				100			

Panel A: Mean economic loss (MEL) := avg $loss^{\text{econ}}(\widetilde{v}, \widehat{z}; \mu)$; A′: % reduction in mean economic loss := $(\text{MEL}_{\text{ma5}} - \text{MEL}_m)/(\text{MEL}_{\text{ma5}} - \text{MEL}_{\text{oracle}})$; B: MSE := avg $(\widetilde{v} - \widehat{v}_m)^2$; B′: $R^2 := 1 - \text{avg } (\widetilde{v} - \widehat{v}_m)^2/\text{avg } (\widetilde{v} - \widehat{v}_{\text{ma5}})^2 = (\text{MSE}_{\text{ma5}} - \text{MSE}_m)/\text{MSE}_{\text{ma5}}$, for each method m and μ ("avg" is the OOS average over i, t). $\widehat{z}$ varies over each method and μ. For statistical methods, $\widehat{v}$ does not depend on μ, hence the horizontal lines indicate the MSE and R^2 do not depend on μ for these methods. These R^2 numbers repeat those from Table 1 row "$\widetilde{\eta}$" by construction. In the header, the two additional rows help interpret the μ values: avg $\widetilde{z} =$ avg $s(\widetilde{v}; \mu)$ is the average trading rate given true volume; Under each "relevant AUM", the corresponding μ is backed out in portfolio optimization hyperparameter tuning (see Footnote 32 for details on tuning μ).

The economic training methods lead to better economic outcomes. The fine-tuned networks with all the predictors yield the best performance, reaching an OOS economic performance that is about 43%∼70% of the unattainable oracle (perfect forecast) benchmark at various AUM levels. These improved OOS outcomes are not mechanically guaranteed since the fine-tuning is to minimize in-sample loss. The empirical results demonstrate the validity of the economic learning design. Looking at Panel B′, the statistical accuracy retreats after fine-tuning, often to levels even worse than the ma_5 baseline resulting in negative R^2. This means to achieve improvements in economic outcomes, the models compromise MSE, and do so by penalizing errors related to overestimating volume more than underestimating volume according to the economic loss function.

Furthermore, for a smaller AUM or, equivalently, a higher μ value, the economic models tend introduce more statistical bias, as indicated by the increasingly negative R^2 values. This can be explained by the changes in the economic loss functions. With a smaller μ, trading is more intensive in general (s curve in Figure 2 shifts to the left), so the penalty for overestimating low volume is more stringent and takes effect earlier (dashed curves in Figure 3 shift to the left). Thus, for smaller AUM, there is a greater difference between the least squares loss and the economic loss.

6 Investment performance in trading experiments

While the previous section looked at hypothetical normalized \$1-trades, we now apply the analysis to real-world investment portfolios.

6.1 Trading experiment design

We form a set of trades $x_{i,t}$ by applying the various trading strategies detailed in the previous section to dynamically track a set of given target positions $x^*_{i,t}$. The target positions are not optimized on trading cost considerations, but formed in a separate process focused solely on return prediction. We evaluate the outcome of the implemented trades $x_{i,t}$, including the mean return, Sharpe ratio, and turnover, in the OOS period. We examine whether a volume prediction method brings improved investment performance net of trading costs and tracking error considerations.

Various sets of target positions $\{x^*_{i,t}\}$ are exogenously supplied to mimic realistic trading tasks. The first set of experiments mimic a quantitative strategy. We simulate an extremely profitable before-cost trading strategy assuming the agent can, with some probability, forecast the realized direction of stock price changes. We experiment with different AUM levels, in which the dollar positions scale linearly while the trading costs increase quadratically. As a result, the optimal μ, which controls the overall trading aggressiveness, varies. The second set of experiments tracks monthly-rebalanced factor portfolios sorted on firm characteristics from the asset pricing literature as the trading targets. These experiments reveal the economic values of volume prediction across the spectrum of investment styles.

Given the target $\{x^*_{i,t}\}$, the implemented trading outcome $\{x_{i,t}\}$ is constructed following the trading rate strategy, such as $\widehat{z}_{i,t} = \text{rnn.econ}(\mathcal{X}_{i,t}; \mu)$, which is formed in the training sample. In particular, $x_{i,t} = x^0_{i,t} + \widehat{z}_{i,t}(x^*_{i,t} - x^0_{i,t})$, where the starting position is formed recursively as $x^0_{i,t} := x_{i,t-1}\,(1 + r_{\text{f},t-1} + \widetilde{r}_{i,t-t})$, which is mechanically adjusted by the arithmetic raw return accrued on day $t-1$.[32] Note the dynamic effect here: the portfolio choice matters for the starting position on the next day, although the optimization does not explicitly consider it. Additionally, the target amount to trade $x^* - x^0$, varies across i, t, which is another aspect abstracted from in the theoretical analysis.

Given the resulting trades $\{x_{i,t}\}$, the accounting of the investment outcome is standard. The daily dollar P&L is (repeating Eq. 1):

$$\widetilde{\text{P\&L}}_t = \sum_i x_{i,t}\widetilde{r}_{i,t} - \sum_i TradingCost_{i,t} + Ar_{\text{f},t}.$$

Normalizing by A gives the net-of-cost excess return of the implementation portfolio:

$$\widetilde{r}_{\text{implemented},t} := \frac{\widetilde{\text{P\&L}}_t}{A} - r_{\text{f},t} = \sum_i w_{i,t}\widetilde{r}_{i,t} - A\sum_i \frac{\widetilde{\lambda}_{i,t}}{2}\left(w_{i,t} - w^0_{i,t}\right)^2, \tag{14}$$

where $w_{i,t} := \frac{x_{i,t}}{A}$ and $w^0_{i,t} := \frac{x^0_{i,t}}{A}$ are portfolio weights as a ratio of AUM. We have the familiar

[32] To initiate the recursive calculation, let $x^0_{i,t} = 0$ on the first day a stock appears in the sample.

result that the before-cost return (the first term) is scale-invariant while the percentage trading cost due to price impacts (the second term) scales with AUM linearly. We report the mean and Sharpe ratio of $\widetilde{r}_{\text{implemented},t}$. Additionally, we also evaluate the annualized turnover as

$$\text{Turnover} := \frac{1}{T}\sum_{i,t}\frac{\left|x_{i,t}-x^0_{i,t}\right|}{2A}\times 252 = \frac{1}{2T}\sum_{i,t}\left|w_{i,t}-w^0_{i,t}\right|\times 252. \tag{15}$$

This equation shows the turnover is scale invariant.

6.2 Implementing a simulated quantitative strategy

We consider a set of trading targets $\{x^*_{i,t}\}$ that simulate a quantitative investment strategy. We simulate a trading signal that, with 1% chance, perfectly forecasts whether a stock goes up or down over the next five days. The signal is independent across i, t. Following the signal, all stocks are allocated to either the long or the short group. When no signal is received, the stock position stays the same. We let, $x^*_{i,t}$ be the equal-weighted long-short strategy in which each leg sums up to 50% of the AUM.

This portfolio has an unrealistically high before-cost OOS Sharpe ratio of around 7, which is brought down substantially (to more reasonable levels) after trading costs. We experiment with a sequence of AUM magnitudes up to $10 billion which command varying levels of trading costs. We first evaluate across a grid of μ values that vary the aggressiveness of trades, and then perform the μ-tuning method that allows for the comparison of the highest investment performance attained by each volume prediction method.

We run a trading experiment for every method and each μ value for various AUM levels. Each experiment plots one dot in Figures 5 and 6, with OOS turnover versus mean return (or Sharpe ratio) as the coordinates. Each curve corresponds to one method, connecting the dots with varying μ values.

To understand the general shape of the curves, first note that under different μ values the strategies vary from always passively holding ($z = 0$) to trading all the way to the target ($z = 1$). As the implemented portfolio becomes more aggressive (μ increase), the turnover increases. On

Figure 5: Trading experiment performance, with only tech predictors

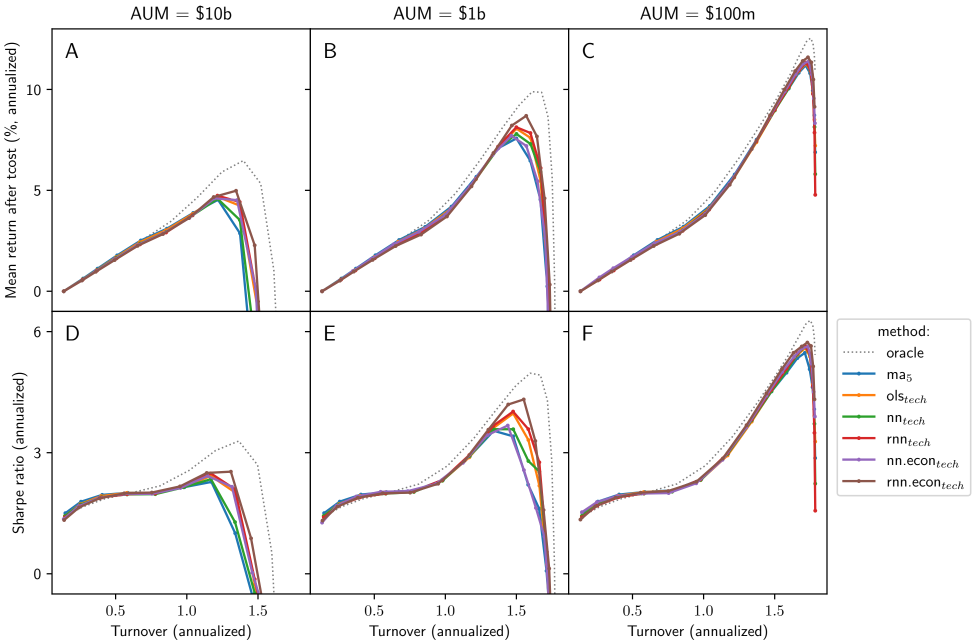
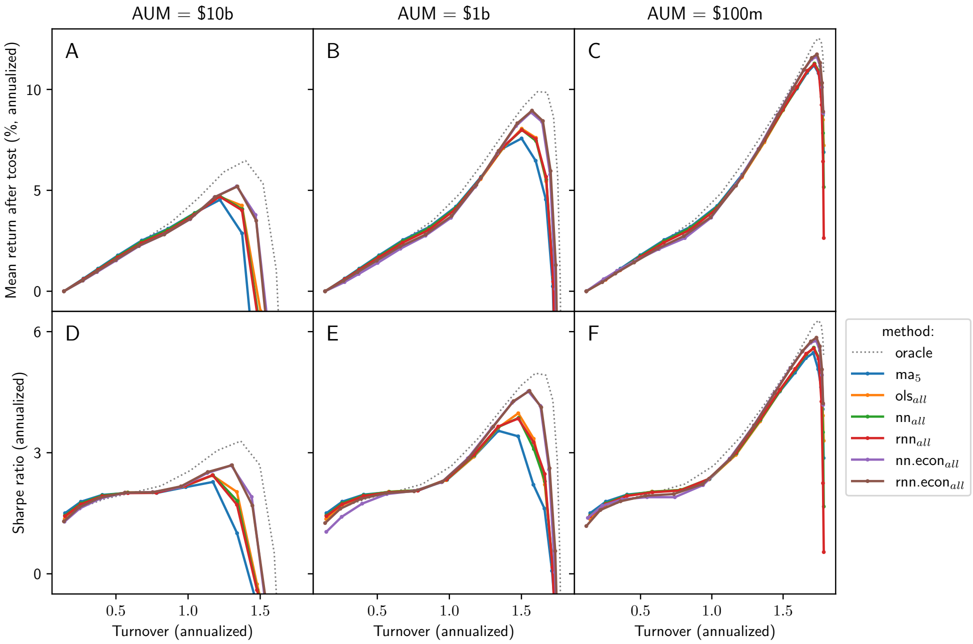


Figure 6: Trading experiment performance, with all predictors



Each dot plots the outcome of one trading experiment: the horizontal coordinate is turnover, vertical is the mean return or Sharpe ratio. Each curve is for a method with varying μ values. Figures 5 and 6 differ in using only the tech or all sets of predictors, while the benchmarks curves “ma$_5$” and “oracle” are the same.

the vertical axis, the mean return (or Sharpe ratio) first increases as a result of active trading for profit and then bends downward due to trading costs, whose effect shows up with high turnover. The two opposing forces result in hump-shaped curves. For higher AUM, the trading costs' effect is stronger, resulting in lower curves that peak at smaller turnover levels (i.e., lower μ values). In the next subsection, we implement the highest attainable investment outcome by selecting the μ value from the peak of the in-sample curves.

The investment gain of a better trading rate strategy $z(\mathcal{X})$ is shown in the vertical displacement of the curves. The improvement comes from two aspects. First, more closely tracking the target portfolio delivers a higher return. Second, reducing trading costs when liquidity is expected to be low, for a given amount of turnover, lowers the trading cost drag on the portfolio. These two objectives are contradictory in our framework, hence the value of predicting volume more accurately is to strike a better balance in the tradeoff between trading costs and tracking error. At the two extremes of the curves ($\mu = 0$ or $+\infty$), there is no room for tradeoff, and hence predicting volume has no value, in which case all curves converge.

Comparing the methods, we can analyze the sources of improvement. First, using the full set of predictors (Figure 6) is better than using only the set of technical predictors (Figure 5), which is still much better than no additional information besides the five-day moving average. Second, holding the information set constant, the neural network model predicts volume better than linear regression. Lastly, given the pre-trained neural network, further fine-tuning the trading strategy by directly optimizing the investment performance provides further improvement.

Finally, we tune the hyperparameter μ to report each method's highest attainable investment performance. We choose the μ value that maximizes the in-sample expected return (or Sharpe ratio) and report the out-of-sample performance at the tuned μ. Effectively, μ is selected from the peaks of the in-sample version of the curves (not plotted) and then applied to the OOS curves (as plotted). We expect the in-sample and OOS curves to peak at relatively close μ ranges so that the method attains an OOS performance close to the peak of the OOS curves.

The results are reported in Table 3. Applying better prediction methods and using a larger set

[32] The "relevant AUM" row in Table 2 is calculated according to the μ-tuning result under each AUM level under method ma_5.

Table 3: Investment performance in trading experiments

A. Mean return (%, annualized)					B. Sharpe ratio (annualized)			
AUM	\$10b	\$1b	\$100m	\$10m	\$10b	\$1b	\$100m	\$10m
ma$_5$	3.88	6.47	11.19	13.20	2.00	2.21	5.47	6.55
ols$_{tech}$	3.82	7.60	11.28	13.14	2.16	3.32	5.59	6.59
nn$_{tech}$	3.76	7.30	11.32	13.13	2.14	2.79	5.63	6.59
rnn$_{tech}$	3.74	7.84	11.33	13.13	2.18	3.58	5.64	6.59
nn.econ$_{tech}$	4.60	7.20	11.29	13.22	2.13	2.57	5.59	6.63
rnn.econ$_{tech}$	4.67	8.69	11.59	13.30	2.50	4.32	5.73	6.66
ols$_{all}$	3.82	7.60	11.28	13.14	2.17	3.35	5.60	6.59
nn$_{all}$	3.86	7.44	11.28	13.13	2.19	3.09	5.60	6.58
rnn$_{all}$	3.79	7.55	11.25	13.09	2.18	3.26	5.59	6.56
nn.econ$_{all}$	4.64	8.87	11.61	13.29	2.18	4.24	5.74	6.66
rnn.econ$_{all}$	4.68	8.95	11.77	13.30	2.50	4.53	5.85	6.68
oracle	6.47	9.89	12.54	13.56	3.05	4.97	6.28	6.80

For each combination of AUM and method, we report the OOS mean return (Panel A) and Sharpe ratio (Panel B) at the tuned μ, which is selected to maximize the in-sample mean return (or Sharpe ratio) over a grid of μ values.

of predictors improves investment performance uniformly across AUM levels. The economic magnitude of improvement is significant, comparable to, if not more than, the marginal improvement from innovating on return prediction signals. For \$10 billion of AUM, the average annual return increases from 3.88% when using the baseline prediction method of only a 5-day moving average volume, to 4.68% when making volume predictions with an economic objective optimization imbedded within a rnn. For \$1 billion AUM, the magnitude of improvement is even greater, going from 6.47% to 8.98%, and more than doubling the Sharpe ratio from 2.21 to 4.53.

For smaller AUM (\$10m), the improvement is still noticeable but smaller, because price impact shrinks and all methods therefore prescribe trading very aggressively. The investment performance converges to the high before-cost level regardless of the prediction method. For more realistic considerations, future research could consider per-unit trading costs such as bid-ask spread in addition to price impact, which tend to show up as dollar trade sizes shrink.

Appendix B.5 breaks down the mean return and Sharpe ratio improvements reported in Table 3 by firm size groups. We find the gain is robustly positive across all size groups from nano to mega stocks. In particular, the gain is not concentrated in the smaller firms, and larger stocks have monotonically greater gains when measured as the percentage relative to the oracle level.

6.3 Implementing factor zoo portfolios

As another set of trading experiments, we use as trading targets the portfolios sorted on characteristics in the JKP dataset that come from the asset pricing literature. The goal is to examine the improvement in implementation outcomes across different investment styles.

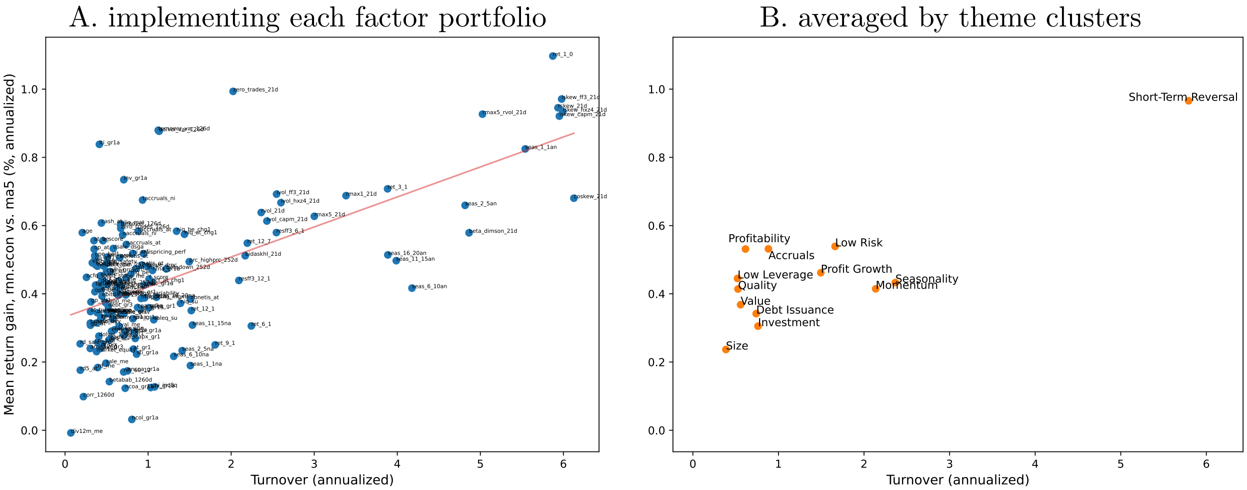
For each of the 153 characteristics, the target $\{x^*_{i,t}\}$ is formed in a standard fashion without considering trading costs: at the start of every month, the cross section of stocks is split at the 50th quantile into equal-weighted long-short portfolios based on each characteristic. We fix the AUM at \$10 billion (\$5 billion for each of the long and short legs).

We fix μ across the 153 factors for consistent comparison, and because factor-specific μ tuning is likely to be unstable. For example, consider factors that happen to earn negative realized returns during the training sample period, the factor-specific optimal μ would be zero – no trading at all. By evaluating at a fixed positive μ, we can address whether this factor that happens to lose money in the sample, would lose less with a better implementation. We pick the μ that optimizes the average gain across all factors, with results robust to perturbations in μ.

Figure 7 plots the gain in mean return when implementing the factor portfolios with the rnn.econ$_{all}$ volume prediction compared with using the ma$_5$ volume prediction. The horizontal axis is the turnover of the target factor portfolio. Panel B averages the points by investment style clusters (from JKP). The plots show volume prediction from the rnn.econ$_{all}$ benefits portfolio implementation across the factors. The average gain in mean after-cost return across factors is 0.44% per year from volume prediction alone using the rnn.econ$_{all}$ model versus the simple moving average. Almost all of the 153 factors have positive gains. With the \$10 billion AUM scale, this translates into an additional \$44 million per year in implementation cost-saving from improved volume prediction.

Across factors, the gain is larger for those factors with higher turnover. In the right region of Figure 7, some raw factors have a turnover approaching six (600% per year or roughly turning over half of the AUM every month). The gains for these factors, including various short-term reversal

Figure 7: Mean return improvements in implementing each factor portfolio


A. implementing each factor portfolio

B. averaged by theme clusters

Each dot implements one JKP factor portfolio. The y-axis is the difference in after-cost mean excess return between implementing with the rnn.econ$_{all}$ and the ma$_5$. The x-axis is the turnover of the factor portfolio target (i.e., Eq. 15 with $x_{i,t} = x^*_{i,t}, x^0_{i,t} = x^*_{i,t-1}$). Panel B averages the points in A by style clusters (from JKP).

strategies, are around 0.5% to 1.0% per year. These factors are constructed with technical signals over a shorter window.[33] In the left part of the figure, even factors with low turnover (those relying on quarterly fundamental signals and signals with greater persistency) show gains that range from 0.2% to 0.6% per year from volume prediction. These improvements are significant.

Appendix Figure B.2 plots the same gains in the vertical axis but changes the horizontal axis to the mean return attained by ma$_5$, i.e., the baseline level in the gain calculation. The figure shows that, regardless of the baseline, the gain is independently distributed around a positive center. That is, a better volume prediction is uniformly effective, and the improvement is not concentrated on factors that have positive (or negative) realized returns. Appendix Figure B.3 reports similar plots with gains measured in Sharpe ratio space instead of mean returns.

[33] Examples: `ret_1_0` short term reversal; `iskew_capm_21d` idiosyncratic skewness from the CAPM; `iskew_ff3_21d` idiosyncratic skewness from the FF3F; `rmax5_rvol_21d` highest 5 days of return; `rskew_21d` return skewness 21d; `seas_1_1an` 1 Year Annual Seasonality; and `coskew_21d` coskewness.

7 Next Steps

While we find substantial economic benefits from volume prediction using our framework and methods, there is much room for improvement. Our goal is not to develop the best trading cost model or even the best volume prediction model, but rather to translate the prediction problem into economic consequences, which yield interesting insights. A more exhaustive search for prediction variables and models that forecast volume more accurately could translate into even larger economic benefits than we show here.

For example, Frazzini, Israel, and Moskowitz (2018) show that market and idiosyncratic volatility are significant variables that impact real-world price impact costs. Using similar techniques as shown here to forecast market and individual stock volatility could further improve OOS net-of-cost performance. Of course, there is already a long literature on forecasting volatility and part of our objective was to focus on non-return variables. Moreover, to illustrate the power of incorporating a large set of predictors, ML techniques, and incorporating an economic objective into the prediction loss function, it is more transparent and clear to focus on only one variable in the simplest of cost models. Once we understand where the predictive benefits are coming from, incorporating multiple variables in a more sophisticated trading cost model is a natural next step (and next paper). Some other promising candidates for additional features of a trading cost model are lead-lag volume relations across stocks, more seasonal indicators, and other market microstructure variables.

Our simple framework for predicting and characterizing "volume alpha" also has limitations. For one, we study a very simple functional form for trading costs that maps volume prediction directly into costs. Other functional forms (and other determinants of costs beyond volume) may lead to improved results.

In addition, we separate the volume prediction problem from the expected return and variance/covariance modeling problem. Combining all three could generate further portfolio improvements and the interaction between these three prediction problems could be enlightening. Of course, attempting all of this at once obscures intuition for what, specifically, is leading to improved OOS performance. The current paper, by just focusing on trading costs, abstracting from predictability of return moments, and only focusing on a singular component of trading costs through trading volume, provides clear intuition for what drives improved performance. The next steps would be to incorporate other important predictors, like returns, which have already received a ton of attention from the literature.

Lastly, our trading experiments are merely a tool to illustrate some possible applications of our insights, but are not designed to optimize any performance outcome. Specifically, two things not considered in our design are: dynamic effects from trading and heterogeneous trade tasks. Adding these more complex features could be an interesting area for future research. Further improvements may also be found in more complex nn and rnn models, which is another research avenue worth pursuing.

8 Conclusion

We translate volume predictability into net-of-cost portfolio performance by linking it to expected trading costs. Volume is highly predictable, especially when using machine learning techniques, large data signals, and exploiting the virtue of complexity in prediction. We find that volume prediction can be as valuable as return prediction in achieving optimal mean-variance portfolios net of trading costs.

Incorporating an economic objective function directly into machine learning is even more effective for obtaining useful predictions. This feature may be general to many finance applications of machine learning, where incorporating the economic objective directly may dominate a two-step process that first satisfies some statistical objective and then incorporates that statistical object into an economic framework. For volume prediction, the asymmetric cost of overestimating versus underestimating volume is captured (ignored) by an economic (statistical) objective, and delivers sizeable economic impact.

Trading volume prediction in general is an interesting research area worthy of further exploration. While we have couched the volume prediction problem into a portfolio context to translate the problem into economic consequences, understanding the role of volume more generally – its causes and consequences – is interesting. Using some of our techniques may shed light on this question and may help pinpoint what components of volume are most valuable to trading costs and, as a byproduct, portfolio construction. For example, informed versus uninformed volume, volume with temporary versus permanent price impact, and short- versus long-term volume may be interesting research pursuits. Examining various aspects of volume could be very useful for improving portfolio optimization and understanding trading activity more broadly. We leave these issues for future work.

More generally, using a framework similar to ours can help assess the value of predicting non-return characteristics for asset pricing, potentially opening up asset pricing research to a host of interesting variables to examine.

References

Amihud, Y. 2002. Illiquidity and stock returns: Cross-section and time-series effects. Journal of Financial Markets 5:31–56.

Avramov, D., S. Cheng, and L. Metzker. 2023. Machine learning vs. economic restrictions: Evidence from stock return predictability. Management Science 69:2587–619.

Azevedo, V., C. Hoegner, and M. Velikov. 2023. The expected returns on machine-learning strategies. Available at SSRN 4702406 .

Balduzzi, P., and A. W. Lynch. 1999. Transaction costs and predictability: Some utility cost calculations. Journal of Financial Economics 52:47–78.

Banerjee, A., X. Guo, and H. Wang. 2005. On the optimality of conditional expectation as a Bregman predictor. IEEE Transactions on Information Theory 51:2664–9. Conference Name: IEEE Transactions on Information Theory.

Benston, G. J., and R. L. Hagerman. 1974. Determinants of bid-asked spreads in the over-the-counter market. Journal of Financial Economics 1:353–64.

Białkowski, J., S. Darolles, and G. Le Fol. 2008. Improving vwap strategies: A dynamic volume approach. Journal of Banking & Finance 32:1709–22.

Brandt, M. W., P. Santa-Clara, and R. Valkanov. 2009. Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns. The Review of Financial Studies 22:3411–47.

Brennan, M. J., and A. Subrahmanyam. 1995. Investment analysis and price formation in securities markets. Journal of Financial Economics 38:361–81.

Brock, W. A., and B. D. LeBaron. 1996. A dynamic structural model for stock return volatility and trading volume. The Review of Economics and Statistics 78:94–110.

Campbell, J. Y., S. J. Grossman, and J. Wang. 1993. Trading volume and serial correlation in stock returns. The Quarterly Journal of Economics 108:905–39.

Çetin, U., R. A. Jarrow, and P. Protter. 2004. Liquidity risk and arbitrage pricing theory. Finance and Stochastics 8:311–41.

Chen, A. Y., and M. Velikov. 2023. Zeroing in on the expected returns of anomalies. Journal of Financial and Quantitative Analysis 58:968–1004.

Chen, A. Y., and T. Zimmermann. 2020. Publication bias and the cross-section of stock returns. The Review of Asset Pricing Studies 10:249–89.

Chen, H., Y. Cheng, Y. Liu, and K. Tang. 2023. Teaching economics to the machines. Available at SSRN 4642167 .

Chen, L., M. Pelger, and J. Zhu. 2023. Deep learning in asset pricing. Management Science .

Chordia, T., S.-W. Huh, and A. Subrahmanyam. 2007. The cross-section of expected trading activity. The Review of Financial Studies 20:709–40.

Chordia, T., R. Roll, and A. Subrahmanyam. 2011. Recent trends in trading activity and market quality. Journal of Financial Economics 101:243–63.

Cong, L. W., K. Tang, J. Wang, and Y. Zhang. 2021. AlphaPortfolio: Direct construction through deep reinforcement learning and interpretable AI. Available at SSRN 3554486 .

Darrat, A. F., S. Rahman, and M. Zhong. 2003. Intraday trading volume and return volatility of the djia stocks: A note. Journal of Banking & Finance 27:2035–43.

Datar, V. T., N. Y. Naik, and R. Radcliffe. 1998. Liquidity and stock returns: An alternative test. Journal of Financial Markets 1:203–19.

DeMiguel, V., A. Martin-Utrera, F. J. Nogales, and R. Uppal. 2020. A transaction-cost perspective on the multitude of firm characteristics. The Review of Financial Studies 33:2180–222.

Detzel, A., R. Novy-Marx, and M. Velikov. 2023. Model comparison with transaction costs. The Journal of Finance 78:1743–75.

Easley, D., M. López de Prado, M. O'Hara, and Z. Zhang. 2020. Microstructure in the machine age. The Review of Financial Studies 34:3316–63.

Engle, R. 2004. Risk and volatility: Econometric models and financial practice. American Economic Review 94:405–20.

Frazzini, A., R. Israel, and T. J. Moskowitz. 2012. Trading costs of asset pricing anomalies. Fama-Miller Working Paper, Chicago Booth Research Paper .

———. 2018. Trading Costs. doi:10.2139/ssrn.3229719.

Gallant, A. R., P. E. Rossi, and G. Tauchen. 1992. Stock prices and volume. The Review of Financial Studies 5:199–242.

Glosten, L. R., and L. E. Harris. 1988. Estimating the components of the bid/ask spread. Journal of Financial Economics 21:123–42.

Glosten, L. R., and P. R. Milgrom. 1985. Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. Journal of financial economics 14:71–100.

Goldstein, I., C. S. Spatt, and M. Ye. 2021. Big data in finance. The Review of Financial Studies 34:3213–25.

Grossman, S. J., and M. H. Miller. 1988. Liquidity and market structure. Journal of Finance 43:617–33.

Gu, S., B. Kelly, and D. Xiu. 2020. Empirical asset pricing via machine learning. The Review of Financial Studies 33:2223–73.

Gârleanu, N., and L. H. Pedersen. 2013. Dynamic trading with predictable returns and transaction costs. Journal of Finance 68:2309–40. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jofi.12080.

———. 2016. Dynamic portfolio choice with frictions. Journal of Economic Theory 165:487–516.

Harvey, C. R., Y. Liu, and H. Zhu. 2016. . . . and the cross-section of expected returns. The Review of Financial Studies 29:5–68.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 770–8.

Ho, T. S., and H. R. Stoll. 1983. The dynamics of dealer markets under competition. Journal of Finance 38:1053–74.

Hochreiter, S., and J. Schmidhuber. 1997. Long short-term memory. Neural Computation 9:1735–80.

Jensen, T. I., B. Kelly, and L. H. Pedersen. 2022. Is there a replication crisis in finance? Journal of Finance .

Jensen, T. I., B. T. Kelly, S. Malamud, and L. H. Pedersen. 2024. Machine learning and the implementable efficient frontier. Swiss Finance Institute Research Paper .

Kaastra, I., and M. S. Boyd. 1995. Forecasting futures trading volume using neural networks. Journal of Futures Markets 15.

Kelly, B., S. Malamud, and K. Zhou. 2024. The virtue of complexity in return prediction. Journal of Finance 79:459–503.

Kelly, B., and D. Xiu. 2023. Financial machine learning. Foundations and Trends in Finance 13:205–363.

Korajczyk, R. A., and R. Sadka. 2004. Are momentum profits robust to trading costs? Journal of Finance 59:1039–82.

Kyle, A. S. 1985. Continuous auctions and insider trading. Econometrica 53:1315–35. Publisher: [Wiley, Econometric Society].

Linnainmaa, J. T., and M. R. Roberts. 2018. The history of the cross-section of stock returns. The Review of Financial Studies 31:2606–49.

Lo, A. W., and J. Wang. 2000. Trading volume: definitions, data analysis, and implications of portfolio theory. The Review of Financial Studies 13:257–300.

McLean, R. D., and J. Pontiff. 2016. Does academic research destroy stock return predictability? Journal of Finance 71:5–32.

Patton, A. J. 2020. Comparing Possibly Misspecified Forecasts. Journal of Business & Economic Statistics 38:796–809. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07350015.2019.1585256.

Shleifer, A., and R. W. Vishny. 1997. The limits of arbitrage. Journal of Finance 52:35–55.

Simon, F., S. Weibels, and T. Zimmermann. 2025. Deep parametric portfolio policies. Working Paper, CFR Working Paper.

Stoll, H. R. 1978. The supply of dealer services in securities markets. Journal of Finance 33:1133–51.

Internet Appendix

Trading Volume Alpha

Ruslan Goyenko — McGill

Bryan Kelly — Yale, AQR, and NBER

Tobias Moskowitz — Yale, AQR, and NBER

Yinan Su — Johns Hopkins

Chao Zhang — HKUST (GZ)

A Technical details

A.1 Neural network implementation details

The nn architecture consists of three fully-connected hidden layers with 32, 16, and 8 neurons, respectively.

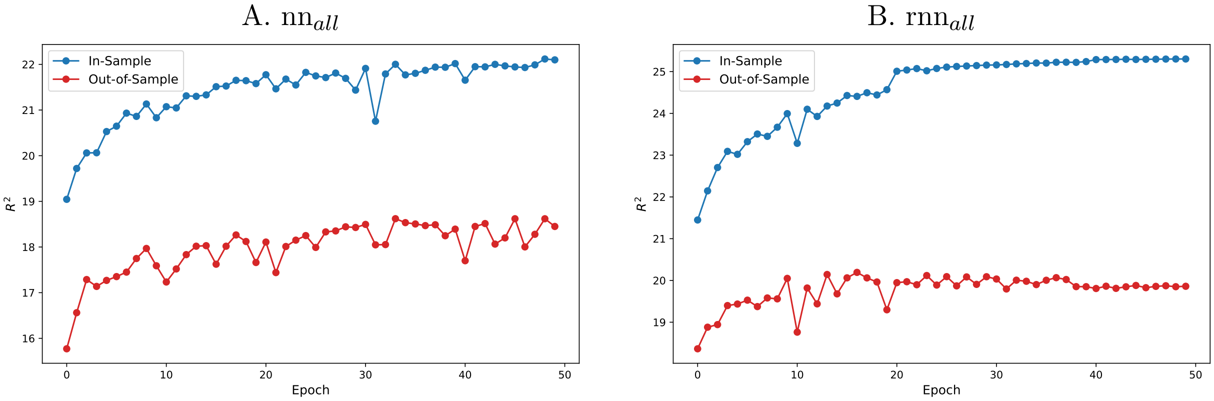
The rnn architecture is similar to that of the nn. The first (bottom) hidden layer in the 3-layer network is upgraded to an lstm layer with 32 hidden states and cell states, respectively. The remaining two layers are unchanged: fully connected with 16 and 8 neurons, respectively.

The formulas for the number of parameters in nn and rnn, as reported in Table 1 B, are stated below. For nn (three fully-connected hidden layers with 32-16-8 neurons), the formula is $(\#\text{ of predictors} + 1) \times 32 + (32 + 1) \times 16 + (16 + 1) \times 8 + (8 + 1)$. In rnn, the first hidden layer has 32 hidden states and 32 cell states with four gates, changing the formula to $(\#\text{ of predictors} + 32 + 1) \times 32 \times 4 + (32 + 1) \times 16 + (16 + 1) \times 8 + (8 + 1)$.

In training the rnn, we implement the “many-to-one” type data pipeline, where the model recursively processes a sequence of 10 inputs, $X_{i,t-9}, \ldots, X_{i,t}$, and produces a single output $\widehat{\eta}_{i,t}$ to calculate the training loss at each data point $\{i, t\}$. (When increasing the sequence length from 10 to 50, the results had minimum improvements but required much larger GPU memory and longer training time.) For data points at the beginning of a stock’s observed period where lagged predictors (e.g., $X_{i,t-9}$) are not available, we fill in with zero vectors.

In training both nn and rnn models, we use the Adam optimizer, with default learning rates and other parameters. The batch size is 1024. The $\widetilde{\eta}$ prediction models are trained with 50 epochs. For the sake of clear benchmarking, we do not adopt early stopping, weights dropout, or hyperparameter tuning with cross-validation, though these techniques could further boost the prediction accuracy. The machine learning program is implemented with the PyTorch package.

Figure A.1: Learning curves



R^2 of the $\widetilde{\eta}$ prediction models as training progresses (epochs).

The learning curves show the gap between the IS and OOS R^2 is relatively small, and does not widen with continued training. This indicates limited in-sample overfitting at this neural network configuration. The rnn learning curves show slightly more severe overfitting, though the OOS learning curve is still relatively stable as training continues. The learning curves display fluctuation due to the randomness of the stochastic gradient descent, though the extent to which is not severe. We also note the exact results depend on the inherent randomness of the training program. We find the quantitative results are insensitive to random seeds and report the average of five independent runs to obtain a robust evaluation outcome.

Table A.1 summarizes computational costs in terms of the training times and memory usage for nn and rnn with all predictors for a single random seed. These experiments were conducted on a system equipped with an Nvidia A100 GPU with 40 GB of GPU memory, an AMD EPYC 7713 64-Core Processor @ 1.80GHz with 128 cores, and 1.0TB of RAM, running Ubuntu 20.04.4 LTS.

Table A.1: Training time and memory usage

	Training time (hours)	CPU memory usage (GB)	GPU memory usage (GB)
nn_{all}	0.48	10.87	1.22
rnn_{all}	0.63	144.98	1.48

B Additional empirical results

B.1 Prediction accuracy result in a different R^2 measure

Table B.2 recast the R^2's in Table 1 in terms of the explained percentage of the total variation of $\widetilde{v}$. There is no change in the model, fitted values, nor the mean squared error. Only the R^2 is calculated with a different numerator. Under this metric, the R^2's are always high since the benchmark moving average explains $\widetilde{v}$ to a large extent already (93.68%), while machine learning and more predictors still provide accuracy improvement.

Table B.2: Table 1 prediction accuracy result in a different R^2 measure

cumulatively adding predictor sets total number of predictors	tech 8	fund-1 14	fund-2 161	calendar 165	earnings 175
A: R^2 relative to $\widetilde{\eta}$ (log dollar volume shock), Table 1A repeated for reference					
ma_5	0				
ols	12.09	12.26	12.27	14.85	15.99
nn	14.31	14.90	14.42	17.13	18.45
rnn	15.80	16.25	15.47	18.12	19.86
A': R^2 relative to $\widetilde{v}$ (log dollar volume)					
ma_5	93.68				
ols	94.44	94.45	94.45	94.62	94.69
nn	94.58	94.62	94.59	94.76	94.85
rnn	94.68	94.69	94.64	94.86	94.93

Panels A and A' respectively express the OOS prediction accuracy in two different R^2's. The R^2 is calculated with ma_5 as the benchmark: $R^2 = 1 - \text{MSE}/\text{avg}(\widetilde{v} - \text{ma}_5)^2$. Panel B changes the benchmark to "predicting 0" essentially: $R^2 = 1 - \text{MSE}/\text{avg}(\widetilde{v} - \text{avg}(\widetilde{v}))^2$, where $\text{MSE} := \text{avg}(\widetilde{v} - \widehat{v})^2 = \text{avg}(\widetilde{\eta} - \widehat{\eta})^2$ and $\text{avg} := \frac{1}{|\text{OOS}|}\sum_{i,t\in\text{OOS}}$ is the OOS average.

B.2 Volume forecasting by standardizing target variable $\widetilde{\eta}$

In Section 3, we forecast volume by having the supervision target variable as the log dollar volume *shock* defined as daily log dollar volume minus the moving average in the past five days, $\widetilde{\eta}_{i,t} := \widetilde{v}_{i,t} - [\text{ma}_5]_{i,t}$, where $[\text{ma}_5]_{i,t} := \frac{1}{5}\left(\widetilde{v}_{i,t-1} + \cdots + \widetilde{v}_{i,t-5}\right)$. We have explained that predicting the log dollar volume shock is essentially predicting the log dollar volume itself, which is simply the sum of the shock forecast and the moving average: $\widehat{v}_{i,t} = \widehat{\eta}_{i,t} + [\text{ma}_5]_{i,t}$.

In this Appendix, we consider a related but different method by further standardizing the target variable $\widetilde{\eta}$, and having it as the supervision target of the prediction models. The idea is that the standardized target might be more well-behaved and be more amenable for the neural networks to form accurate predictions. However, we find that this standardization method does not improve the prediction accuracy compared to the benchmark method.

The procedure is the following. We standardize the target variable $\widetilde{\eta}$ by dividing its standard in the past 22 days: $\widetilde{\zeta}_{i,t} := \frac{\widetilde{\eta}_{i,t}}{[\text{stddev}]_{i,t}}$, where $[\text{stddev}]_{i,t} := \sqrt{\frac{1}{22}\left(\widetilde{\eta}^2_{i,t-1} + \cdots + \widetilde{\eta}^2_{i,t-22}\right)}$.[34] Then we run the same ols, nn, and rnn models to predict $\widetilde{\zeta}$ with the same sets of predictors. We back out the forecasted values by reversing the standardization: $\widehat{\eta}_{i,t} = \widehat{\zeta}_{i,t} \times [\text{stddev}]_{i,t}$. And finally, we report the same OOS R^2 evaluations as in Section 3.

The results are reported in Table B.3, and are compared with that of the benchmark method in Table 1 for ease of comparison. We find although the standardization method still yields positive R^2 values for the nn and rnn methods, these are consistently lower than the benchmark values reported in Table 1. The relative comparisons between the nn and rnn methods as well as among the different sets of predictors are still the same as the benchmark method, highlighting the robustness of the results presented in the main text.

B.3 Prediction results in firm size groups and "mixture of experts" forecasts

Table B.4 Panel A provides additional assessments of prediction accuracy by evaluating volume forecasts in different size groups. We use the five groups from the JKP data sorted on the firms'

[34] We have experimented with five-day standardization as well and found the results are similar.

Table B.3: Prediction accuracy (R^2 in %) using the standardization method

cumulatively adding predictor sets	tech	fund-1	fund-2	calendar	earnings
A: R^2 original method (repeating Table 1 Panel A)					
ma_5	0				
ols	12.09	12.26	12.27	14.85	15.99
nn	14.31	14.90	14.42	17.13	18.45
rnn	15.80	16.25	15.47	18.12	19.86
B. R^2 using the standardization method					
ols	-0.06	-0.06	-0.07	-0.06	-0.04
nn	11.79	12.05	11.30	12.29	13.73
rnn	14.42	14.51	12.36	14.94	16.23

market capitalization.[35]

The prediction accuracy increases as firm size increases, regardless of the prediction method. The R^2's evaluated in the mega firms are roughly twice those of the nano firms. As explained in the main text, smaller firms have a greater magnitude of unexpected trading volume shocks that are hardest to predict. This result makes sense since small firms are volatile and have low trading volume, hence unexpected events that give rise to volume spikes are more likely for these firms. This finding also indicates that in addition to small firms being less liquid on average, their liquidity is also less predictable and more volatile. Hence, tiny firms are not only costly to trade in general, but their costs are less predictable. These results are intuitive and suggest that our prediction models are capturing true variation in volume and not simply noise.

We also examine whether firms of different size groups should be modeled differently. We train a model on each size group separately to attempt to better capture the heterogeneity across the firm size dimension rather than pooling all firms in the same model. We implement a simple mixture of experts (moe) method, where each size group is trained separately to form "expert" models and then compared against the pooled training model in Panel A.

The moe improves performance of the ols method. Comparing the first lines in Panels A and B, linear models catered to different size groups are more accurate than the pooled ols. For nn and rnn, however, the sample size reduction outweighs the potential benefits of separate training,

[35] The five groups are defined according to the market capitalization breakpoints of NYSE stock percentiles: mega stocks, greater than the 80th percentile; large, 50–80; small, 20–50; micro, 1-20; and nano, below the 1st percentile.

Table B.4: Prediction accuracy (R^2 in %) in different size groups and "mixture of experts"

size group	jointly	nano	micro	small	large	mega
training obs	2,522,619	300,790	797,880	680,209	479,839	263,901
testing obs	1,893,067	273,792	467,413	552,503	384,819	214,540
A: pooled training evaluated in size groups and jointly (same models as in Table 1)						
ols_{all}	15.99	13.32	12.60	20.90	25.49	26.16
nn_{all}	18.45	15.80	14.86	23.71	27.76	29.12
rnn_{all}	19.86	16.63	16.14	26.00	30.50	32.02
B: size group training evaluated in size groups and jointly (mixture of experts)						
ols+moe_{all}	16.34	13.68	12.73	21.43	25.93	27.47
nn+moe_{all}	17.78	15.29	14.43	22.69	26.57	27.71
rnn+moe_{all}	18.26	15.24	14.71	24.76	29.02	30.99

Panel A evaluates the benchmark models (pooled training) in the five size groups, respectively, in the OOS period. Each model uses "all" 175 predictors. Column "jointly" repeats Table 1, Panel A, last column. Panel B trains "expert" models for each size group separately and evaluates them in their corresponding size groups in the OOS period. Column "jointly" evaluates the mixture of experts (moe) model, which predicts with the corresponding expert model trained on the same size group for each OOS data point. Each R^2 value is the average of five runs, same as Table 1.

making the mixture of expert models less accurate (either conditioning on size groups or jointly). Since the potential non-linear effects of firm size are already allowed in the neural networks, forcing size groups into different models is less effective. For this reason, we stick with pooled training samples.

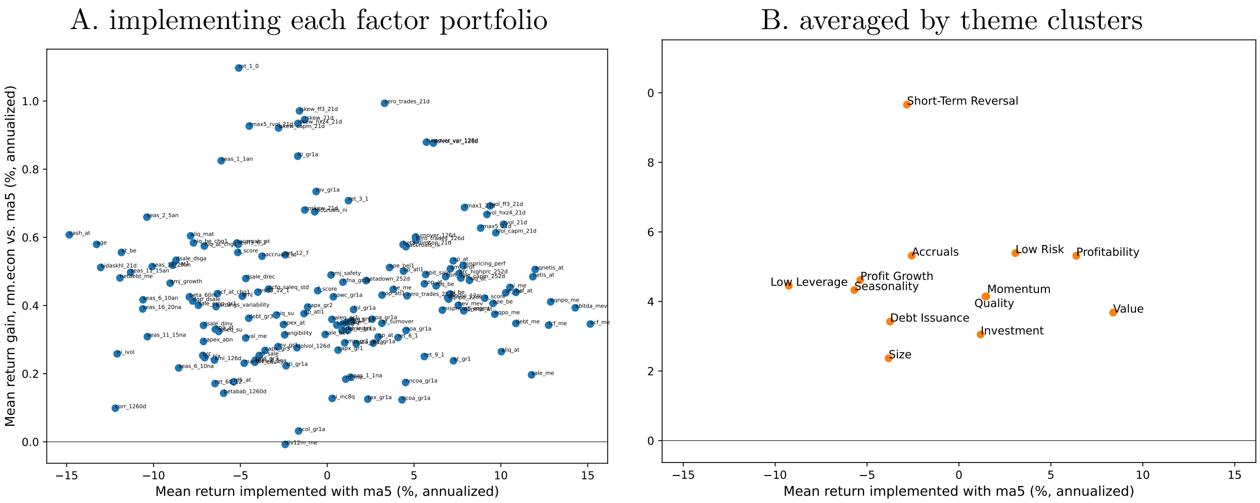
B.4 Additional results of implementing factor zoo portfolios

We provide results on the trading experiments that implement the factor zoo portfolios in addition to Subsection 6.3.

Figure B.2 has the same vertical axis as Figure 7, which is the improvement in after-cost mean excess return from ma_5 to rnn.econ_{all}. The horizontal axis changes to the mean excess return achieved with the ma_5 method, i.e., the baseline level of the improvement. The plot shows the gain is distributed around a positive center uncorrelated with the baseline. That is, a better volume prediction is uniformly effective, and the improvement is not concentrated on factors that have positive (or negative) realized returns.

Figure B.3 is the Sharpe ratio version of Figure 7 by showing the gain in Sharpe ratio instead of the mean return. The plot shows a similar pattern to the one reported in the main text. The gains in Sharpe ratios are larger for those factors with higher turnover, reaching around 0.3 to 0.4 per year.

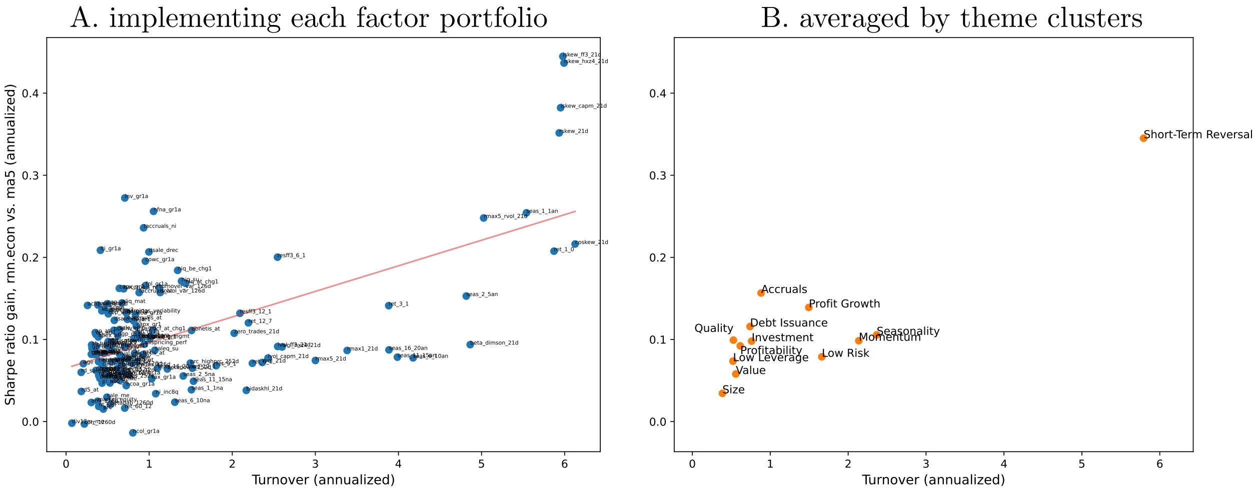
Figure B.2: Mean return improvements in implementing each factor portfolio


A. implementing each factor portfolio

B. averaged by theme clusters

The same plot as Figure 7, but changing the x-axis to mean return achieved with the ma_5 method, i.e., the baseline of the gain.

Figure B.3: Sharpe ratio improvements in implementing each factor portfolio


A. implementing each factor portfolio

B. averaged by theme clusters

The same plot as Figure 7, but showing the gain in Sharpe ratio instead of mean return.

B.5 Volume prediction's economic values in different firm size groups

In Section 6, we show volume prediction has significant economic value in portfolio tracking tasks. We now further evaluate the economic value in different firm size groups. We break down the improvement of the portfolio tracking experiment in mean return and Sharpe ratio by firm size groups as defined in JKP. We find the gain is robustly positive across all size groups. In particular, the gain is not concentrated in the smaller firms as the mega and large firms also have significant mean return and Sharpe ratio gains. The ratio of the gain relatively to what the oracle model (which represents the impossible upper bound) could have achieved is monotonically higher in the larger size group, whereas the absolute level of the gain varies.

The detailed evaluation method is the following. We do not retrain the volume prediction models nor create new portfolio implementation tasks. Instead, we take the same portfolio trading outcome as reported in Section 6.2 and break down the sample panel of firm-day portfolio outcomes $\{x_{i,t}\}$ by the five size groups. This is as if our portfolio tracking task is conducted jointly by five trading desks, each specialized in the stocks of a size group. We respectively evaluate each desk's performances (mean return and Sharpe ratio). We focus on the simulated tracking task with a $10 billion AUM as defined in Section 6.2.

Table B.5: Volume "alpha" (improvement of investment performance) in different size groups

A. Mean return (%, annualized)						B. Sharpe Ratio (annualized)				
	mega	large	small	micro	nano	mega	large	small	micro	nano
ma_5	5.54	6.18	3.58	2.58	2.54	1.81	2.24	1.36	0.86	0.46
rnn.econ_{all}	6.09	7.46	4.51	2.93	3.46	2.55	3.46	1.68	1.15	0.54
gain	0.55	1.28	0.93	0.35	0.92	0.74	1.22	0.32	0.29	0.08
oracle	6.28	8.27	5.98	4.26	9.84	2.60	3.65	1.94	1.40	1.22
pct gain to oracle	75	61	39	21	13	94	87	55	54	11

Line "gain" is the difference between the first two rows in each column (rnn.econ_{all} - ma_5). Line "pct gain to oracle" is the percentage of the gain of the rnn.econ_{all} relative to what the oracle model could have achieved over the ma_5 strategy. That is, "pct gain to oracle" = (rnn.econ_{all} - ma_5) / (oracle - ma_5)×100.

We find the trading volume alpha, i.e. the improvement in investment performance brought by volume prediction, is robustly positive across all size groups. The mega (NYSE 80th market cap percentile and above) and large (50-80 percentiles) firms have significant mean return and Sharpe ratio gains. We also report what the oracle model could have achieved, which is the highest portfolio outcome should the true volume be revealed to the investor. There is a varying level of oracle performance across size groups, which has nothing to do with the volume prediction but reflects the tracking target's different performance in different size groups as well as different levels of liquidity across size groups. To control for that variation, we calculate the percentage of the gain relative to the oracle performance. We find the percentage gain is monotonically higher in a larger size group. We believe this is because larger firms tend to have higher volume levels and overall more aggressive trading, which are most susceptible to highlighted trading costs when volume is over-predicted; and our trading intensity strategy (z) with economic learning is particularly effective in these cases to avoid the high costs of trading too much when volume is low.

C Additional theoretical analysis

C.1 Microfoundation of the tracking error penalty in the portfolio objective

The track error penalty term in the portfolio optimization objective function can be economically founded. We show the quadratic tracking error penalty can be derived from a mean-variance utility function. This analysis connects the target positions x_i^* to the before-cost mean-variance efficient portfolio weight, which increases in the asset's return expectations as well as the total portfolio size (AUM). It also provides a microfoundation of the hyperparameter μ and the quadratic penalty: as the position deviates away from the target, the mean-variance loss increases in a quadratic fashion.

Assume the agent has a mean-variance utility function, $U = \mathbb{E}A' - \frac{\gamma}{2A}\mathbb{V}\text{ar}A' - TCost$, where A' is the before-cost investment outcome, A is the initial wealth (AUM), and γ is the risk aversion coefficient, $TCost$ is the term for the objective of minimizing transaction costs (not the focus here as we are justifying the tracking error part). A' is the outcome of the portfolio strategy: $A' = A(1 + r_\text{f}) + \sum_i x_i r_i$, where x_i is the dollar position in risky asset i with excess return r_i. Assume $\mathbb{E}r_i = m_i$, $\mathbb{V}\text{ar}r_i = \sigma^2$, and zero covariances. The agent's portfolio optimization problem is choosing $\{x_i\}$ to maximize U.

Then, the objective function is

$$
\begin{aligned}
U &= A(1+r_{\mathrm{f}}) + \sum_i x_i m_i - \frac{\gamma}{2A}\sum_i x_i^2\sigma^2 - TCost && (16)\\
&= \sum_i \left(-\frac{\gamma\sigma^2}{2A}x_i^2 + m_i x_i\right) + A(1+r_{\mathrm{f}}) - TCost && (17)\\
&= -\frac{\gamma\sigma^2}{2A}\sum_i \left(x_i^2 - \frac{2A}{\gamma\sigma^2}m_i x_i\right) + A(1+r_{\mathrm{f}}) - TCost && (18)\\
&= -\frac{\gamma\sigma^2}{2A}\sum_i \left(x_i - \frac{A}{\gamma\sigma^2}m_i\right)^2 + \left(\frac{A}{2\gamma\sigma^2}\sum_i m_i^2 + A(1+r_{\mathrm{f}})\right) - TCost && (19)
\end{aligned}
$$

The first term matches the tracking error term modeled in Eq. 4. The second term is the before TCost utility at the zero tracking error portfolio ($x = x^*$). This term is constant of the x choice, hence can be ignored in the optimization problem.

Comparing the first term with the tracking error modeled in Eq. 4, we see the target portfolio is $x_i^* = \frac{A}{\gamma\sigma^2}m_i$. As expected, the target positions are the result of Markowitz mean-variance optimization. They are proportional to the return expectations in the cross section. They scale linearly with the AUM (A) and inversely with the risk aversion coefficient and volatility. Additionally, the overall tracking error penalizing coefficient $\mu = \frac{\gamma\sigma^2}{A}$, which is decreasing in AUM. The rationale is that the quadratic penalty stems from the quadratic risk change as the position deviates away from the target, and that the absolute risk aversion coefficient decreases with the wealth level. Although the main analysis takes μ as a hyperparameter ignoring its microfoundation, we still observe the negative relationship between the tuned μ and the economically relevant AUM (e.g., in Section 5.3).

C.2 The economic task as predicting $\widetilde{z}$

We have shown the economic task of choosing the z strategy, $\min_{z(\cdot)} \sum_{i,t\in\text{train}} loss^{\text{econ}}(\widetilde{v}_{i,t}, z(\mathcal{X}_{i,t}); \mu)$, can be seen as a prediction task of predicting the $\widetilde{v}$ with the economic loss function: $\min_{v(\cdot)} \sum_{i,t\in\text{train}} loss^{\text{econ}}_{\text{vv}}(\widetilde{v}_{i,t}, v(\mathcal{X}_{i,t}); \mu)$. In this appendix, we provide the equivalent representation as a prediction problem of the oracle trading rate $\widetilde{z} := s(\widetilde{v}; \mu)$.

Define $loss^{\text{econ}}_{\text{zz}}(\widetilde{z}, z; \mu) := loss^{\text{econ}}(s^{-1}(\widetilde{z}; \mu), z; \mu)$, where $s^{-1}(\,\cdot\,; \mu)$ is inverse of $s(\,\cdot\,; \mu)$ function. Under this definition, the economic task can be viewed as the problem of looking for a function $z(\cdot)$ that maps $\mathcal{X}$ into z to minimize the training sample average loss:

$$\min_{z(\cdot)} \sum_{i,t \in \text{train}} loss^{\text{econ}}_{\text{zz}}(\widetilde{z}_{i,t}, z(\mathcal{X}_{i,t}); \mu) \tag{20}$$

According to the definition, the analytical expression of $loss^{\text{econ}}_{\text{zz}}(\widetilde{z}, z; \mu)$ is

$$loss^{\text{econ}}_{\text{zz}}(\widetilde{z}, z; \mu) = \frac{\mu}{\widetilde{z}}(z - \widetilde{z})^2 + \mu(1 - \widetilde{z}). \tag{21}$$

In this expression, the loss can be seen as the squared z prediction error weighted by $\frac{\mu}{\widetilde{z}}$. The last term is constant of choice z so can be ignored in optimization. It equals $loss^{\text{econ}}_{\text{zz}}(\widetilde{z}, \widetilde{z}; \mu)$, the baseline loss incurred even with the perfect prediction.

To derive this expression, notice $\widetilde{z}$ is already defined as the perfect trading given $\widetilde{\lambda}$ or $\widetilde{v}$.

$$\widetilde{z} = \frac{\mu}{\mu + \widetilde{\lambda}} \Longrightarrow \widetilde{\lambda} = \frac{\mu}{\widetilde{z}} - \mu \tag{22}$$

Then, start from Eq. 8, represent $\widetilde{\lambda}$ with $\widetilde{z}$ and then complete the square:

$$\begin{aligned} loss^{\text{econ}}_{\text{zz}}(\widetilde{z}, z; \mu) &= \widetilde{\lambda} z^2 + \mu(1 - z)^2 \\ &= \left(\frac{\mu}{\widetilde{z}} - \mu\right) z^2 + \mu(1 - z)^2 \\ &= \frac{\mu}{\widetilde{z}}(z - \widetilde{z})^2 + \mu(1 - \widetilde{z}) \end{aligned}$$

We show $loss^{\text{econ}}_{\text{zz}}(\widetilde{z}, z; \mu)$ is still meaningfully different from the standard squared error loss function, and that $\mathbb{E}\left[\widetilde{z}|\mathcal{X}\right]$ will not be the optimal choice either further below.

C.3 Economic loss functions are not in Bregman class

We show functions $loss^{\text{econ}}_{\text{vv}}(\widetilde{v}, v; \mu)$ and $loss^{\text{econ}}_{\text{zz}}(\widetilde{z}, z; \mu)$ are not in the Bregman class, for all μ.

Without loss of generality, any loss function $F(p, q)$ can be normalized as $\bar{F}(p, q) := F(p, q) -$

$F(p,p)$, such that that the function acquires the convenient property that $\bar{F}(p,p)=0$, and that the solution to the optimization problem $\min_{q\in\sigma\{\mathcal{X}\}}\mathbb{E}\left[F(p,q)\right]$ does not change. Therefore, in the following propositions, we normalize accordingly and consider $\overline{loss}_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v},v;\mu) := loss_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v},v;\mu) - loss_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v},\widetilde{v};\mu)$ and $\overline{loss}_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z},z;\mu) := loss_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z},z;\mu) - loss_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z},\widetilde{z};\mu)$.

The definition of the Bregman loss function is (Banerjee, Guo, and Wang, 2005):

Definition 1. *Let $\phi : \boldsymbol{R}^d \to \boldsymbol{R}$ be a strictly convex differentiable function, then, the Bregman loss function $D_\phi : \boldsymbol{R}^d \times \boldsymbol{R}^d \to \boldsymbol{R}$ is defined as:*

$$D_\phi(p,q) := \phi(p) - \phi(q) - \langle p-q, \nabla\phi(q)\rangle \tag{23}$$

We consider the simpler case where p,q are scalars. In this case, a Bregman function has the property that its partial second derivative in the first argument is independent of the second argument.

$$\frac{\partial^2 D_\phi(p,q)}{\partial p^2} = \phi''(p) \tag{24}$$

The propositions below rely on this property.

Proposition 2. *Function $\overline{loss}_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z},z;\mu)$ is not in the Bregman class, for all μ.*

Proof. We verify that $\overline{loss}_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z},z;\mu)$ violates the property in Eq. 24.

$$\overline{loss}_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z},z;\mu) = \frac{\mu}{\widetilde{z}}(z-\widetilde{z})^2 \tag{25}$$

$$\frac{\partial^2\overline{loss}_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z},z;\mu)}{\partial\widetilde{z}^2} = \frac{2\mu z^2}{\widetilde{z}^3} \tag{26}$$

It is clear that this is not irrelevant to z. □

Proposition 3. *Function $\overline{loss}_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v},v;\mu)$ is not in the Bregman class, for all μ.*

Proof. Given the property in Eq. 24, a Bregman loss function must be unbounded as $p\to+\infty$. This is because for any fixed q, when $p>q$, $D_\phi(p,q)$ is increasing and convex in p.

However, we verify that $loss_{\text{vv}}^{\text{econ}}(\widetilde{v}, v; \mu)$ is bounded by showing it converges to a finite number as $\widetilde{v} \to +\infty$.

$$\begin{aligned}\overline{loss}_{\text{vv}}^{\text{econ}}(\widetilde{v}, v; \mu) :=& \frac{0.2\exp(-\widetilde{v}) + \mu\left(\exp(-v + \log 0.2 - \log\mu)\right)^2}{\left(1 + \exp(-v + \log 0.2 - \log\mu)\right)^2} \\ &- \frac{0.2\exp(-\widetilde{v}) + \mu\left(\exp(-\widetilde{v} + \log 0.2 - \log\mu)\right)^2}{\left(1 + \exp(-\widetilde{v} + \log 0.2 - \log\mu)\right)^2} \\ =& \frac{(0.2\mu)^2\exp(-\widetilde{v})\left(\exp(\widetilde{v}) - \exp(v)\right)^2}{\left(\mu\exp(\widetilde{v}) + 0.2\right)\left(\mu\exp(v) + 0.2\right)^2} \end{aligned} \tag{27}$$

By L'Hôpital's rule, we have the limit of it as:

$$\begin{aligned}\lim_{\widetilde{v}\to\infty} \overline{loss}_{\text{vv}}^{\text{econ}}(\widetilde{v}, v; \mu) =& \lim_{\widetilde{v}\to\infty} \frac{(0.2\mu)^2\left(\exp(\widetilde{v}) - \exp(2v - \widetilde{v})\right)}{\mu\left(\mu\exp(v) + 0.2\right)^2\exp(\widetilde{v})} \\ =& \frac{0.04\mu}{\left(\mu\exp(v) + 0.2\right)^2} \end{aligned} \tag{28}$$

□

C.4 Further analysis on the loss functions

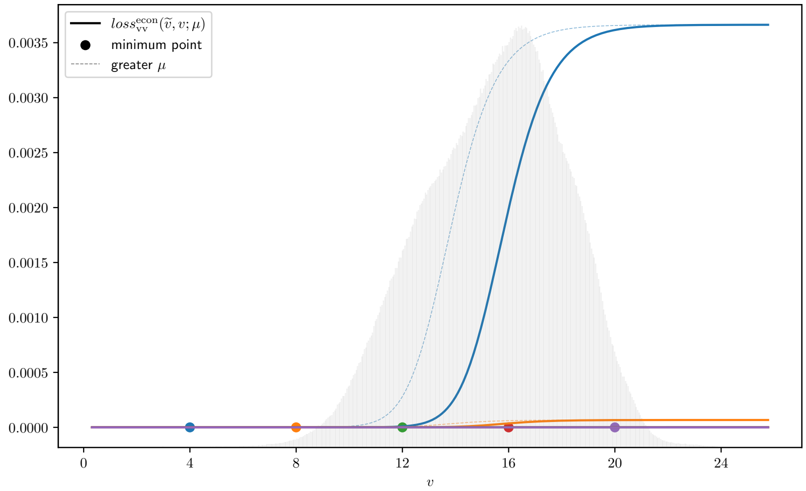
The following is a visualization of the economic loss function in addition to the one in Figure 3 Panel B. The vertical axis is changed to the linear scale from the log scale. For large $\widetilde{v}$ (12, 16, 20, in green, red, purple), the curves are indistinguishable from a flat line because the blue curve is at a much greater magnitude, which is for the loss of overestimating low actual volume ($\widetilde{v} = 4$).

The following proposition formally states the asymmetric property of the loss function for over/under-estimating volume.

Proposition 4. *Consider two symmetrical cases with low and high liquidity $\widetilde{v}_1$ and $\widetilde{v}_2$ such that $\widetilde{z}_1 = 1 - \widetilde{z}_2 < 0.5$. Suppose one makes an overestimation in the low-volume case $\widehat{v}_1 = \widetilde{v}_1 + \varepsilon$, comparing with an equal amount of underestimation in the high-volume case $\widehat{v}_2 = \widetilde{v}_2 - \varepsilon$, the additional loss incurred in the first case is greater than the second:*

$$loss_{\text{vv}}^{\text{econ}}(\widetilde{v}_1, \widehat{v}_1; \mu) - loss_{\text{vv}}^{\text{econ}}(\widetilde{v}_1, \widetilde{v}_1; \mu) > loss_{\text{vv}}^{\text{econ}}(\widetilde{v}_2, \widehat{v}_2; \mu) - loss_{\text{vv}}^{\text{econ}}(\widetilde{v}_2, \widetilde{v}_2; \mu), \quad \forall \mu > 0. \tag{29}$$

Figure C.4: Economic loss function in linear scale



Note: same as Figure 3 Panel B but with the vertical axis in linear scale.

Proof. We first show that, given $\widetilde{z}_2 = 1 - \widetilde{z}_1$, as well as $\widehat{v}_1 - \widetilde{v}_1 = \widetilde{v}_2 - \widehat{v}_2 = \varepsilon$, we have $\widehat{z}_2 = 1 - \widehat{z}_1$, and that $\widetilde{z}_1 - \widehat{z}_1 = \widehat{z}_2 - \widetilde{z}_2$.

We know $s(v;\mu) = \frac{1}{1+\exp(-v+\log 0.2-\log\mu)}$. From $\widetilde{z}_1 = 1 - \widetilde{z}_2$, we have:

$$\begin{aligned}\frac{1}{1+\exp(-\widetilde{v}_2 + \log 0.2 - \log\mu)} &= 1 - \frac{1}{1+\exp(-\widetilde{v}_1 + \log 0.2 - \log\mu)}\\ \exp(-\widetilde{v}_1 - \widetilde{v}_2 + 2\log 0.2 - 2\log\mu) &= 1\\ \widetilde{v}_2 &= 2(\log 0.2 - \log\mu) - \widetilde{v}_1 \qquad (30)\end{aligned}$$

Then we have:

$$\begin{aligned}\widehat{z}_1 + \widehat{z}_2 &= \frac{1}{1+\exp(-\widetilde{v}_1 - \varepsilon + \log 0.2 - \log\mu)} + \frac{1}{1+\exp(-\widetilde{v}_2 + \varepsilon + \log 0.2 - \log\mu)}\\ &= \frac{1}{1+\exp(-\widetilde{v}_1 - \varepsilon + \log 0.2 - \log\mu)} + \frac{1}{1+\exp(\widetilde{v}_1 + \varepsilon - \log 0.2 + \log\mu)} = 1\end{aligned}$$

The last equation comes from the fact that $\frac{1}{1+\exp(x)} + \frac{1}{1+\exp(-x)} = 1$. Then we make use of the loss function expressed in terms of $\widetilde{z}$ and z, as defined in Eq. 21:

$$loss_{\mathrm{zz}}^{\mathrm{econ}}(\widetilde{z}, z; \mu) = \frac{\mu}{\widetilde{z}}(z - \widetilde{z})^2 + \mu(1 - \widetilde{z}) \tag{31}$$

Since $\widetilde{z}_1 - \widehat{z}_1 = \widehat{z}_2 - \widetilde{z}_2$ and $\widetilde{z}_1 < \widetilde{z}_2$, that is $\frac{\mu}{\widetilde{z}_1} > \frac{\mu}{\widetilde{z}_2}$, we have the required result:

$$loss_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v}_1, \widehat{v}_1; \mu) - loss_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v}_1, \widetilde{v}_1; \mu) > loss_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v}_2, \widehat{v}_2; \mu) - loss_{\mathrm{vv}}^{\mathrm{econ}}(\widetilde{v}_2, \widetilde{v}_2; \mu), \quad \forall \mu > 0. \tag{32}$$

□