Compute, Complexity, and the Scaling Laws of Return Predictability

Allan Timmermann
UC San Diego

Luka Vulicevic
*UC San Diego**

February 21, 2026

Abstract

Investors' computing power ("compute") governs how much signal they can extract from high-dimensional data. We show that forecast performance follows scaling laws—stable power-law relationships between accuracy and training compute—that pin down both an irreducible bound on return predictability and the rate at which additional compute improves accuracy. In firm-level cross-sectional return prediction, scaling laws explain over 80% of performance variation across models. Treating compute as a primitive yields sharp economic implications: scaling laws quantify predictability limits and the value of new data, distinguish problems with strong versus weak returns to scale, and deliver a market-efficiency metric by mapping computational advantage into a certainty equivalent. Measured this way, computational superiority earns sizable rents: a 25% increase in compute would have raised an investor's Sharpe ratio by about 10% over the last 30 years, indicating substantial returns to computational sophistication.

Keywords: Machine learning, Asset Pricing, Model Complexity, Return Predictability, Market Efficiency
JEL Classification Numbers: C3, C58, C61, G11, G12, G14

*Contact: atimmermann@ucsd.edu & lvulicevic@ucsd.edu. We thank William Mullins and Joseph Engelberg for valuable comments. The Rady Research Fund and the Brandes Center provided critical funding for this project.

1. Introduction

Processing power ("compute") plays a key role in determining how much information investors can extract from a given information set and, in equilibrium, how much of that information is reflected in prices. Leading quantitative firms increasingly treat compute and data as strategic capital and say so *explicitly*. Renaissance Technologies, for example, advertise a "research database that grows by more than **40 terabytes a day**" and an infrastructure of "**52,000 computer cores**" with "**redundant computational facilities**" on their website. Hudson River Trading makes the same point in different words, emphasizing a deliberately "**high compute-per-researcher ratio**" as core to its mission.[1]

If much of the raw information set is widely available, competitive advantage shifts to the rate and effectiveness with which investors can process high-dimensional signals—making compute a natural primitive. In this paper, we study the limits of return predictability as a function of compute, and we connect these limits to recent advances in forecasting-intensive domains—such as computer vision, meteorology, autonomous driving, and natural language processing—where predictive accuracy improves systematically with model scale and complexity (Kaplan et al. (2020), Henighan et al. (2020)).

A leading explanation for improvements in predictive performance is the "double descent" phenomenon, in which sufficiently large models continue to improve beyond the classical bias–variance turning point (Belkin et al. (2019)). The insight that increasing model scale can raise out-of-sample forecasting performance has reshaped modern machine learning and is often summarized by *scaling laws*. Scaling laws posit stable, approximately power-law relationships between forecast error and computational scale for a given task, characterizing both the rate at which forecasting performance improves with compute and its asymptotic limit. Put differently, while any single model provides *an* estimate of predictability, per-

[1] A sort of arms race can be seen in Table A1 which has more quotes and sources from industry leading hedge funds. They openly advertise their computational resources to attract researchers and often describe their compute as an 'edge'.

formance patterns across a family of models help discipline inference about the underlying data-generating-process.

These developments stand in contrast to prevailing views in economics and finance. Conventional econometric advice cautions against complex models on the grounds of overfitting and weak out-of-sample performance. Empirical practice emphasizes parsimony, favoring low-dimensional specifications and penalizing complexity through information criteria or explicit regularization such as LASSO or ridge regression. This classical paradigm, grounded in a U-shaped bias–variance tradeoff, has also shaped most applications of machine learning in economics and finance.

Figure 1: Modern Machine Learning

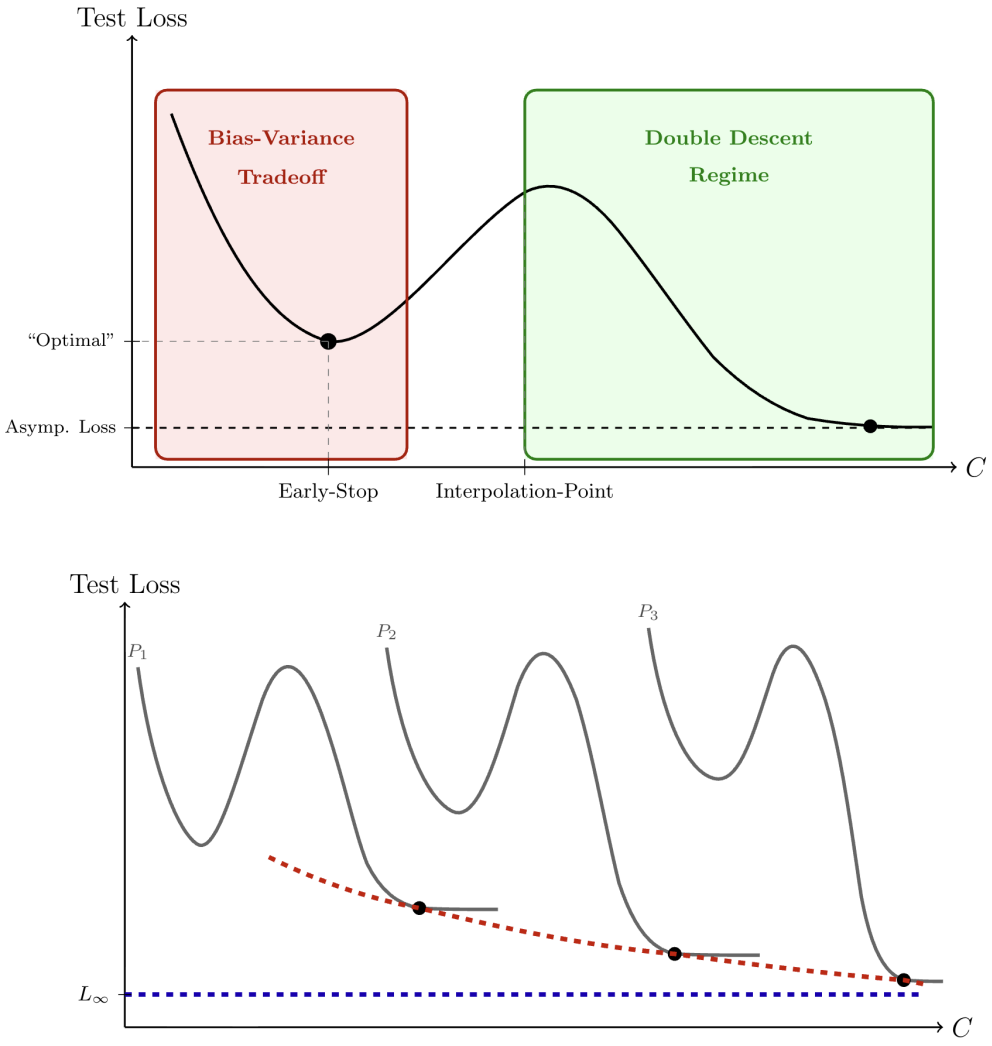


Figure 1 provides a stylized illustration of the double descent phenomenon in compute and its relation to scaling laws, plotting test loss against model compute (C).[2] The red region of the top panel depicts the classical U-shaped bias–variance tradeoff in which test loss initially decreases as models are trained, reaching a minimum before starting to rise as a result of what is often attributed to overfitting. The green region highlights the highly overparameterized double-descent regime in which test loss decreases again beyond the interpolation threshold, approaching the dashed asymptotic loss line. By training many such models with increasing parameter count ($P_1 < P_2 < P_3$), we can then trace out the individual models' asymptotic losses as a function of how much compute was required for the models to enter the double descent regime. The scaling law (red line in the bottom panel) approaches its own asymptote L_∞ which is the irreducible loss (blue dashed line), the theoretical lower bound that no model can surpass regardless of size or compute and therefore represents the limits to predictability or learning for a given forecast problem.[3]

More formally, scaling laws posit a power relation between test loss L as a function of compute c of the form

$$L(c) = L_\infty + \left(\frac{c_0}{c}\right)^\alpha, \tag{1}$$

where $L_\infty > 0$ represents the irreducible loss, $c_0 > 0$ is a scale parameter, and $\alpha > 0$ is the scaling exponent. The power-law representation in (1) yields a compact set of objects that quantify and interpret model complexity. The power law term $(c_0/c)^\alpha$ captures the reducible error that decreases with increasing model capacity. As $c \to \infty$, the model loss approaches L_∞ from above and the exponent (α) governs the rate at which forecasting performance improves with compute. The convex shape indicates large gains at low levels of compute with diminishing improvements thereafter. Larger values of α indicate faster

[2] As Nakkiran et al. (2021) show, double descent exists in a wide variety of models and model primitives like parameters, data, and training time or compute. In this paper we focus on compute (number of floating point operations) required to train a model.

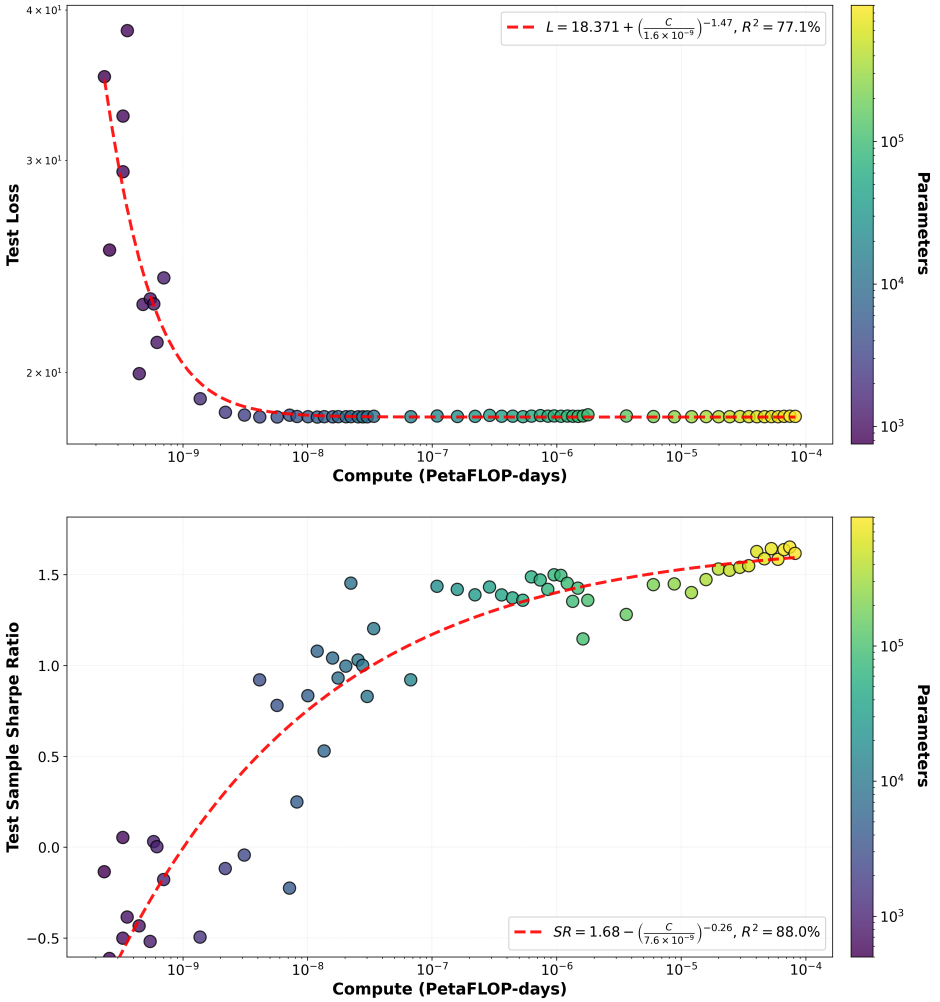
[3] At the interpolation threshold, the model has just enough parameters to perfectly fit the training data, forcing a typically jagged and fragile solution. Moving into the overparameterized regime, many interpolating solutions become feasible, and optimization procedures such as stochastic gradient descent implicitly select smoother, more regularized ones.

improvement with additional parameters, while smaller values suggest a lower improvement rate. The asymptote of the scaling curve approximates the maximal attainable accuracy for a given information set. The irreducible loss L_∞ can be interpreted as the fundamental uncertainty that cannot be reduced by any forecasting model given a particular dataset (list of predictors).

Although scaling laws such as (1) have been discovered in a variety of fields, their existence has not been established for financial return prediction problems. To examine if scaling laws do indeed apply to financial data, we estimate neural networks ranging from 100 parameters to more than five million, and exploit the double-descent regime in which sufficiently large models continue to improve out-of-sample. Using each model's asymptotic loss in compute, we estimate scaling laws that relate return predictability to computational scale (or, equivalently, model size as measured by the number of parameters). We show that scaling-law relationships emerge robustly across training, validation, and test samples, as well as across a range of trading strategies.

Figure 2 provides an illustration of our empirical results. Using the Jensen et al. (2021) characteristics data, the figure plots the root mean squared error (top panel) or sample Sharpe ratio (bottom panel) over a thirty-year test sample (1994–2024). Each circle represents a neural network of a particular parameter count which required some level of compute to be fully trained. Simple models with low compute to the left of the panels tend to produce high and highly volatile test loss and low (even negative) Sharpe ratios. As model compute increases, the test performance of the individual models approaches a clear asymptote and aligns more closely with the scaling law tracked by the red dashed line. This limit, a Sharpe ratio close to 1.7 for the dataset in Figure 2, establishes a bound for how much economically valuable information can be extracted from the underlying dataset, in this case a cross-section of 64 accounting characteristics.

Figure 2: Firm-Level Return Scaling Laws (1994 - 2024)



Note. The top panel plots out-of-sample forecast loss (root mean squared error; RMSE) and the bottom panel plots the out-of-sample Sharpe ratio of a trading strategy constructed from the same forecasts, both as functions of cumulative training compute (in PetaFLOP-days, log scale). Each marker corresponds to a feedforward neural network trained to predict next-month firm-level excess returns using the Jensen et al. (2021) characteristics data, evaluated over a test sample spanning 1994–2024. Marker color indicates model size (number of parameters). Cumulative compute aggregates the floating-point operations used over the full training run for each model, so larger models correspond to higher compute. The dashed red curve in each panel is the fitted scaling law estimated across the family of trained networks; the plotted equation reports the fitted asymptote and scaling exponent, and the associated R^2. In the bottom panel, the Sharpe ratio is computed from a zero-investment long–short portfolio formed each month by sorting firms on predicted returns using a median breakpoint and taking equal-weight positions long (above-median) and short (below-median); Sharpe ratios are computed from monthly portfolio returns and annualized.

The power law in Figure 2 points to two underappreciated benefits of large models in finance. First, large models can be less risky to deploy. Under double descent, sufficiently overparameterized models are implicitly regularized and therefore tend to produce more sta-

ble training outcomes than smaller models, which are more prone to converging to poor local minima (Belkin et al. (2019)). Consistent with this mechanism, our estimated scaling laws exhibit pronounced heteroskedasticity at low levels of compute: some small models achieve performance comparable to models orders of magnitude larger, while others converge to unfavorable minima and perform markedly worse. Second, small models trained with low compute are more likely to perform well in test samples by chance, even over samples as long as that underlying Figure 2. At low compute, test performance is less stable and less representative: models trained with slightly different compute can exhibit meaningfully different results, and performance is more sensitive to hyperparameter choices, complicating replication and interpretation. Both concerns diminish as compute increases and performance stabilizes near the power law asymptote. Much of the machine learning literature in finance—largely framed within the bias–variance paradigm—has operated in this low-compute region, potentially understating both the reliability and economic value of these methods.

Scaling laws carry two implications for asset pricing and market efficiency. First, they identify computing power as central to how information gets embedded into prices. Firms with greater computational resources can more effectively extract, interpret, and forecast from public and private information, impounding more of it into prices. Second, scaling laws explain why leading hedge funds are investing heavily in compute. Just as execution speed—enabled by fiber, microwave, and laser technology—defined high-frequency trading around 2010, processing power now drives investment firms' capital expenditure. Without decreasing returns to additional compute, it would be hard to escape a "winner-takes-all" equilibrium in which a single firm dominates by virtue of superior resources; the concavity of the scaling relationship prevents this. Our results further suggest that this arms race will push firms to invest in unique datasets that can only be fully exploited at scale.

An additional advantage of the scaling-laws approach is its application to testing the informativeness of different datasets. We develop a formal statistical test for the economic

value of augmenting an existing dataset with new data sources by asking whether they shift the scaling law. This reflects a key point: all power-law estimates are conditional on a given information set (list of predictors). If the information set changes in a substantively informative way, the power-law relationship should change as well. By comparing power-law estimates across datasets while holding the forecasting sample fixed, we can therefore assess the economic and statistical value of any additional information.[4]

We organize the remainder of the paper as follows. Section 2 details related literature. Section 3 develops the econometric framework and the estimation of scaling laws. Section 4 describes the data and empirical scaling-law results across firm-level and market-risk-premium prediction problems. Section 5 presents an asset-pricing model that formalizes why heterogeneity in compute can have economic consequences. Section 6 concludes. Details on architecture and training of the neural network are described in Appendix A, while Appendices B and C contain additional empirical results and proofs used in the analysis of the asset pricing model.

2. Related Literature

Modern financial markets are increasingly characterized by heterogeneity in computational capacity. While classic models such as Grossman and Stiglitz (1980), Hellwig (1980), Kyle (1985), Admati (1985), Glosten and Milgrom (1985) emphasize differentials in information sets, today's markets increasingly feature algorithmic traders, high-frequency firms, retail traders, and institutional investors who all differ drastically in their ability to process their information sets.[5] For instance, sophisticated algorithms powered by machine learning can extract signals from data that human traders or simpler algorithms cannot, even when

[4]Note that both the irreducible loss L_∞ as well as the scaling exponent α may change as a result of augmenting an existing dataset with more variables, as new data may make estimation of the conditional mean function more complex but potentially also more accurate.

[5]See also the extensive literature on strategic information acquisition (Van Nieuwerburgh and Veldkamp (2009), Vives (2010)), information diffusion and learning (Veldkamp (2006), Banerjee et al. (2009)), rational inattention (Sims (2003), Kacperczyk et al. (2016), Mondria (2010)), and the more recent focus on machine learning and alternative data (Dugast and Foucault (2018), Farboodi and Veldkamp (2020, 2021)).

both entities observe identical information sets such as earnings announcements. In parallel, access to high-dimensional public information about individual firms and the macroeconomy has expanded through EDGAR, CRSP, Bloomberg, and the St. Louis Fed's FRED. For investors with access to these sources, the central challenge is not data availability, but how to process a high-dimensional vector of public signals. In this setting, heterogeneity in information-processing ability is likely to be equally important as heterogeneity in *private* information.

To address these developments and help interpret our empirical findings, we develop a simple asset pricing model with two groups of investors characterized by having low or high compute, but otherwise being identical. Both groups have to process the information content from a publicly observed signal. We derive three main insights from this model. First, investors with greater computational capacity hold larger and more aggressive positions because they form more precise beliefs from the same data. Second, equilibrium prices reflect the average computational capacity in the market, with more sophisticated markets placing greater weight on public signals. Third, and most importantly for our empirical work, we can measure market efficiency through the marginal benefit of computational improvements: efficient markets exhibit low marginal returns to additional compute, while inefficient markets offer more substantial gains from better processing.

Our work connects to different strands of the finance literature on (i) data-snooping and multiple testing in financial prediction; (ii) economically motivated bounds on attainable performance; and (iii) the virtue of complexity. Foundational warnings about overfitting and specification search include Lo and MacKinlay (1990), Sullivan et al. (1999), White (2000), Hansen (2005), and the factor-zoo view emphasizes that many "discoveries" may not be robust to distortions to inference resulting from multiple comparisons (Harvey et al. (2016)). In parallel, asset pricing theory provides limits on risk-adjusted returns and predictability, including the Hansen–Jagannathan bound (Hansen and Jagannathan (1991)), good-deal bounds on Sharpe ratios (Cochrane and Saa-Requejo (2000)), and explicit upper bounds

on predictive R^2 (Zhou (2010), Huang and Zhou (2017), Potì (2018)). Our scaling-law approach complements these by making the role of computation explicit: conditional on a fixed information set, we estimate how forecasting performance improves with compute and where it ultimately plateaus. Our scaling laws are broadly consistent with recent work in Kelly et al. (2024b)[6] which has challenged the older view of parsimony, arguing that larger models tend to outperform smaller ones using Random Fourier Features ('RFF') in a variety of empirical and theoretical settings.

Our work is also related to the growing empirical literature using neural networks and other high-capacity learners in finance and macroeconomics. Early work argues deep architectures are well-suited for capturing nonlinear structure in portfolios and prediction (Heaton et al. (2017)). In asset pricing, Gu et al. (2020) document out-of-sample gains from flexible ML methods (including neural networks), while Chen et al. (2024) incorporate no-arbitrage restrictions directly into deep learning. Deep learning has also been applied to market microstructure and limit order books (Sirignano and Cont (2021)), and to macro forecasting when nonlinearities are important (Goulet Coulombe et al. (2022)). Finally, our paper speaks directly to the "virtue of complexity" debate (Kelly et al. (2024b), Nagel (2025)): rather than asking only whether bigger models help, we quantify *how* gains scale with compute (the exponent) and *what* remains learnable from the data (the asymptote), across a broad set of financial and economic forecasting problems.

3. Estimation of Scaling Laws

Consider a general forecasting problem which takes the form of a mapping g from a possibly high-dimensional vector of predictors x_t observed at time t to some dependent (target) variable y_{t+h} (such as a financial market return) observed at a future point in time $t+h$, where $h \geq 1$ denotes the forecast horizon. Assuming a squared error loss function, the optimal forecast takes the form of the conditional mean of y_{t+h} given the information x_t,

[6] As well as their follow-on work in Kelly et al. (2022, 2024a), Didisheim et al. (2024).

$g := E[y_{t+h}|x_t]$, and we can write

$$y_{t+h} = g(x_t; \theta_t) + \varepsilon_{t+h}, \qquad \varepsilon_{t+h} \sim (0, \sigma^2). \tag{2}$$

In practice, the functional form of g is unknown and could be highly complex, especially in situations in which x is high-dimensional. Theoretical results by Hornik et al. (1989) suggest that neural nets have universal approximating properties which means that the distance between the true conditional mean function and the best approximating neural net can be made arbitrarily small. However, the best approximating model may be highly complex and require considerable compute to get close to the true conditional mean.

To explore the relationship between compute and predictability of economic outcomes, we organize the remainder of this section as follows. The first three subsections demonstrate how the scaling laws are modeled and estimated. Specifically, we first outline the scaling law for conventional measures of forecast loss such as root mean squared error (RMSE) and define equivalent economic measures such as the Sharpe ratio scaling law. With these objects defined, we cover how we estimate the power law curves. The following subsections cover applied matters in the following order. First, we explain how to estimate market efficiency through a scaling law. Second, we discuss how to compare scaling laws across different forecast problems. Finally, we introduce a statistical test for the value of new input variables and if the information they provide alters the parameters of the scaling law in a meaningful way compared to a null scaling law.

3.1 Loss scaling law

Following the neural scaling law framework in Kaplan et al. (2020), Henighan et al. (2020), we hypothesize that the out-of-sample prediction error follows a power law relationship with compute stated in (1). We estimate the parameters (L_∞, c_0, α) by fitting (1) to the empirical relationship between compute and test loss using nonlinear least squares.

The same power law plus constant functional form is hypothesized for training loss

$L_{\text{train}}(c)$ and validation loss $L_{\text{val}}(c)$, though the specific parameters may differ across these three loss measures. The relationship between these curves provides insight into the bias-variance tradeoff and the degree of overfitting as compute increases.

3.2 Sharpe ratio scaling law

In addition to statistical metrics such as out-of-sample loss or R^2, investors ultimately care about the economic value of forecasts when used in trading strategies. A widely used summary of this value is the unconditional Sharpe ratio of the resulting portfolio. Theoretical results on high-dimensional return prediction emphasize that timing strategies can deliver strictly positive Sharpe ratios even when the associated predictive R^2 is negative, implying that R^2, and therefore also the underlying mean squared forecast error (MSFE), is an incomplete proxy for economic usefulness. Motivated by this insight, we study how the Sharpe ratio scales with compute c across different portfolio constructions, and we refer to the resulting relationships as *Sharpe ratio scaling laws*. We consider two classes of strategies. First, panel-based long–short portfolios that exploit cross-sectional dispersion in model forecasts. Second, a time-series individual asset strategy that takes a dynamic position based on neural network forecasts. In both cases, we map forecasts into portfolios and compute the resulting Sharpe ratios as a function of compute c.

We hypothesize that the Sharpe ratio also follows a scaling law, but with opposite behavior to the loss function. Specifically, we propose

$$\text{SR}(c) = \text{SR}_{\max} - \left(\frac{c_0'}{c}\right)^{\beta}, \tag{3}$$

where $\text{SR}_{\max}$ represents the maximum attainable Sharpe ratio (an upper bound), $c_0' > 0$ is a scale parameter, and $\beta > 0$ is the scaling exponent. As compute increases, $\text{SR}(c)$ approaches $\text{SR}_{\max}$ from below, with the gap decreasing according to the power law. The parameter $\text{SR}_{\max}$ reflects fundamental limits on risk-adjusted returns in the market.

3.3 Portfolio and time-series strategies

To evaluate the economic significance of forecast improvements in return prediction, we construct zero-investment portfolios based on the model predictions. At each time t, we form equal-weighted portfolios based on the decile breakpoints of the forecasted next month returns across all stocks. The return for decile group D at time $t+1$ is computed as

$$r_{D,t+1} = \tfrac{1}{|D_t|} \sum_{i \in D_t} r_{i,t+1}, \tag{4}$$

where the sum is over the D^{th} decile of stocks sorted by the forecasts $y_{i,t+1}$ made at time t. Over the test period $\mathcal{S}$ which spans T time periods, the Sharpe ratio is calculated as

$$\text{SR}_{t \in \mathcal{S}} = \frac{\bar{r}_D}{\sigma(r_D)} = \frac{\frac{1}{T} \sum_{t \in \mathcal{S}} r_{D,t+1}}{\sqrt{\frac{1}{T-1} \sum_{t \in \mathcal{S}} (r_{D,t+1} - \bar{r}_D)^2}}, \tag{5}$$

where $\bar{r}_p$ is the mean portfolio return and $\sigma(r_p)$ is its standard deviation.

Forecasting the return on a single asset such as the aggregate market poses a time-series problem. To this end, let y_{t+1} denote the next-period forecast for the market return. At the end of month t, we take a scaled position in the market portfolio given by

$$w_t = \kappa \, y_{t+1}, \tag{6}$$

where $\kappa > 0$ is a fixed leverage parameter that controls the overall risk of the timing strategy. The realized return on the time-series strategy is

$$r_{\text{portfolio},t+1} = w_t \cdot r_{t+1}. \tag{7}$$

3.4 Estimating scaling laws

We estimate the parameters of the scaling laws for both the MSE loss function and the Sharpe ratio using a separable non-linear least squares approach. Let $\{(c_i, y_i)\}_{i=1}^{M}$ denote the set of observed pairs of compute and the corresponding performance metric (either MSE loss or Sharpe ratio) for M trained models. We fit a generalized power law of the form

$$y_i = \theta_0 + \theta_1 c_i^{\gamma} + \epsilon_i, \quad i = 1, ..., M, \tag{8}$$

where ϵ_i represents the measurement error. This functional form encompasses both the loss scaling law in equation (1), where $\gamma = -\alpha$, and the Sharpe ratio scaling law in equation (3), where $\gamma = -\beta$. Direct estimation of the parameters $(\theta_0, \theta_1, \gamma)$ via standard gradient-based optimization is often numerically unstable due to the high correlation between the asymptotic term θ_0 and the power term. To address this, we employ a variable projection method which exploits the partial linearity of the model. For a fixed exponent γ, the optimal linear parameters $\hat{\theta}_0(\gamma)$ and $\hat{\theta}_1(\gamma)$ have a closed-form solution obtained via weighted linear least squares. The estimation problem thus reduces to a one-dimensional scalar optimization over the exponent γ to minimize the weighted residual sum of squares

$$\min_{\gamma} \sum_{i=1}^{M} w_i \left(y_i - \hat{\theta}_0(\gamma) - \hat{\theta}_1(\gamma) c_i^{\gamma} \right)^2. \tag{9}$$

We employ a log-space weighting scheme with weights defined as

$$w_i = \frac{\log(c_i) - \log(c_{\min}) + 1}{\sum_{j=1}^{M} \left(\log(c_j) - \log(c_{\min}) + 1 \right)}, \tag{10}$$

where $c_{\min} = \min_j c_j$. This weighting scheme is chosen to prioritize the fit of the scaling curve for higher compute levels, which are more indicative of the asymptotic behavior and the irreducible error L_∞ (or the maximum Sharpe ratio $\mathrm{SR}_{\max}$), while avoiding the extreme weight imbalance that arises from linear weighting when compute spans several orders of magnitude.

Constraints are imposed during the optimization to ensure the exponent corresponds to a decaying power law, consistent with the theoretical framework where performance converges to an asymptote as compute increases. Once the optimal exponent $\hat{\gamma}$ is found, the scale parameters c_0 or c_0' are recovered from the linear coefficient $\hat{\theta}_1$.

3.5 Market efficiency

We leverage the Sharpe ratio scaling law in equation (3) to develop a measure of market efficiency that evolves with aggregate investor sophistication. The key insight is that as average computational capacity grows over time, the marginal benefit of additional sophistication diminishes due to the concave nature of the scaling relationship.

Consider an investor with $(1+\delta)$ times the marginal investor's compute (c_t) where $\delta > 0$ represents the investor's relative sophistication. The absolute outperformance relative to the marginal investor is

$$\Delta\mathrm{SR}(\delta; c_t) = \mathrm{SR}((1+\delta)c_t) - \mathrm{SR}(c_t) = \left(\frac{c_0}{c_t}\right)^{\beta} \left[1 - (1+\delta)^{-\beta}\right]. \tag{11}$$

To express this as a percentage improvement, we define the relative outperformance as

$$\frac{\Delta\mathrm{SR}(\delta; c_t)}{\mathrm{SR}(c_t)} = \frac{\left(\frac{c_0}{c_t}\right)^{\beta} \left[1 - (1+\delta)^{-\beta}\right]}{\mathrm{SR}_{\max} - \left(\frac{c_0}{c_t}\right)^{\beta}}. \tag{12}$$

For small δ, a first-order Taylor expansion yields

$$\Delta\mathrm{SR}(\delta; c_t) \approx \beta\delta \left(\frac{c_0}{c_t}\right)^{\beta}, \tag{13}$$

which shows that the absolute gain from a δ increase in compute scales linearly with δ but decays as a power law in aggregate compute, c_t. The percentage improvement in the Sharpe

ratio for an investor with δ more compute is approximately

$$\%\Delta\mathrm{SR}(\delta; c_t) \approx \frac{\beta\delta \left(\frac{c_0}{c_t}\right)^{\beta}}{\mathrm{SR}_{\max} - \left(\frac{c_0}{c_t}\right)^{\beta}} \times 100. \tag{14}$$

As $c_t \to \infty$, $\Delta\mathrm{SR}(\delta; c_t) \to 0$ and $\%\Delta\mathrm{SR}(\delta; c_t) \to 0$ for any fixed relative advantage δ. This implies that the maximum alpha achievable by a proportionally more sophisticated investor shrinks as aggregate market sophistication increases. The market thus becomes increasingly efficient—not because predictability vanishes, but because the marginal returns to additional sophistication diminish as the marginal investor's capacity approaches the asymptotic regime of the scaling law.

One challenge with this approach is estimating the marginal investor's compute c_t, which is unobservable. To address this, we introduce the concept of market-implied compute. Given that we have estimated the structural parameters of the Sharpe ratio scaling law $(\mathrm{SR}_{\max}, c_0, \beta)$ and we can empirically estimate the Sharpe ratio of the aggregate market portfolio, denoted $\mathrm{SR}_{\mathrm{mkt}}$, we can invert the scaling function to solve for the compute level consistent with market performance. Setting $\mathrm{SR}(c_t) = \mathrm{SR}_{\mathrm{mkt}}$ in equation (3) yields the implied compute

$$c_t^{\mathrm{implied}} = c_0 \left(\mathrm{SR}_{\max} - \mathrm{SR}_{\mathrm{mkt}}\right)^{-1/\beta}. \tag{15}$$

This implied compute serves as a revealed preference proxy for the sophistication of the average investor conditional on the information set used to estimate SR_{max}. By substituting c_t^{implied} back into equation (14), we can quantify the market's efficiency in terms of the diminishing returns to computational scale. If the implied compute is high, the market is positioned on the flat portion of the scaling curve, meaning that even substantial increases in relative sophistication δ yield negligible improvements in risk-adjusted returns.

3.6 Testing for the significance of new information

If the econometrician is uncomfortable asserting the functional form of $g(\cdot)$, then she may also be uncomfortable with her choice of input data x_t. In this subsection we will show how to conduct hypothesis testing on the value of the addition of one or more new input variables. In short, we will require the estimation of a scaling law on the data set x_t and on some x'_t. With these estimated, we can formally test if the scaling law given by x'_t differs from that generated by the null x_t. Critically, this difference can be either in the irreducible loss, the exponent parameter of the scaling law, or in both.

To evaluate whether an additional predictor belongs in the true model, we formalize a comparison between two fitted scaling laws. Let $L(c; \boldsymbol{\theta})$ denote the baseline scaling law estimated from d input features in x_t, with parameters $\boldsymbol{\theta} = (L_\infty, c_0, \alpha)$ obtained via nonlinear least squares over a range of compute levels $\{c_1, c_2, \ldots, c_K\}$. Similarly, let $L^*(c; \boldsymbol{\theta}^*)$ denote the augmented scaling law estimated from $d+1$ features, with parameters $\boldsymbol{\theta}^* = (L^*_\infty, c^*_0, \alpha^*)$.

Our hypothesis test concerns whether the augmented feature set yields a statistically distinguishable reduction in irreducible loss and potentially a different scaling exponent. Formally, we test the null hypothesis:

$$H_0 : L_\infty = L^*_\infty, \quad \text{and} \quad \alpha = \alpha^*, \tag{16}$$

against the alternative that at least one parameter differs significantly. Under H_0, the additional variable provides no information about returns beyond what is already captured by the baseline features.

Since the two models may be non-nested—one cannot be obtained from the other through parameter restrictions—standard likelihood ratio tests for nested models are not applicable. Following Vuong (1989), we employ a likelihood-based approach suitable for comparing non-nested models. We construct a test statistic based on the sum of squared residuals from each fitted power law. Let SSR and SSR^* denote the residual sum of squares for the baseline and

augmented models, respectively, computed as:

$$\text{SSR} = \sum_{k=1}^{K} \left[L_{\text{test}}(c_k) - L(c_k; \hat{\boldsymbol{\theta}}) \right]^2, \quad \text{SSR}^* = \sum_{k=1}^{K} \left[L_{\text{test}}(c_k) - L^*(c_k; \hat{\boldsymbol{\theta}}^*) \right]^2, \tag{17}$$

where $L_{\text{test}}(c_k)$ is the observed test loss at compute level c_k. We then compute the log-likelihood ratio assuming Gaussian errors:

$$\text{LR} = K \log \left(\frac{\text{SSR}}{\text{SSR}^*} \right). \tag{18}$$

If LR is sufficiently large, we reject H_0 and conclude that the additional variable meaningfully improves the scaling law. To establish statistical significance, we use a bootstrap procedure: we resample the empirical scaling curve data with replacement, refit both power laws to each bootstrap sample, and compute the distribution of LR under resampling. The p-value is then the proportion of bootstrap samples where $\text{LR}_{\text{bootstrap}} \geq \text{LR}_{\text{observed}}$. This approach accounts for estimation uncertainty in both $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ without requiring strong distributional assumptions.

4. Empirical results

This section provides details of the data, followed by the main empirical results from applying scaling laws to our data, a detailed look into trading strategy scaling laws and, finally, an examination of market efficiency.

4.1 Data

We study scaling laws in the context of two canonical forecasting problems: (i) predicting firm-level cross-sectional stock returns; and (ii) predicting aggregate market returns.

4.1.1 Cross-sectional data

For cross-sectional return prediction, we use the firm characteristics datasets from Freyberger et al. (2020) (FNW-36) and Jensen et al. (2021) (GFD).

The FNW sample covers 1,404,048 firm-month observations from July 1962 through May 2014, encompassing 12,813 unique firms across 623 months with an average of 2,254 firms per month. We employ 36 characteristics as predictors spanning valuation ratios (book-to-market, earnings-to-price), size (log market equity), momentum (12-minus-2 month returns, short-term reversal), risk (CAPM beta, idiosyncratic volatility), and profitability measures (ROA, asset growth). The target variable is firm-level monthly excess stock returns. We split the data sample reserving 20% of monthly observations for testing and 12.5% of the remaining dates for validation. This yields a training sample from July 1962 to September 1998 (821,804 observations), a validation sample from October 1998 to December 2003 (223,335 observations), and a test sample from January 2004 to May 2014 (358,909 observations).

Similarly, we use the GFD data in two different settings. First, we directly compare this dataset to FNW by sampling down to the same PERMNO-month observations across the two datasets. This allows us to directly compare the value of the additional features GFD provides. From the available features in Jensen et al. (2021) we select 114 by removing any features with more than 20% missing observations.

The second use of the GFD dataset is to estimate scaling laws for the 30-year out-of-sample period from 1994 to 2024. In this second setting, we keep all US exchange traded securities and once again enforce a less than 20% missing observation feature selection rule which leaves 64 features. This sample is comprised of 4,252,826 observations with just under 30,000 firms. Temporally, the data is split into 40% in the training set, 10% in the validation set, and 50% in the test sample.

4.1.2 Market data

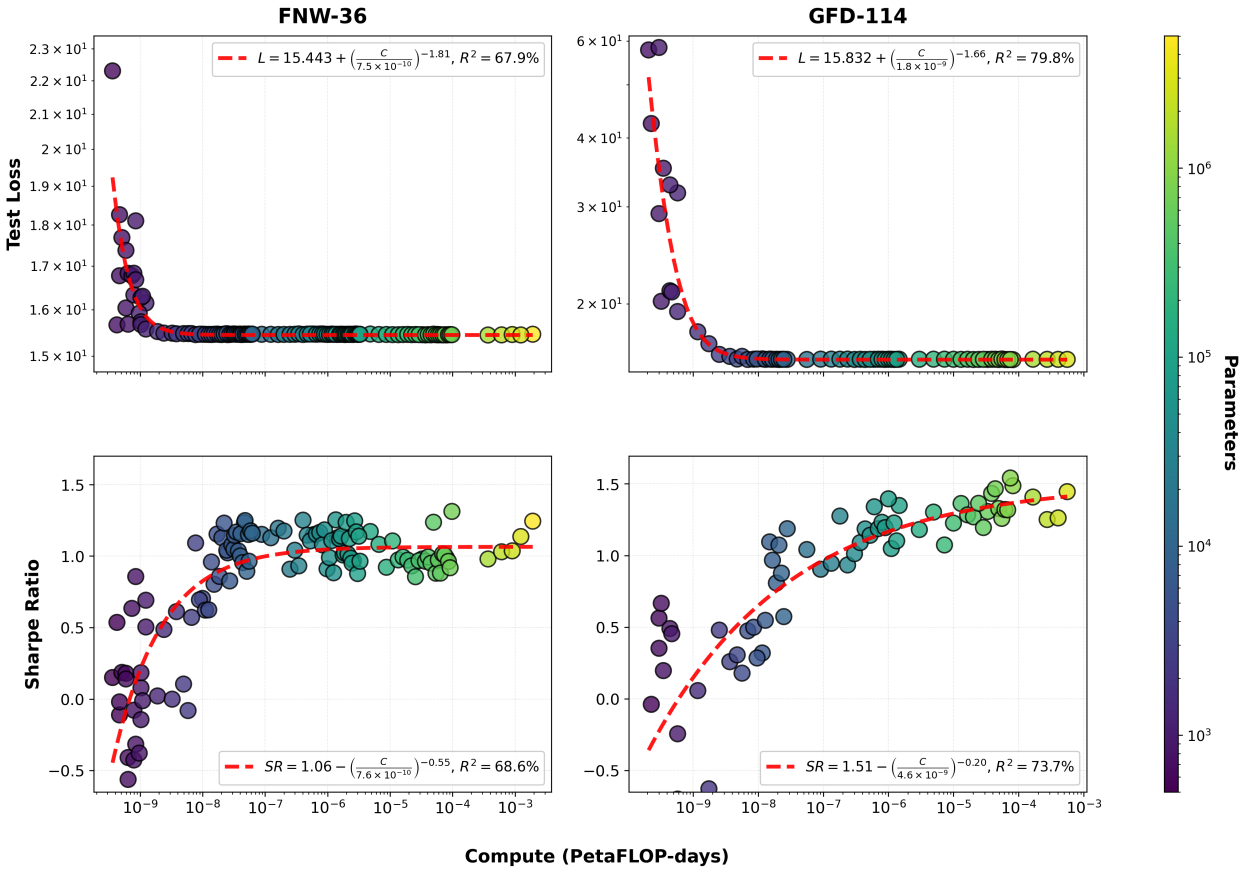
For market return prediction, we use the predictor variables from Welch and Goyal (2008). After merging with Fama-French data, our sample contains 1,176 monthly observations from January 1927 through December 2024. We use 14 predictors including valuation ratios (dividend-price, earnings-price), interest rates (Treasury bill, long-term yield), spreads (term spread, default spread), stock variance, and inflation. Given the smaller sample size, we allocate 35% of dates to testing and 15% of the remaining dates to validation. This yields a training sample from January 1927 to January 1981 (649 months), a validation sample from February 1981 to August 1990 (115 months), and a test sample from September 1990 to December 2024 (412 months).

4.2 Scaling laws: Cross-sectional results

To examine if firm-level return prediction models follow scaling laws, we estimate neural network models with an increasing number of parameters.[7] Since we employ double descent in the training of the models, this requires more compute for larger parameterized models. We vary the number of parameters from 500 to just over five million. This translates to a range from 10^{-10} to 10^{-3} PetaFLOP-days of compute. For each trained model, we obtain the associated test loss which we plot against the compute required to train the model in Figure 3. Each circle in the figure represents one neural network with a particular parameter count which is colored from deep purple (smallest models) to yellow (largest).

[7]Details on how these models are constructed, trained, and evaluated are in the Appendix.

Figure 3: Scaling Laws of Firm-Level Return Prediction



Note. This figure depicts the fitted scaling law for return prediction in the Freyberger et al. (2020) dataset. The first row shows the test sample loss, and the second row shows test sample Sharpe ratios. The first column shows models fit with the original 36 Freyberger et al. (2020) ('FNW-36') firm-level characteristics, the second column represents scaling laws for neural networks trained on 114 features from Jensen et al. (2021). The training, validation, and test samples are selected as the intersection of permno-month observations between the two datasets. The dashed red curves are the scaling laws which relate compute (measured in PetaFLOP-days) to forecast accuracy. Loss is root mean squared error and Sharpe ratios are annualized. All models are trained with data from 1962-07 to 1998-09, validated on data from 1998-10 to 2003-12 and finally the above results are generated using data from 2004-01 to 2014-05.

Figure 3 summarizes the performance of a broad set of neural networks trained for return prediction based on two different information sets. Specifically, the left column shows firm-level next-month return prediction using the Freyberger et al. (2020) 36 characteristics (FNW-36) while the right column shows the equivalent result for the Global Factor Data from Jensen et al. (2021) subset to the same firm-month observations. The first row reports test-sample loss (root mean squared error) as a function of training compute. The second

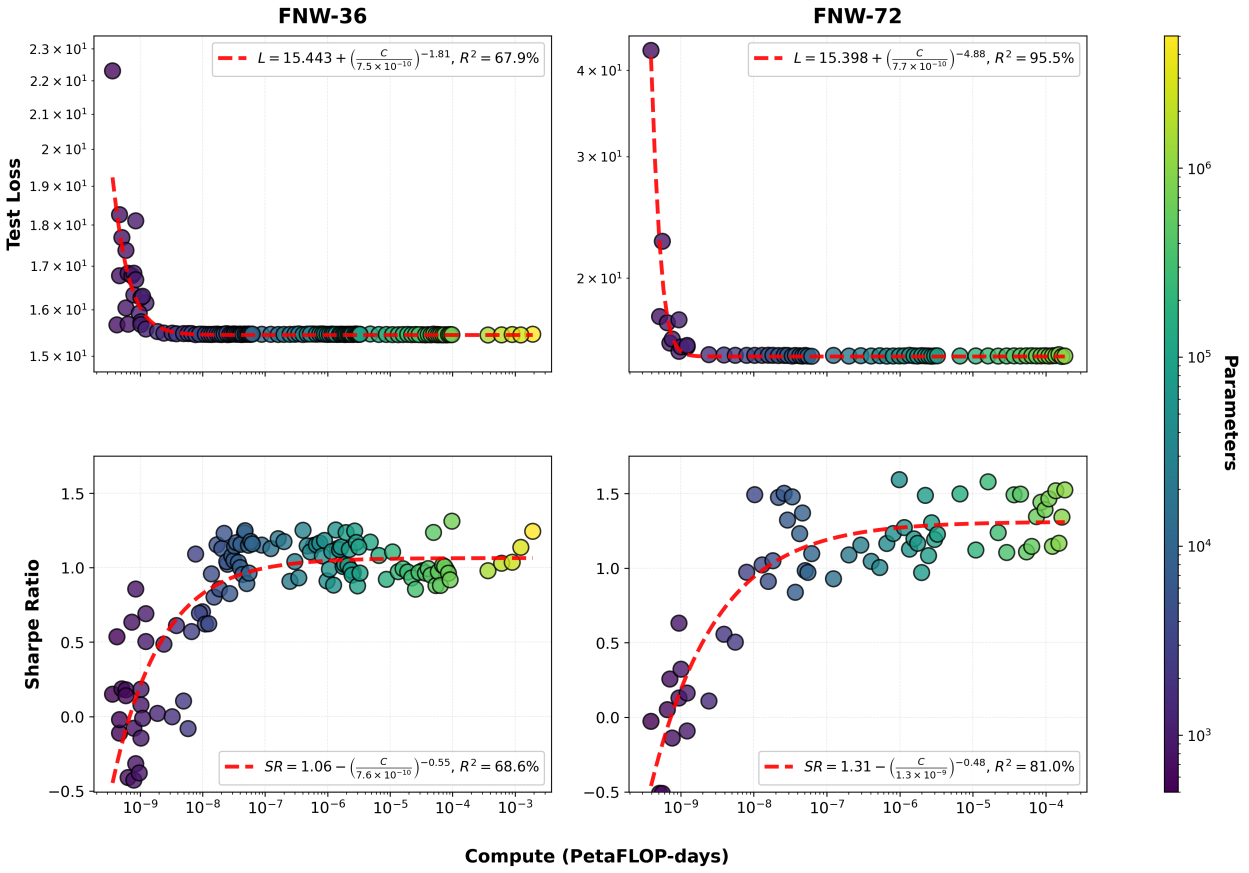
row reports test-sample annualized Sharpe ratios from trading strategies formed on the neural networks' predictions, also plotted against training compute. Dashed red curves show fitted scaling laws relating performance to compute.

The power law does not fit the data perfectly, particularly at low levels of compute towards the left side of the plots. However, based on 36 firm-level characteristics it achieves an R^2 of 68% under both the RMSE loss and Sharpe ratio. This fit improves for the larger information set with 114 variables to 79.8% and 73.7% for the RMSE loss and Sharpe ratio settings, respectively.

Under the statistical loss (RMSE, upper panel), the asymptote of the power law is reached relatively quickly, at levels of compute close to 10^{-9}. At low or medium levels of compute, we observe some cases with very high levels of test loss. Conversely, we never see very high levels of test loss at sufficiently high levels of compute. The risk of ending up with a model with a high test loss is therefore comparatively lower under the highest levels of compute. In this sense, models estimated with high levels of compute are "safer" than models estimated with low levels of compute. This is consistent with the finding from the loss contour plots that smaller models have a higher risk of getting stuck in local minima than larger models.

The power law for the Sharpe ratio shows how the models that use the smallest amount of compute tend to perform quite poorly. Many of the models based on levels of compute at or below 10^{-9} PetaFLOP-days produce negative Sharpe ratios that vary between -0.5 and 0.8, highlighting again the risk of ending up with a poorly-performing forecasting model at low levels of compute. Once compute is raised above 10^{-7} PetaFLOP-days, we no longer see cases with low Sharpe ratios which stabilize above 0.8, asymptoting just above 1 and ranging as high as 1.40 for the FNW-36 data.

Figure 4: Value of Imposing Priors on Data



Note. This figure depicts the fitted scaling law for return prediction in the Freyberger et al. (2020) dataset. The first row shows the test sample loss, and the second row shows test sample Sharpe ratios. The first column shows models fit with the original 36 Freyberger et al. (2020) ('FNW-36') firm-level characteristics, the second columns adds the point in time cross-sectional z-score of each characteristic ('FNW-72'). The dashed red curves are the scaling laws which relate compute (measured in PetaFLOP-days) to forecast accuracy. Loss is root mean squared error and Sharpe ratios are annualized. All models are trained with data from 1962-07 to 1998-09, validated on data from 1998-10 to 2003-12 and finally the above results are generated using data from 2004-01 to 2014-05.

Figure 4 shows the increase in performance one can get by taking the FNW-36 characteristics and including cross-sectional z-scores of the features to get a total of 72 features. By including this new information to the dataset, neural networks are able to scale to higher Sharpe ratios as indicated by the asymptotic Sharpe ratio of 1.31 for the FNW-72 dataset.

Table 1: Scaling Laws of Finance

Forecast	Input		Validation	Test	Sharpe
Firm Ret.	FNW-36	Scaling Law	$22.0408 + \left(\frac{C}{6.7\times10^{-10}}\right)^{-1.8805}$	$15.4435 + \left(\frac{C}{7.5\times10^{-10}}\right)^{-1.8134}$	$1.0646 - \left(\frac{C}{7.6\times10^{-10}}\right)^{-0.5548}$
		R^2	70.61	67.94	68.60
Firm Ret.	FNW-72	Scaling Law	$22.0541 + \left(\frac{C}{7.2\times10^{-10}}\right)^{-4.8793}$	$15.3985 + \left(\frac{C}{7.7\times10^{-10}}\right)^{-4.8793}$	$1.3115 - \left(\frac{C}{1.3\times10^{-9}}\right)^{-0.4823}$
		R^2	95.50	95.52	80.96
Firm Ret.	GFD-114	Scaling Law	$17.1276 + \left(\frac{C}{1.9\times10^{-9}}\right)^{-1.5701}$	$15.8318 + \left(\frac{C}{1.8\times10^{-9}}\right)^{-1.6554}$	$1.5069 - \left(\frac{C}{4.6\times10^{-9}}\right)^{-0.2009}$
		R^2	78.09	79.77	73.66
MKT-Rf	GW-14	Scaling Law	$4.7942 + \left(\frac{C}{2.5\times10^{-13}}\right)^{-1.0547}$	$4.3757 + \left(\frac{C}{2.5\times10^{-13}}\right)^{-1.1391}$	$0.6425 - \left(\frac{C}{1.4\times10^{-13}}\right)^{-0.8065}$
		R^2	92.42	94.70	77.10

Note. This table reports estimated scaling laws for financial forecasting models. Each scaling law takes the form $L_\infty + (C/C_0)^{-\alpha}$, where L_∞ is the irreducible loss, C denotes model compute, C_0 is a scale parameter, and α governs the rate of performance improvement with model size. The first column indicates the forecast target: firm-level monthly excess returns (Firm Ret.) and the market risk premium (MKT-Rf). The second column denotes the predictor feature set. We report the fitted scaling law and its R^2 (in percent) across three metrics: validation sample RMSE, test sample RMSE, and out-of-sample Sharpe ratio.

Table 1 shows that the two estimated Sharpe ratio power laws for the small and large information sets of Freyberger et al. (2020) are $1.0646-(C/(7.6\times10^{-10}))^{-0.5548}$ and $1.3115-\left(\frac{C}{1.3\times10^{-9}}\right)^{-0.4823}$, respectively. For investors with very large amounts of compute (large C), adopting the larger information set seems to be the best strategy since the Sharpe ratio asymptotes at a larger value (1.31) than for the smaller information set (1.06). Conversely, for investors with small computing resources, it is not worthwhile to use the larger information set. The amount of compute at which the two curves cross equals $C \approx 1.51\times10^{-9}$. For values of C above this point, the larger information set provides a higher Sharpe ratio whereas for values of C below this cutoff, the smaller information set produces the highest Sharpe ratio.

Table 2: Hypothesis Tests for Scaling Law Parameters

		Validation		Test		Sharpe Ratio	
Comparison		L_∞	α	L_∞	α	S_∞	α
	FNW-36	22.0408	−1.8805	15.4435	−1.8129	1.0646	−0.5548
(1)	FNW-72	22.0541	−4.8793	15.3985	−4.8793	1.3115	−0.4823
	Diff	0.0134	−2.9988	-0.0450***	−3.0664	0.2469***	0.0725
		(0.0180)	(1.7699)	(0.0270)	(1.6130)	(0.0424)	(0.1211)
	FNW-36	22.0408	−1.8805	15.4435	−1.8129	1.0646	−0.5548
(2)	GFD-114	17.1276	−1.5701	15.8318	−1.6554	1.5069	−0.2009
	Diff	-4.9131***	0.3105	0.3883***	0.1575	0.4422***	0.3539***
		(0.0136)	(0.9979)	(0.0130)	(0.9486)	(0.0765)	(0.0660)
	FNW-72	22.0541	−4.8793	15.3985	−4.8793	1.3115	−0.4823
(3)	GFD-114	17.1276	−1.5701	15.8318	−1.6554	1.5069	−0.2009
	Diff	-4.9265***	3.3093	0.4333***	3.2239	0.1954**	0.2814***
		(0.0222)	(1.6312)	(0.0297)	(1.5141)	(0.0960)	(0.2103)

Note. Pairwise differences in scaling law parameters. Scaling laws follow $L_\infty + (C/C_0)^{-\alpha}$ for loss and $S_\infty - (C/C_0)^{-\alpha}$ for Sharpe ratios, where L_∞ and S_∞ denote asymptotic performance and α the convergence rate. Standard errors in parentheses. $^{*}p < 0.10$, $^{**}p < 0.05$, $^{***}p < 0.01$.

Table 2 reports L_∞ and α parameters of the loss function along with differences across the various datasets and standard errors for these and indications of statistical significance based on our scaling-law difference test. We begin by assessing whether augmenting the baseline FNW-36 characteristics with cross-sectional z-scores (FNW-72) materially shifts

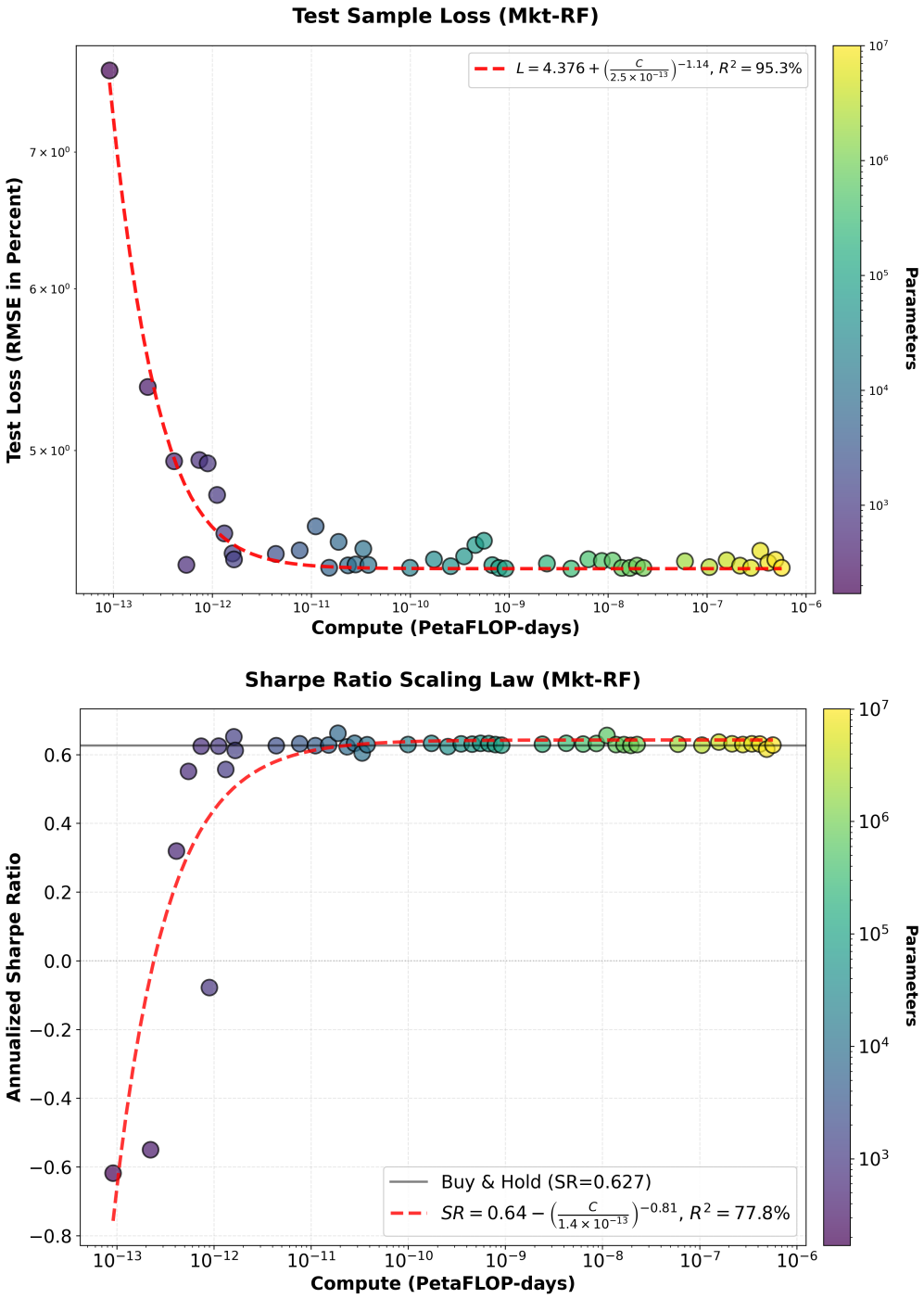
the Sharpe ratio scaling law. The estimated asymptotic Sharpe ratio increases from 1.0646 (FNW-36) to 1.3115 (FNW-72), a difference of 0.2469 (SE = 0.0424; $t = 5.82$), and we reject equality at conventional levels (95% CI: [0.1721, 0.3430]). By contrast, we fail to reject equality of the scaling exponents: $\hat{b}_{36} = -0.5548$ versus $\hat{b}_{72} = -0.4823$ (difference 0.0725; SE = 0.1211; $t = 0.60$). Thus, the additional cross-sectional information primarily raises the attainable frontier rather than changing the rate at which performance improves with compute. More broadly, the result illustrates how expanding the information set can alter inferences about the data-generating process: for firm-level next-month return prediction, both firms' characteristics and their relative position in the cross-section appear to matter.

For the larger dataset of Jensen et al. (2021) displayed in Figure 2, the estimated Sharpe ratio scaling law $1.5069 - (C/(4.6 \times 10^{-9}))^{-0.2009}$ suggests an even higher asymptote for the Sharpe ratio and a slower convergence than for the smaller data sets over the same period. In this case, both the asymptotic bound on the Sharpe ratio (1.5069) as well as the scaling coefficient (-0.2009) are significantly different from both the small and medium FNW-36, FNW-72 data sets.

In summary, comparing the Sharpe ratio scaling laws across the small, medium, and large information sets, we find that investors with small compute resources are better off using the smaller information sets and only investors with relatively high amounts of compute can expect to benefit from access to larger information sets.

4.3 Scaling laws: Market risk premia

Figure 5: Scaling Laws of Risk Premia Prediction



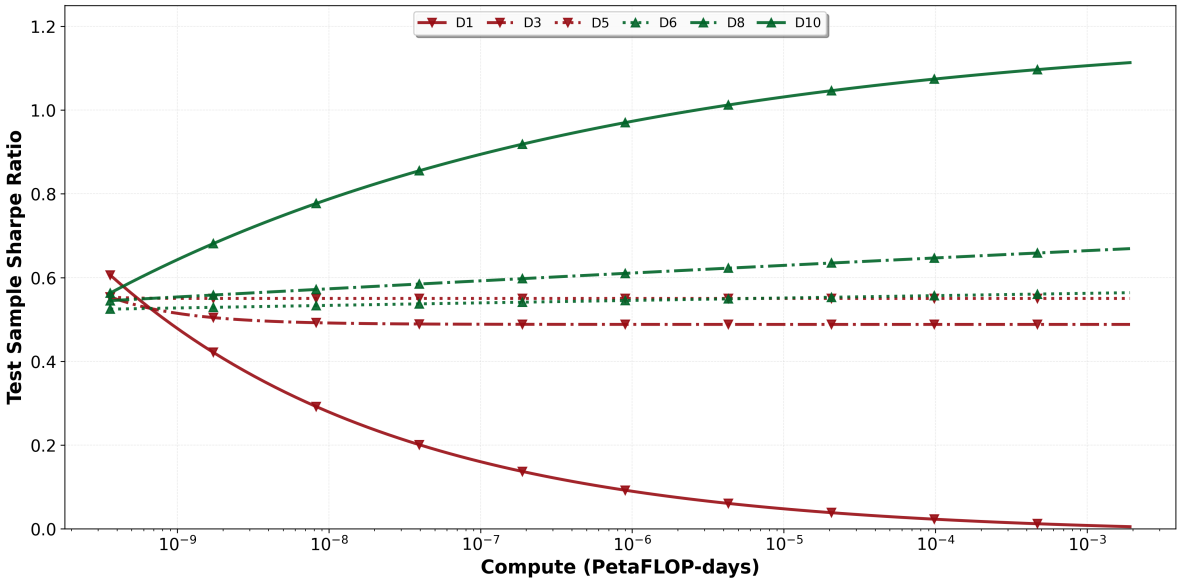
Note. This figure plots test sample RMSE and trading strategy Sharpe ratio against cumulative compute (PetaFLOP-days) for risk premia prediction using the 14 Welch and Goyal (2008) predictors.

In Figure 5 and Table 1, we provide results for market risk premia prediction using the original predictor set of Welch and Goyal (2008). We find well-fit scaling laws linking forecast performance to compute. While the asset-pricing literature has typically treated market and firm-level return prediction as largely separate problems, placing them side by side within a common scaling-law framework allows a direct comparison of their effective data-generating processes. A first implication is that risk premia prediction appears substantially easier in the low-compute regime. In our estimates, compute on the order of 10^{-12} is sufficient to reach the Sharpe-ratio asymptote of 0.6 for the market problem and the RMSE asymptote is reached at a similarly low level of compute. By contrast, the firm-level prediction task remains far from its asymptote at that scale; in that setting, the Sharpe-ratio frontier is reached only around 10^{-6}.

4.4 Firm-level return trading strategy

We next examine firm-level trading-strategy scaling laws. In particular, we decompose the strategy to identify from where the Sharpe ratio scaling law originates and which components of the trading rule drive the relationship between performance and compute.

Figure 6: Sharpe Ratio Scaling Laws by Decile



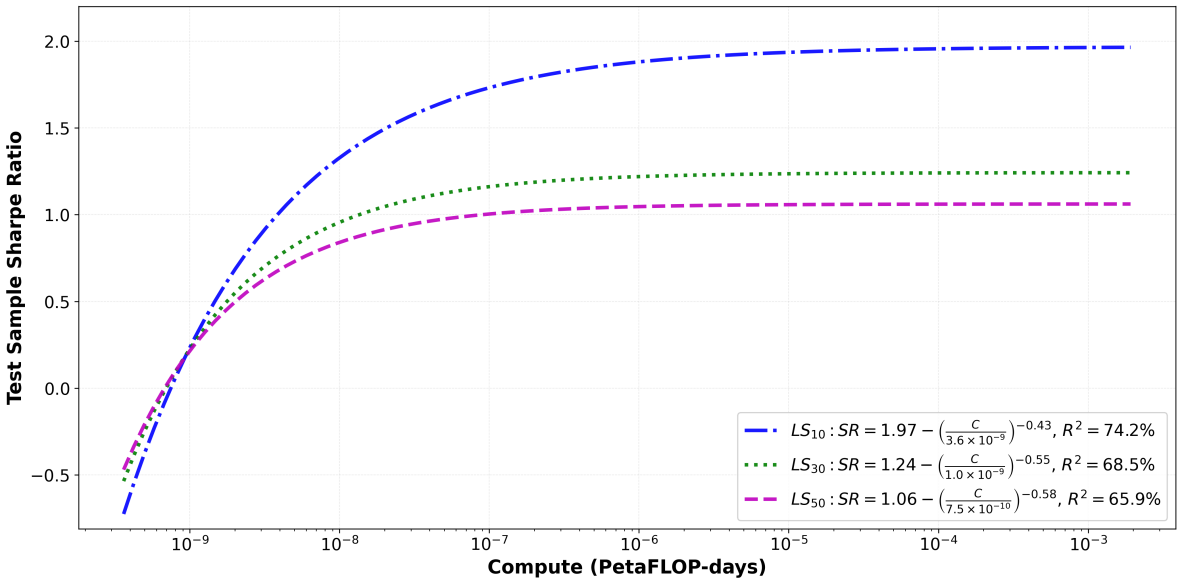
Note. This figure plots the annualized Sharpe ratios of the underlying decile portfolios from Figure 7.

Figure 6 shows the equivalent relation between compute and test sample Sharpe ratios for deciles of stocks sorted by expected returns with decile 1 containing those stocks with the lowest (often most negative) predicted return, decile 2 containing the stocks with the second-lowest expected returns, and decile 10 containing the stocks with the highest expected returns. We identify a clearly monotonically increasing (for stocks with the highest expected returns) or decreasing (for stocks with the lowest expected returns) compute-Sharpe ratio relation with essentially flat curves for the middle deciles.[8] This is consistent with additional compute improving forecast accuracy and portfolio performance only for the subset of stocks with a large enough predictable return component that places them in the top or bottom deciles. Stocks for which the NN models do not identify a strong predictive component are likely to have return expectations close to their unconditional mean, and so tend to be assigned to the middle portfolios as ranked by expected returns.

Figure 7 plots Sharpe ratio scaling laws by breakpoints. Specifically, the figure plots the annualized Sharpe ratios of long-short portfolios formed each month by sorting stocks into deciles by the model's forecast, then computing portfolio-level returns from going long in one or more high-ranked deciles and shorting a set of low-ranked deciles. For example, LS_{10} goes long in decile 10 and shorts decile 1, while LS_{50} goes long in the top half of deciles (i.e., stocks in deciles 6-10 with above-median expected returns) and shorts stocks in the bottom half of deciles (1-5). The power law of the stocks in the tails of the expected return distribution (deciles 1 and 10) starts at a lower point but asymptotes at a higher level (SR of 1.97) than stocks in the center of the distribution (SR asymptote of 1.06). Consistent with our earlier findings, investors with low compute would have been better off holding a broad portfolio of stocks rather than investing in the stocks ranked in the tails of the expected return distribution.

[8]Table A2 reports parameter estimates for these decile portfolios. For the top two deciles with the highest expected returns, the scaling law explains 79% and 60% of the compute-test-RMSE relation while for the bottom two deciles with the lowest expected returns, we obtain R^2 values of 66% and 39%, respectively.

Figure 7: Sharpe Ratio Scaling Laws by Breakpoint



Note. This figure reports scaling laws for annualized Sharpe ratios of long-short portfolios formed from Freyberger et al. (2020)-36 firm-level return predictions. Each month, stocks are sorted into deciles by the model's forecast and portfolio returns are constructed by going long a set of upper deciles and short a set of lower deciles. For example, LS_{10} is long decile 10 and short decile 1, while LS_{50} is long the top half of deciles (6–10) and short the bottom half (1–5). Curves show fitted Sharpe scaling with compute/model size for each breakpoint specification.

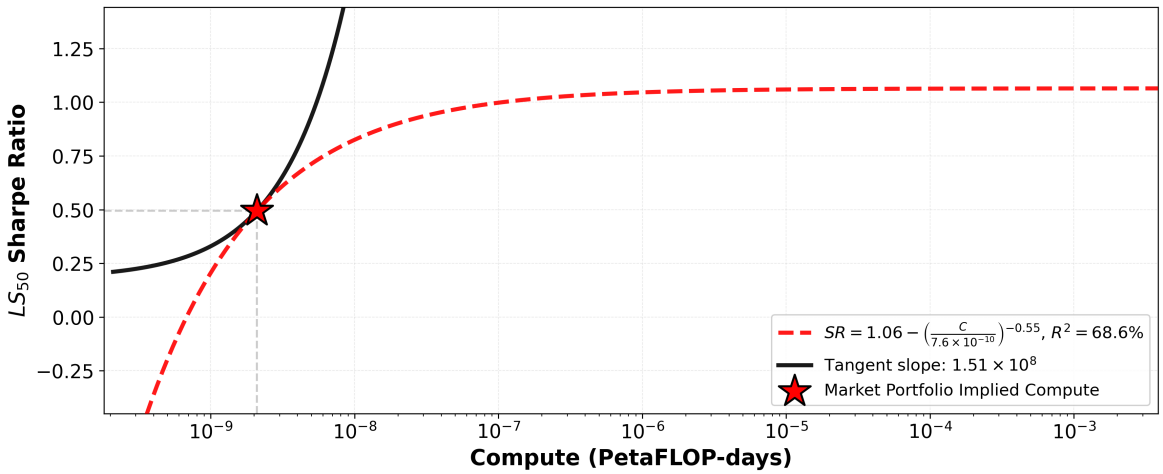
4.5 Market efficiency

Section 4 uses a simple asset pricing model to show that the tangent slope of the scaling curve evaluated at the implied compute level can be used as a market efficiency metric:

$$\left.\frac{\partial \mathrm{SR}}{\partial C}\right|_{C=C_t^{\text{implied}}} \tag{19}$$

This derivative captures the marginal return to additional compute at the market's current sophistication level. A high tangent slope indicates that the market operates on the steep portion of the scaling curve where incremental compute yields substantial alpha, consistent with a more inefficient market. Conversely, a low tangent slope implies that the market has reached the asymptotically flat region, where even large increases in compute produce negligible improvements in risk-adjusted returns.

Figure 8: Market Efficiency Example (2004-2014)



Note. This figure reports an example of how to estimate market efficiency from the scaling laws made with the 36 firm characteristics from Freyberger et al. (2020). The market buy and hold Sharpe ratio is denoted as a red star (Sharpe ratio is calculated from 2004 to 2014), which implies a level of market average compute. We then use this implied market compute to find the tangency line of the scaling law (black solid curve in log-linear space).

Figure 8 illustrates this construction for the 2004–2014 estimation window using the LS_{50} strategy. The red dashed curve depicts the estimated scaling law $\text{SR} = 1.06 - (C/7.6 \times 10^{-10})^{-0.55}$, which achieves an R^2 of 68.6%. The red star marks the market portfolio's implied compute of approximately 3×10^{-9} PetaFLOP-days, corresponding to a Sharpe ratio of roughly 0.5. The black tangent line, with slope 1.51×10^8, visualizes the marginal return to compute at this point. The steep tangent indicates that during this period, the market remained on the concave portion of the curve where additional sophistication could still generate meaningful outperformance. At the tangency point, a 25% increase in compute would result in a 10% increase in Sharpe ratio, implying rents to computational superiority at the margin.

4.6 Loss landscape visualization

An implicit benefit of greater model complexity is improved training stability. In practice, larger networks are less sensitive to hyperparameter choices and exhibit smoother, more reli-

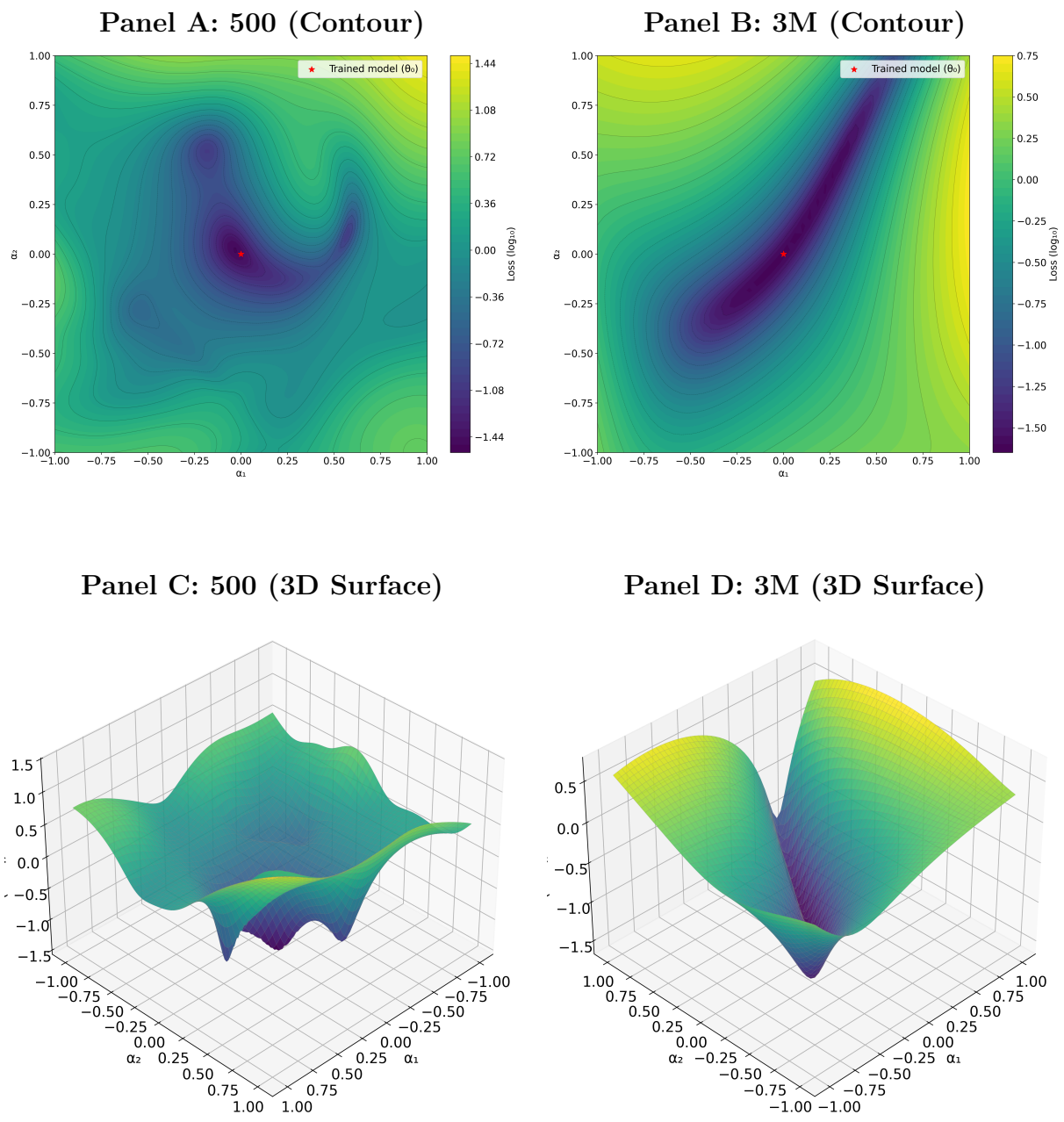
able optimization dynamics. A natural explanation is geometric: as model size increases, the effective loss landscape in the vicinity of the learned solution becomes implicitly regularized and easier to navigate with first-order methods.

To characterize the local optimization geometry around trained models, we visualize the loss landscape in a two-dimensional affine subspace through the trained parameter vector θ_0. We sample two random Gaussian direction vectors d_1 and d_2 in parameter space and rescale them with two perturbation parameters (α_1, α_2). When a model is trained, it learns parameters θ_0 though a loss landscape. We can then visualize the loss landscape with the following perturbation: $\theta(\alpha_1, \alpha_2) = \theta_0 + \alpha_1 d_1 + \alpha_2 d_2$ for (α_1, α_2) ranging over a grid. With each grid value of (α_1, α_2) we can re-compute mean squared error on the test sample.

The resulting loss surface is visualized as both a contour map and a three-dimensional surface, with the origin corresponding to the trained model. These visualizations reveal the local curvature of the loss landscape, the presence or absence of nearby local minima, and the sharpness of the optimum, all of which provide insight into why larger models exhibit more stable performance in the double-descent regime.

In particular, Figure 9 plots the loss landscape ($log_{10}MSE$) for 500- and 3-million parameter firm-level return models in a two-dimensional random subspace around the trained parameter θ_0. Comparing the contour and 3D surfaces of the two models, we see that the small model has more local minima in which the model can get trapped. Conversely, the loss surface for the large model is more of a continuous ridge which makes identification of the global minimum easier via gradient descent.

Figure 9: Firm-Level Return Loss Landscapes Across Model Size



Note. Panels A and C show, respectively, a contour map and 3D surface of the loss landscape ($\log_{10}$ MSE) for the 500-parameter firm-level return model using the FNW-36 characteristics in a two-dimensional random subspace around the trained parameters θ_0. Panels B and D show the analogous plots for the 3M-parameter firm-level return model. The red star marks the trained model at $(\alpha_1, \alpha_2) = (0, 0)$.

4.7 Evolution in computing power over time

Our final empirical result speaks to why investors may differ in computational capacity. Having shown a strong relationship between compute and asset-pricing outcomes, we now provide evidence on the sources of compute heterogeneity to motivate the theory that follows.

A natural source of heterogeneity in computational capacity arises from the cost of maintaining cutting-edge hardware. Moore's Law implies that computational power has grown exponentially over time, but acquiring state-of-the-art processors entails substantial capital expenditure. The marginal benefit of superior forecasting ability must therefore be weighed against the marginal cost of continuous hardware upgrades.

Higher computational capacity c reduces forecast error and increases the precision of posterior beliefs. However, cutting-edge hardware commands a price premium. An investor unwilling to bear these costs will instead operate hardware that lags the frontier by some interval $\Delta > 0$. If the cutting-edge investor at time t employs capacity c_t^*, the lagging investor operates at $c_t' = c_{t-\Delta}^*$. The gap $c_t^* - c_t'$ thus reflects the economic trade-off between forecasting accuracy and hardware costs.

Figure A3 illustrates this heterogeneity using historical data on processor transistor counts obtained from Wikipedia's Transistor Count tables. For each year, we identify the maximum transistor count among all listed CPUs, representing cutting-edge technology. The "Marginal Compute Investor" series reports the frontier from ten years prior.[9] The shaded region represents the computational gap between investor types, which widens from negligible levels in the early 1980s to several orders of magnitude by 1990 and remains steady ever since, providing empirical grounding for heterogeneity in c.

Crucially, this hardware-induced heterogeneity generates forecast performance differences even when all investors observe identical public information. The cutting-edge investor, positioned higher on the scaling law, achieves lower forecast error and higher Sharpe ratios.

[9]Long lags like 10 years can also represent investors upgrading each year but to lower-end hardware which will, by Moore's Law, operate at the level of older cutting-edge hardware.

Whether this improvement justifies the cost of frontier hardware depends on the investor's scale and the competitiveness of the market.

5. Compute and Asset Pricing

In this section, we develop a highly stylized asset pricing model to formalize the heterogeneity in investors' access to compute and explore its implications for market equilibrium. Our framework builds on the CARA-normal approach of Grossman and Stiglitz (1980), but replaces information asymmetry with heterogeneity in processing power. Our analysis suggests a new interpretation of existing methods: computational capacity affects how effectively investors extract information from public signals through Bayesian updating. This generates economically meaningful heterogeneity even when all investors observe identical (public) information signals.

The model delivers three main insights. First, investors with greater computational capacity hold larger and more aggressive positions because they form more precise beliefs from the same data. Second, equilibrium prices reflect the average computational capacity in the market, with more sophisticated markets placing greater weight on public signals. Third, we can measure market efficiency through the marginal benefit of computational improvements: efficient markets exhibit low marginal returns to additional compute, while inefficient markets offer more substantial gains from better processing.

5.1 Model setup

We consider a single-period economy with a continuum of risk-averse investors who trade a risky asset and a risk-free asset. A fraction $\lambda \in (0, 1)$ of investors possesses high computational capacity c^*, while the remaining fraction $1 - \lambda$ has lower computational capacity c', where $c^* > c' > 0$. All investors have CARA utility $U(W) = -\exp(-AW)$ with common risk aversion $A > 0$ and identical initial wealth W_0.

The risky asset has liquidation value $\tilde{\theta} \sim \mathcal{N}(\bar{\theta}, \tau_0^{-1})$ and trades at price P. The risk-free

asset has a zero normalized return. An investor demanding x shares achieves terminal wealth $W = W_0 + x(\tilde{\theta} - P)$. The asset supply is noisy: $\tilde{z} = \bar{z} + \tilde{u}$ where $\tilde{u} \sim \mathcal{N}(0, \tau_u^{-1})$.

All investors observe the same public signal $s = \tilde{\theta} + \tilde{\eta}$, where $\tilde{\eta} \sim \mathcal{N}(0, \tau_s^{-1})$ is independent noise. The critical assumption is that computational capacity c determines the effective precision with which investors process this signal. Specifically, we model the relationship between computational capacity and signal processing effectiveness through a *compute transformation function* $f : \mathbb{R}_+ \to \mathbb{R}_+$. This function $f(c)$ maps the raw computational capacity c of an investor to the effective multiplier on signal precision. We assume f is continuously differentiable with $f(0) = 0$, $f'(c) > 0$ for all $c > 0$, and $\lim_{c\to\infty} f(c) = \infty$. These conditions ensure that compute has no benefit without capacity, that more compute always helps, and that arbitrarily precise signal extraction is achievable in principle.

An investor with capacity c forms posterior beliefs through Bayesian updating with effective signal precision $f(c)\tau_s$, yielding posterior precision

$$\tau(c) = \tau_0 + f(c)\tau_s \tag{20}$$

and posterior mean

$$\mu(c, s) = \frac{\tau_0\bar{\theta} + f(c)\tau_s s}{\tau_0 + f(c)\tau_s}. \tag{21}$$

Higher computational capacity thus generates two advantages: more precise beliefs (higher $\tau(c)$) and greater responsiveness to the signal (higher weight on s in $\mu(c, s)$). This formulation captures the idea that sophisticated algorithms (higher c) extract more information from the same data. The generality of the transformation function $f(c)$ allows us to characterize what functional forms are consistent with the empirically observed relationships between compute and forecasting performance.

5.2 Equilibrium characterization

Standard CARA-normal analysis yields optimal demands that are proportional to expected excess returns scaled by risk tolerance as we summarize in the first Proposition:

Proposition 1 (Optimal Demands)**.** *An investor with computational capacity c demands*

$$x(c) = \frac{1}{A}\left[\tau_0\bar{\theta} + f(c)\tau_s s - P(\tau_0 + f(c)\tau_s)\right]. \tag{22}$$

Define the population-weighted average of the transformed computational capacity as $\bar{f} \equiv \lambda f(c^*) + (1-\lambda) f(c')$ and aggregate precision as $\bar{\tau} \equiv \tau_0 + \tau_s \bar{f}$. Market clearing determines the equilibrium price:

Proposition 2 (Equilibrium Price)**.** *The equilibrium price is*

$$P = \frac{\tau_0\bar{\theta} + \tau_s \bar{f} s}{\bar{\tau}} - \frac{A(\bar{z} + \tilde{u})}{\bar{\tau}}. \tag{23}$$

The price is a precision-weighted average of the prior mean and signal, where the weight on the signal increases with average market computational effectiveness $\bar{f}$. The risk premium decreases with $\bar{f}$ because higher aggregate precision reduces perceived risk. Importantly, even though the price perfectly reveals s when supply is deterministic, heterogeneity persists because investors process the signal with different effectiveness.

Substituting the equilibrium price into demands reveals the trading structure:

Proposition 3 (Equilibrium Demands)**.** *In equilibrium, demands decompose as*

$$x^* = \frac{1}{A\bar{\tau}}\left[\tau_0\tau_s(f(c^*) - \bar{f})(s - \bar{\theta}) + A(\bar{z} + \tilde{u})(\tau_0 + f(c^*)\tau_s)\right] \tag{24}$$

and

$$x' = \frac{1}{A\bar{\tau}}\left[\tau_0\tau_s(f(c') - \bar{f})(s - \bar{\theta}) + A(\bar{z} + \tilde{u})(\tau_0 + f(c')\tau_s)\right]. \tag{25}$$

High-computational-capacity investors ($f(c^*) > \bar{f}$) take larger positions in the risky asset when the signal exceeds the prior mean ($s > \bar{\theta}$), while low-capacity investors ($f(c') < \bar{f}$) take smaller positions. Both types absorb shares of the aggregate supply proportional to their precision. This trading pattern holds even when the price fully reveals the signal, distinguishing our mechanism from traditional information asymmetry models.

5.3 Measuring market efficiency through computational capacity

In models with information asymmetry, efficiency is typically measured by how quickly prices incorporate private information. In our model where information is public, efficiency instead relates to how well the market processes available information.

We formalize this through the marginal benefit of computational capacity. An investor's certainty equivalent is $CE(c) = W_0 + \frac{1}{2A}[\mu(c,s) - P]^2\tau(c)$, and we define $MB(c) \equiv \partial CE(c)/\partial c$ as the marginal benefit of compute. The following lemma establishes a useful property.

Lemma 3.1. *An investor with computational capacity $\bar{c}$ such that $f(\bar{c}) = \bar{f}$ has posterior beliefs $\mu(\bar{c}, s) = P + \frac{A(\bar{z}+\tilde{u})}{\bar{\tau}}$, differing from the price only by the risk premium.*

This lemma requires a clarification. In general, $\bar{f} = \lambda f(c^*) + (1-\lambda) f(c') \neq f(\bar{c})$, where $\bar{c} = \lambda c^* + (1-\lambda)c'$, since f may be nonlinear. However, for any market configuration, we can define a representative computational capacity $\tilde{c}$ implicitly by $f(\tilde{c}) = \bar{f}$. By the properties of f (continuous, strictly increasing, with range $[0, \infty)$), such a $\tilde{c}$ exists and is unique. The lemma then states that an investor operating at this representative capacity level has beliefs that differ from the price only by the risk premium.

This implies that $\mu(\tilde{c}, s) - P = \frac{A(\bar{z}+\tilde{u})}{\bar{\tau}}$ does not depend on the realization of s. Consequently, while the realized marginal benefit $MB(\tilde{c})$ can depend on s when $\tau_0 > 0$, its unconditional expectation $E[MB(\tilde{c})]$ depends only on market primitives and the function f.

Proposition 4 (Marginal Benefit at Market Average)**.** *Consider an investor operating at computational capacity $\tilde{c}$ defined by $f(\tilde{c}) = \bar{f}$. The expected marginal benefit of computational*

capacity at this level is

$$E[MB(\tilde{c})] = \frac{f'(\tilde{c})\tau_s A\, E[(\bar{z}+\tilde{u})^2]}{2(\tau_0+\tau_s\bar{f})^2} = \frac{f'(\tilde{c})\tau_s A(\bar{z}^2+\tau_u^{-1})}{2(\tau_0+\tau_s\bar{f})^2}, \tag{26}$$

where the expectation is taken over $(s, \tilde{u})$.

When $f(c) = c$, we have $f'(c) = 1$ and $\bar{f} = \bar{c}$. The key insight is that the marginal benefit now depends on $f'(\tilde{c})$, the slope of the compute transformation function at the representative capacity level. This derivative captures how effectively additional computational resources translate into improved signal processing at the margin.

To connect our model to the empirically observed power law relationship between compute and forecasting performance, we now characterize the *Compute Efficient Frontier*, defined as the relationship between computational capacity and the expected marginal benefit of additional compute. Our empirical analysis reveals that this frontier exhibits power law decay: on a log-log scale, the marginal benefit decreases linearly with compute. Along the compute efficient frontier, we index markets by a representative computational capacity c defined implicitly by $f(c) = \bar{f}$ and evaluate the expected marginal benefit at this representative capacity. We now give a functional form of $f(c)$ that is sufficient to generate this power law structure.

Proposition 5 (Power Law Compute Efficient Frontier)**.** *Suppose that the compute transformation function takes the form*

$$f(c) = \kappa c^{\alpha}, \tag{27}$$

for constants $\kappa > 0$ *and* $\alpha \in (0, 1)$*. Then* $E[MB(c)] \propto c^{-\gamma}$ *with* $\gamma = 1 + \alpha$*. The relationship holds exactly when* $\tau_0 = 0$ *and asymptotically when* $\tau_s \kappa c^{\alpha} \gg \tau_0$*.*

This result provides the theoretical foundation for our empirical analysis. The power law form $f(c) = \kappa c^{\alpha}$ implies that the effectiveness of signal processing exhibits diminishing returns to scale in computational capacity. The parameter α governs the curvature as smaller

values indicate more severe diminishing returns. Within the power-law family $f(c) = \kappa c^{\alpha}$, the compute efficient frontier exponent satisfies $\gamma = 1 + \alpha$ and hence $\alpha = \gamma - 1$.

To understand the economic content of this result, consider the derivative of the expected marginal benefit with respect to compute. Under the power law specification $f(c) = \kappa c^{\alpha}$ with $\tau_0 = 0$, we have, for all $c > 0$,

$$\frac{\partial E[MB(c)]}{\partial c} = -\frac{(1+\alpha)\alpha A(\bar{z}^2 + \tau_u^{-1})}{2\kappa\tau_s} c^{-(2+\alpha)} < 0 \tag{28}$$

This confirms that the compute efficient frontier is downward sloping: early investments in compute yield large gains, but improvements shrink as markets become sophisticated.

The power law structure has important implications for cross-sectional variation in market efficiency. Markets with higher average computational capacity lie further along the efficient frontier and exhibit smaller marginal returns to additional compute. Empirically, this implies that log-log plots of marginal benefit versus computational capacity should exhibit negative slopes with magnitude $\gamma = 1 + \alpha$. Since $\alpha \in (0, 1)$ ensures diminishing returns in the transformation function, the power law exponent satisfies $\gamma \in (1, 2)$.

Definition (Computational Efficiency)**.** *Market computational efficiency is given by*

$$\mathcal{E} \equiv \frac{(\tau_0 + \tau_s \bar{f})^2}{f'(\tilde{c})\tau_s}, \tag{29}$$

where $\tilde{c}$ satisfies $f(\tilde{c}) = \bar{f}$.

This measure generalizes the efficiency concept to accommodate the nonlinear compute transformation. Under the power law specification $f(c) = \kappa c^{\alpha}$, we have $f'(\tilde{c}) = \alpha\kappa\tilde{c}^{\alpha-1}$. Since $f(\tilde{c}) = \kappa\tilde{c}^{\alpha} = \bar{f}$, we can write $\tilde{c} = (\bar{f}/\kappa)^{1/\alpha}$ and therefore

$$f'(\tilde{c}) = \alpha\kappa \left(\frac{\bar{f}}{\kappa}\right)^{(\alpha-1)/\alpha} = \alpha \bar{f}^{(\alpha-1)/\alpha} \kappa^{1/\alpha}. \tag{30}$$

The efficiency measure is inversely proportional to the marginal benefit: $E[MB(\tilde{c})] =$

$\frac{A(\bar{z}^2+\tau_u^{-1})}{2\mathcal{E}}$. Markets with high $\mathcal{E}$ are efficient in the sense that additional computational improvements provide minimal benefits. Markets with low $\mathcal{E}$ are inefficient because substantial gains remain from better information processing.

Proposition 6 (Properties of Computational Efficiency)**.** *Suppose f is twice continuously differentiable on $(0,\infty)$ with $f'(c) > 0$ and $f''(c) \leq 0$ for all $c > 0$. Then market computational efficiency $\mathcal{E}$*

(i) is strictly increasing in $\bar{f}$;

(ii) approaches infinity as $\bar{f} \to \infty$ (perfect efficiency); and

(iii) approaches $\tau_0^2/(f'(0^+)\tau_s)$ as $\bar{f} \to 0$ (minimal efficiency), where $f'(0^+) = \lim_{c\to 0^+} f'(c)$.

Under the power law specification with $\alpha \in (0,1)$, we have $f'(c) = \alpha\kappa c^{\alpha-1} \to \infty$ as $c \to 0^+$. This implies that the minimal efficiency level $\mathcal{E}_{\min} = \tau_0^2/(f'(0^+)\tau_s) = 0$. Intuitively, when the market has vanishingly small computational capacity, even tiny improvements in compute yield unboundedly large benefits, resulting in zero efficiency. This limiting behavior is consistent with the empirical observation that the largest gains from compute accrue at the lowest capacity levels.

5.4 Economic implications

Our model provides several insights consistent with our earlier empirical analysis. First, computational heterogeneity generates economically meaningful trading differences even when all information is public (Proposition 3). In Grossman and Stiglitz (1980), uninformed investors would be indifferent between trading strategies if prices were fully revealing. By contrast, our low-compute investors optimally choose less aggressive positions because they cannot process information as effectively.

Second, the model highlights a fundamental difference between information frictions and processing frictions. In traditional asymmetric information models, market efficiency depends on the extent to which prices aggregate dispersed private information. In our framework, efficiency depends on the market's collective ability to extract information from public

data. This distinction is crucial for modern markets where most price-relevant information is publicly available but requires sophisticated processing.

Third, the efficiency measure $\mathcal{E}$ provides an operational approach to assessing market quality. Rather than asking whether prices incorporate information (the traditional measure), we address how much value remains in additional computational improvements. This measure can be estimated empirically through the returns to algorithmic sophistication as demonstrated in the empirical analysis.

Fourth, and most importantly for our empirical work, the power law structure of the compute efficient frontier provides a direct link between theoretical primitives and observable quantities. Within the power-law family $f(c) = \kappa c^{\alpha}$, the empirically estimated frontier exponent γ maps to the curvature parameter $\alpha = \gamma - 1$ governing the compute transformation function. This connection allows us to test the model's predictions and to quantify the economic value of computational investments across different market environments.

6. Conclusion

We examine the role of compute for return predictability. Our analysis shows that the findings across a broad range of sciences that scaling (power) laws dictate the relation between compute (model complexity) and forecasting performance carry over to predictability of the cross-section of stock returns. Our analysis measures complexity using compute rather than the raw number of parameters, which is not commensurable across model classes.

We find that using low amounts of compute (e.g., simple neural network models) exposes the modeler to the risk of ending up in a local optimum. Using larger amounts of compute largely resolves this issue, at least for the dimensions of the types of information sets considered in our analysis. At the opposite end of the compute spectrum, the estimated power scaling law allows us to bound the limits to return predictability along with the rate at which additional compute facilitates further gains in economic or statistical forecasting performance for a given data set.

The steepness of the power law informs us about the marginal value of additional compute. We propose a test of how this depends on the underlying information set and show empirically that agents with low compute resources are better off with smaller information sets than agents with large compute resources.

Our results highlight a tension between modern machine learning approaches versus the tightly parameterized canonical asset pricing models. In traditional asset pricing and econometrics the capital asset pricing model (CAPM) is often described as having "one parameter": the loading β on the market portfolio. This description obscures the substantial structure embedded in the model. The CAPM imposes a fully specified functional form, a particular choice of state variables, and a collection of strong identifying assumptions—linearity, exogeneity, homoskedasticity, and serially uncorrelated disturbances—that together underpin the Gauss–Markov theorem. The effective complexity of the CAPM is therefore not captured by the dimension of its parameter vector, but by this bundle of theoretical restrictions.[10] Put differently, the "low parameter count" of the CAPM reflects that much of the economic and statistical structure has been imposed ex ante, but is then suppressed when complexity is proxied by parameter counts.

Modern deep learning models, by contrast, impose relatively little ex ante structure on the data-generating process. A deep neural network defines a highly flexible function class, subject only to weak regularity conditions, as formalized by universal approximation results such as Hornik et al. (1989). The network does not presume a linear factor structure, nor does it impose the orthogonality and homoskedasticity conditions that underlie Gauss–Markov. Instead, it expends computational resources to search over a high-dimensional function space and learn a representation that approximates the conditional expectation. Consequently, even if the true market model were exactly the CAPM, a generic neural network would not recover it with a one-dimensional parameter vector: the parsimony of the CAPM derives from restrictions imposed a priori, whereas in deep learning analogous structure must be

[10]This perspective connects to a long-standing distinction between the data-modeling and algorithmic-modeling cultures in statistics Breiman (2001).

learned from the data.

We view this proliferation of parameters not as a defect, but as the cost of relaxing strong model-specification assumptions. In this sense, complexity substitutes for the parametric structure that is hard-coded in the much "simpler" canonical asset pricing models. Modern machine learning methods replace the assumption of a correctly specified asset pricing model with an acknowledgment that the true model specification is unknown but that with sufficient compute we can approximate it arbitrarily well. Because this trade-off is mediated primarily through computational effort rather than nominal parameter counts (Belkin et al. (2019), Kaplan et al. (2020), Henighan et al. (2020)), compute is the natural primitive for characterizing scaling laws in financial forecasting.

Our analysis shows that the advantage associated with additional compute is closely linked with the underlying dataset used by investors as both the asymptotic gain from additional compute along with the rate at which such compute benefits investors varies significantly across different data sets. This suggests that investors should consider the information acquisition and compute decisions jointly in order to optimally exploit their resources. We leave formalizing this joint information and compute allocation problem for future work.

References

Admati, Anat R, 1985, A noisy rational expectations equilibrium for multi-asset securities markets, *Econometrica: Journal of the Econometric Society* 629–657.

Banerjee, Snehal, Ron Kaniel, and Ilan Kremer, 2009, Price drift as an outcome of differences in higher-order beliefs, *The Review of Financial Studies* 22, 3707–3734.

Belkin, Mikhail, Daniel Hsu, Siyuan Ma, and Soumik Mandal, 2019, Reconciling modern machine-learning practice and the classical bias–variance trade-off, *Proceedings of the National Academy of Sciences* 116, 15849–15854.

Breiman, Leo, 2001, Statistical modeling: The two cultures (with comments and a rejoinder by the author), *Statistical science* 16, 199–231.

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

Cochrane, John H, and Jesus Saa-Requejo, 2000, Beyond arbitrage: Good-deal asset price bounds in incomplete markets, *Journal of political economy* 108, 79–119.

Didisheim, Antoine, Shikun Barry Ke, Bryan T Kelly, and Semyon Malamud, 2024, Apt or "aipt"? the surprising dominance of large factor models, Technical report, National Bureau of Economic Research.

Dugast, Jérôme, and Thierry Foucault, 2018, Data abundance and asset price informativeness, *Journal of Financial economics* 130, 367–391.

Farboodi, Maryam, and Laura Veldkamp, 2020, Long-run growth of financial data technology, *American Economic Review* 110, 2485–2523.

Farboodi, Maryam, and Laura Veldkamp, 2021, A model of the data economy, Technical report, National Bureau of Economic Research Cambridge, MA, USA.

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

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

Goulet Coulombe, Philippe, Maxime Leroux, Dalibor Stevanovic, and Stéphane Surprenant, 2022, How is machine learning useful for macroeconomic forecasting?, *Journal of Applied Econometrics* 37, 920–964.

Grossman, Sanford J, and Joseph E Stiglitz, 1980, On the impossibility of informationally efficient markets, *The American economic review* 70, 393–408.

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

Hansen, Lars Peter, and Ravi Jagannathan, 1991, Implications of security market data for models of dynamic economies, *Journal of political economy* 99, 225–262.

Hansen, Peter Reinhard, 2005, A test for superior predictive ability, *Journal of Business & Economic Statistics* 23, 365–380.

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

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

Hellwig, Martin F, 1980, On the aggregation of information in competitive markets, *Journal of economic theory* 22, 477–498.

Henighan, Tom, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al., 2020, Scaling laws for autoregressive generative modeling, *arXiv preprint arXiv:2010.14701* .

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White, 1989, Multilayer feedforward networks are universal approximators, *Neural networks* 2, 359–366.

Huang, Dashan, and Guofu Zhou, 2017, Upper bounds on return predictability, *Journal of Financial and Quantitative Analysis* 52, 401–425.

Jensen, Theis Ingerslev, Bryan T Kelly, and Lasse Heje Pedersen, 2021, Is there a replication crisis in finance?, Technical report, National Bureau of Economic Research.

Kacperczyk, Marcin, Stijn Van Nieuwerburgh, and Laura Veldkamp, 2016, A rational theory of mutual funds' attention allocation, *Econometrica* 84, 571–626.

Kaplan, Jared, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei, 2020, Scaling laws for neural language models, *arXiv preprint arXiv:2001.08361* .

Kelly, Bryan, Boris Kuznetsov, Semyon Malamud, and Teng Andrea Xu, 2024a, Large (and deep) factor models, *arXiv preprint arXiv:2402.06635* .

Kelly, Bryan, Semyon Malamud, and Kangying Zhou, 2024b, The virtue of complexity in return prediction, *The Journal of Finance* 79, 459–503.

Kelly, Bryan T, Semyon Malamud, and Kangying Zhou, 2022, The virtue of complexity everywhere, *Swiss Finance Institute Research Paper* .

Kyle, Albert S, 1985, Continuous auctions and insider trading, *Econometrica: Journal of the Econometric Society* 1315–1335.

Lo, Andrew W, and A Craig MacKinlay, 1990, Data-snooping biases in tests of financial asset pricing models, *The Review of Financial Studies* 3, 431–467.

Mondria, Jordi, 2010, Portfolio choice, attention allocation, and price comovement, *Journal of Economic Theory* 145, 1837–1864.

Nagel, Stefan, 2025, Seemingly virtuous complexity in return prediction, *Chicago Booth Research Paper* .

Nakkiran, Preetum, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever, 2021, Deep double descent: Where bigger models and more data hurt, *Journal of Statistical Mechanics: Theory and Experiment* 2021, 124003.

Potì, Valerio, 2018, A new tight and general bound on return predictability, *Economics Letters* 162, 140–145.

Sims, Christopher A, 2003, Implications of rational inattention, *Journal of monetary Economics* 50, 665–690.

Sirignano, Justin, and Rama Cont, 2021, Universal features of price formation in financial markets: perspectives from deep learning, in *Machine learning and AI in finance*, 5–15 (Routledge).

Sullivan, Ryan, Allan Timmermann, and Halbert White, 1999, Data-snooping, technical trading rule performance, and the bootstrap, *The journal of Finance* 54, 1647–1691.

Van Nieuwerburgh, Stijn, and Laura Veldkamp, 2009, Information immobility and the home bias puzzle, *The Journal of Finance* 64, 1187–1215.

Veldkamp, Laura L, 2006, Information markets and the comovement of asset prices, *The Review of Economic Studies* 73, 823–845.

Vives, Xavier, 2010, *Information and learning in markets: the impact of market microstructure* (Princeton University Press).

Vuong, Quang H, 1989, Likelihood ratio tests for model selection and non-nested hypotheses, *Econometrica: journal of the Econometric Society* 307–333.

Welch, Ivo, and Amit Goyal, 2008, A comprehensive look at the empirical performance of equity premium prediction, *The Review of Financial Studies* 21, 1455–1508.

White, Halbert, 2000, A reality check for data snooping, *Econometrica* 68, 1097–1126.

Zhou, Guofu, 2010, How much stock return predictability can we expect from an asset pricing model?, *Economics Letters* 108, 184–186.

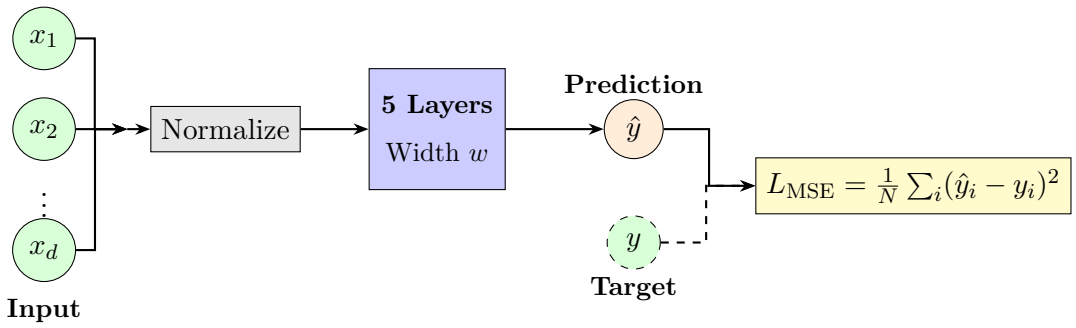
Internet Appendix

A. Neural Network Architecture & Training

In this section, we explain how we approximate the unknown functional mapping from the high-dimensional information set x_t to expected returns, $g(x_t)$.

Building on the universal approximators (Hornik et al. (1989)) theorem, we use neural nets as the class of models for this mapping. Specifically, we approximate the conditional mean function using feedforward neural networks of varying complexity. The architecture and training procedure are designed to isolate the effect of model capacity on forecasting performance while maintaining consistency across the wide range of model sizes required to estimate scaling laws. We provide details of the construction, training, and evaluation of these networks.

Figure A1: Mean Squared Error Neural Network Architecture



Note. Neural network architecture for return prediction. Input features $\mathbf{x}_t \in \mathbb{R}^d$ are first normalized to zero mean and unit variance, then processed through five hidden layers of equal width w. Each hidden layer applies a linear transformation (Dense), ReLU activation, and Layer Normalization. Dropout (rate 0.1) is applied to the middle three hidden layers only. The output layer produces a scalar prediction $\hat{y}_{t+h}$. The network is trained by minimizing mean squared error (MSE) between predictions and realized outcomes y_{t+h}. Model complexity is controlled by varying the width w, which determines the total parameter count.

A.1 Network architecture

Figure A1 provides a schematic overview of the network architecture for neural networks trained on mean squared error. Our neural network architecture follows a fixed-depth design with five hidden layers of equal width. This architectural choice serves two purposes. First,

holding depth constant while varying width isolates the effect of parameter count on model performance, avoiding confounding variation in network depth. Second, equal-width layers provide smoother transitions across parameter counts compared to tapered architectures which exhibit discrete jumps in effective capacity as layers are added or removed.

Given a target parameter count, we determine the layer width through binary search. For an input dimension of d features, five hidden layers of width w, and a scalar output, the total parameter count equals $dw + 4w^2 + w$ when counting only weight matrices. We search for the width w that matches the target parameter count within one percent, ensuring that models across the parameter range are comparable in their depth-to-width ratio.

Input features are first passed through a normalization layer that standardizes each feature to zero mean and unit variance using statistics computed from the training sample. This input normalization is critical for stable optimization across the wide range of model sizes we consider, as it ensures that gradient magnitudes remain comparable regardless of the original feature scales.

Each hidden layer consists of a linear transformation followed by a rectified linear unit activation function and layer normalization. We apply layer normalization rather than batch normalization because its statistics are computed per-sample rather than per-batch, making it more stable when batch sizes are large relative to the effective degrees of freedom in smaller models. The output layer applies a linear transformation without activation to produce a scalar prediction.

To reduce extreme double-descent fluctuations, we apply dropout at rates of 10% or 20%, depending on the application. For firm-level prediction tasks, we use a 10% dropout rate because the larger number of observations provides greater implicit regularization. In contrast, market risk premium prediction relies on a single time series, which substantially limits implicit regularization; accordingly, we use 20% dropout to stabilize training. Dropout is applied only to the middle three hidden layers, not the first or last, preserving the early-layer input representation and the late-layer output mapping while regularizing intermediate

representations where overfitting is most likely. We initialize weights with He normal initialization to account for the variance scaling induced by ReLU activations, and we initialize all biases to zero.

A.2 Training procedure

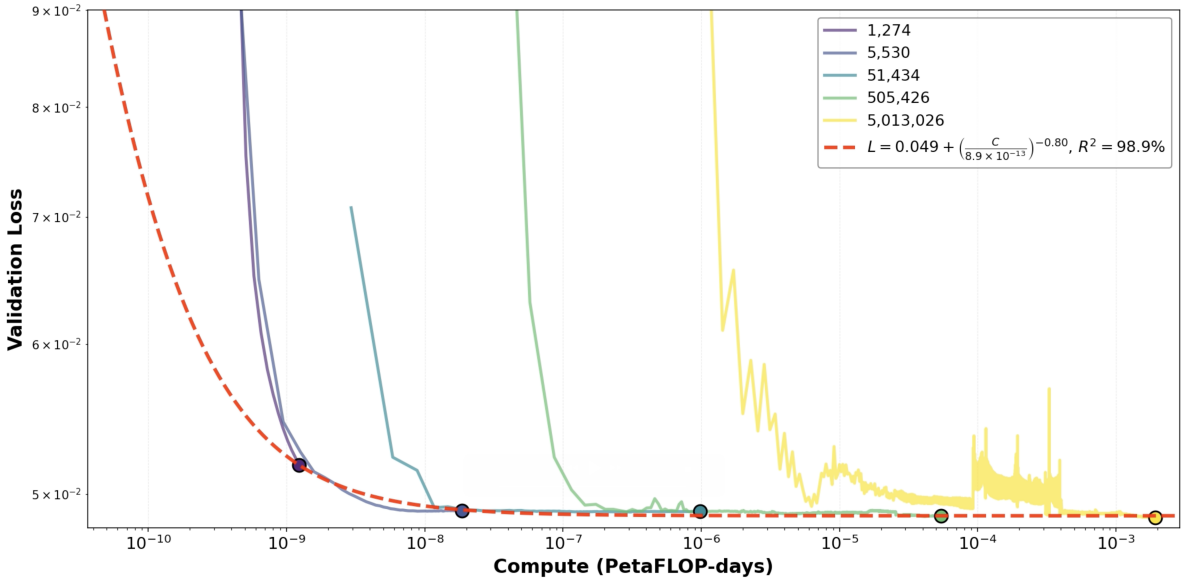
We train all networks by minimizing mean squared error using the Adam optimizer with an initial learning rate of 0.001. To prevent gradient explosion which can occur when training very large models on noisy financial data, we clip gradient norms at unity before each parameter update. This gradient clipping ensures stable training dynamics across the full range of model sizes without requiring size-specific hyperparameter tuning.

The learning rate is adjusted during training using a plateau-based scheduler. When validation loss fails to improve for a patience window equal to one-fifth of the total training epochs, the learning rate is reduced by half. The plateau scheduler allows the optimizer to escape shallow local minima early in training while enabling fine-grained convergence in later epochs.

Training duration scales with model size to ensure that larger models have sufficient time to traverse the interpolation threshold and enter the double-descent regime. Specifically, we train firm-level return models for a number of epochs proportional to the parameter count raised to the 0.75 power. This sublinear scaling reflects the empirical observation that larger models require more epochs to converge, with diminishing marginal returns to additional training. For the firm-level return prediction application, we use $\lfloor 0.1 \times p^{0.75} \rfloor$ epochs, whereas for the aggregate market prediction application, we use $\lfloor 2 \times p^{0.4} \rfloor$ epochs to account for the smaller sample size. Our scaling choices reflect a practical trade-off: provided models are trained sufficiently long, we can still recover scaling laws, but excessively long training is computationally costly for the full set of experiments in this paper. We therefore set training schedules at levels where performance begins to asymptote across model sizes, without incurring unnecessary additional computation.

We deliberately do not employ early stopping based on validation loss. While early stopping can prevent overfitting in the classical bias-variance regime, it would prevent our models from passing through the interpolation threshold into the double-descent regime where test loss decreases again. This can be empirically seen in Figure A2. If one were to implement early stopping, the 5M parameter model would not be able to achieve the asymptotic loss it could achieve. This is analogous to the stylized effect drawn in Figure 1. By training to completion, we allow sufficiently large models to benefit from the implicit regularization that emerges in the overparameterized regime.

Figure A2: Double Descent



Note. This figure plots validation MSE against cumulative compute (PetaFLOP-days) for a subset of FNW36 firm-level return prediction networks. Colored lines trace validation loss over training for five model sizes (parameter counts in the legend); circled markers indicate each run's best validation loss, and the dashed red curve is the fitted scaling law. The non-monotone trajectories illustrate double descent around the interpolation regimes. Over-parameterized models must be trained sufficiently long enough (no early stopping) to pass the interpolation region and exploit implicit regularization that yields lower and more stable validation loss.

The batch size is fixed at 65,536 observations for both applications. This large batch size serves multiple purposes. It reduces gradient variance, which is particularly important given the high noise-to-signal ratio in financial return prediction. It also improves GPU throughput by maximizing parallelism in matrix operations. Most importantly, it ensures that the scaling

laws we estimate reflect model capacity rather than optimization noise, as smaller batch sizes would introduce additional stochasticity that could obscure the relationship between compute and performance.

All models use 32-bit floating point precision with deterministic operations enabled. We fix random seeds for weight initialization, dropout masks, and data shuffling to ensure reproducibility. These choices prioritize consistency and reproducibility over computational efficiency, as our goal is to characterize the fundamental relationship between model capacity and forecasting performance rather than to minimize training time.

A.3 Data splitting

Data are split temporally to prevent look-ahead bias, which is particularly important in financial applications where future information could artificially inflate performance metrics. For the firm-level return prediction application using the characteristics of Freyberger et al. (2020), the final twenty percent of dates form the out-of-sample test set. From the remaining eighty percent, 10 percent is reserved for validation, leaving 70 percent for training. For the aggregate market prediction application using the predictors of Welch and Goyal (2008), we use a test size of 35 percent and validation size of 15 percent to account for the smaller total sample size.

The validation set serves to monitor training progress and trigger learning rate reductions but is not used for model selection via early stopping. This distinction is important: we use validation loss to adapt the optimization trajectory during training, but we do not use it to determine when to stop training. The test set is held out entirely during training and used only for final evaluation, ensuring that our reported scaling laws reflect true out-of-sample performance.

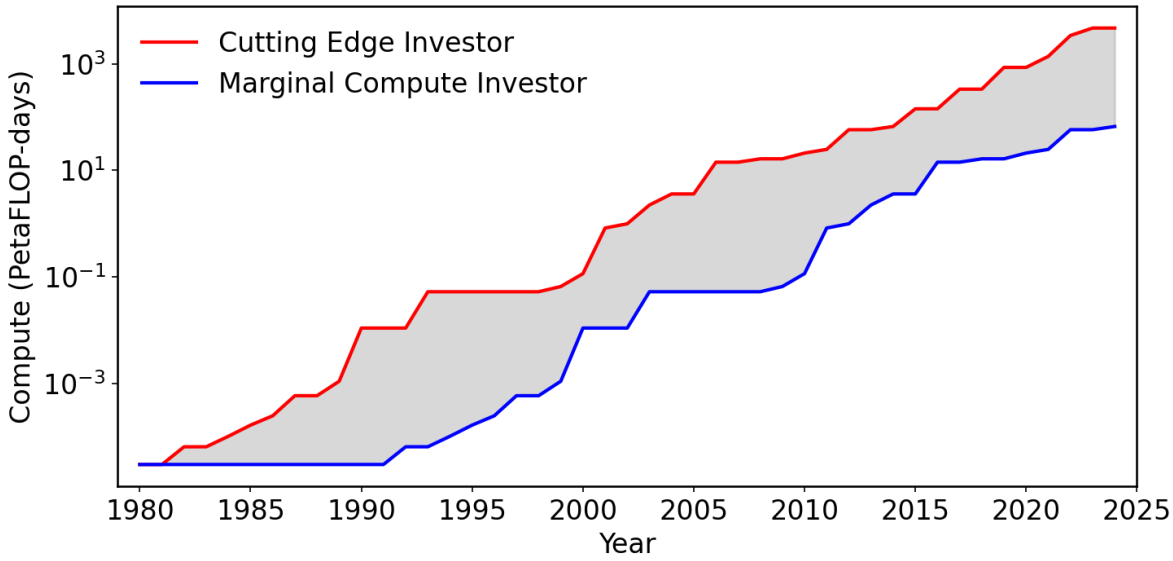
A.4 Compute measurement

To construct scaling laws, we measure computational cost in PetaFLOP-days, following the convention established in the machine learning literature. For each model, we estimate the floating point operations per training epoch as six times the parameter count times the number of training observations. This approximation accounts for the forward pass, backward pass, and parameter update, each of which requires approximately two floating point operations per parameter per observation. Cumulative compute is then computed as the sum of per-epoch compute across all training epochs, converted to PetaFLOP-days by dividing by 8.64×10^{19} operations per PetaFLOP-day.

This compute measure provides a hardware-agnostic way to compare models of different sizes and enables comparison with scaling laws estimated in other domains such as natural language processing and computer vision. By expressing performance as a function of compute rather than raw parameter count, we account for the fact that larger models require not only more parameters but also more training iterations to reach their asymptotic performance.

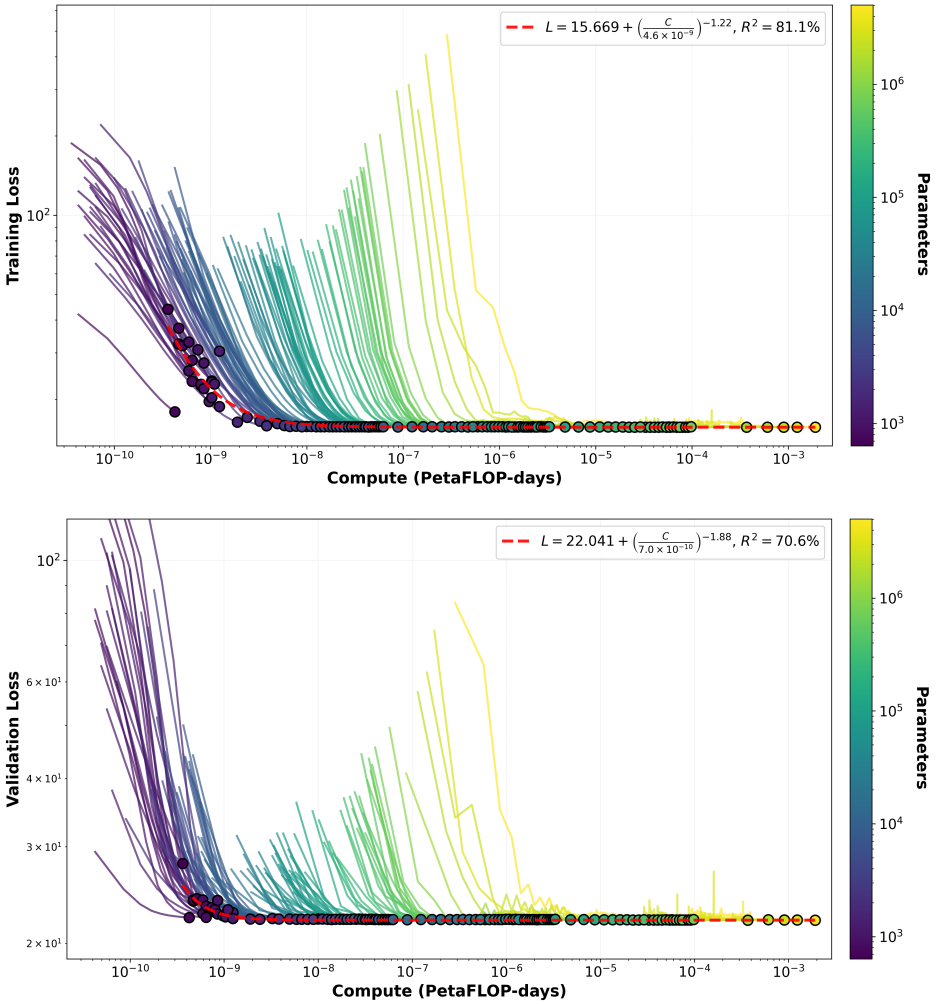
B. Additional Figures

Figure A3: Compute Over Time



Note. This figure plots estimated computational capacity (in PetaFLOP-days) for cutting-edge and marginal investors from 1980 to 2024. The "Cutting Edge Investor" series reports compute capacity derived from the maximum CPU transistor count available in each year. The "Marginal Compute Investor" series reports the cutting-edge technology from ten years prior, representing an investor who delays hardware upgrades. Transistor counts are obtained from *Wikipedia's Transistor Count* tables and converted to PetaFLOP-days using an empirically calibrated power-law model. The shaded region represents the computational gap between investor types.

Figure A4: Scaling Laws in Training and Validation



Note. This figure plots training and validation MSE against cumulative compute (PetaFLOP-days) for FNW-36 firm-level return prediction networks. The non-monotone trajectories illustrate double descent around the interpolation regimes. Over-parameterized models must be trained sufficiently long enough (no early stopping) to pass the interpolation region and exploit implicit regularization that yields lower and more stable validation loss.

Table A1: Public statements emphasizing compute and/or data scale.

Firm (linked)	**Public statement (verbatim excerpt)**
Renaissance Technologies	"A research database that grows by more than **40 terabytes a day**"; "**52,000 computer cores**"; "**Redundant computational facilities**."
Hudson River Trading	"Researchers have access to massive compute, custom tooling ..."; "**high compute-per-researcher ratio**"
Two Sigma	"Store **300+ petabytes** of data"; "Conduct more than **100,000** market data simulations daily"; "processing power measured in **zettaFLOPS**."
Jane Street	"We work with petabytes of data, a computing cluster with **hundreds of thousands of cores**, and a growing GPU cluster containing **thousands of high-end GPUs**."
Citadel Securities	"turn cutting-edge ideas and **petabyte-scale** data into ... models with direct trading impact."; "Leverage large scale compute and data **(petabytes; large budgets)** to run **ambitious experiments** and **push the frontier**"
Jump Trading	"Real-time inference on **petabytes** of market and alternative data"; "Massive HPC grid with **thousands of GPUs** ..."; "**5M+** Simulations Run Daily"
XTX Markets	"serve **hundreds of petabytes** and more than **100,000 compute nodes**"; "**650 Petabytes** of usable storage"; "**25,000 GPUs** in our research cluster"
D. E. Shaw	"... designs and maintains the firm's **largest compute infrastructure** ... **high-performance computing**, networking, and storage for research and trading."

Table A2: Estimated Scaling Laws by Firm-Level Forecast Decile

Decile	Equation	R^2 (%)
1	$-0.01 - \left(\frac{C}{4.3 \times 10^{-11}}\right)^{-0.23}$	79.2
2	$0.34 - \left(\frac{C}{3.4 \times 10^{-13}}\right)^{-0.21}$	60.5
3	$0.49 - \left(\frac{C}{1.7 \times 10^{-11}}\right)^{-0.89}$	7.0
4	$0.52 - \left(\frac{C}{7.3 \times 10^{-11}}\right)^{-1.97}$	1.8
5	$0.55 - \left(\frac{C}{5.8 \times 10^{-11}}\right)^{-3.13}$	-1.2
6	$0.80 - \left(\frac{C}{1.9 \times 10^{-66}}\right)^{-0.01}$	3.3
7	$1.10 - \left(\frac{C}{1.9 \times 10^{-34}}\right)^{-0.01}$	14.6
8	$1.41 - \left(\frac{C}{2.1 \times 10^{-16}}\right)^{-0.01}$	29.4
9	$0.74 - \left(\frac{C}{2.1 \times 10^{-13}}\right)^{-0.22}$	39.1
10	$1.20 - \left(\frac{C}{1.1 \times 10^{-11}}\right)^{-0.13}$	66.5
LS_{50}	$1.06 - \left(\frac{C}{7.5 \times 10^{-10}}\right)^{-0.58}$	65.9
LS_{30}	$1.24 - \left(\frac{C}{1.0 \times 10^{-9}}\right)^{-0.55}$	68.5
LS_{10}	$1.97 - \left(\frac{C}{3.6 \times 10^{-9}}\right)^{-0.43}$	74.2

Note. This table reports the estimated scaling-law fits $a-(C/b)^{\gamma}$ for each decile constructed on the firm-level forecasted return. The forecasts are generated with neural networks trained on the 36 Freyberger et al. (2020) characteristics. The last three rows represent long-short portfolios. They are constructed by going long and short on specific deciles, for instance, LS_{10} goes long on decile 10 and short on decile 1. The scaling laws' R^2 values are reported in percent.

C. Proofs

C.1 Proof of Proposition 1

Proof. The investor solves $\max_x E[-\exp(-A[W_0 + x(\tilde{\theta} - P)])|s, c]$. By CARA translation invariance, W_0 drops out. Given $\tilde{\theta}|s, c \sim \mathcal{N}(\mu(c, s), \tau(c)^{-1})$, we have $x(\tilde{\theta} - P)|s, c \sim \mathcal{N}(x[\mu(c, s) - P], x^2\tau(c)^{-1})$.

For $\tilde{Y} \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$, the CARA expectation is $E[-\exp(-A\tilde{Y})] = -\exp(-A\mu_Y + \frac{A^2\sigma_Y^2}{2})$. Therefore,

$$E[-\exp(-Ax(\tilde{\theta} - P))|s, c] = -\exp\left(-Ax[\mu(c, s) - P] + \frac{A^2x^2}{2\tau(c)}\right). \tag{31}$$

Maximizing is equivalent to maximizing the certainty equivalent $CE(x) = x[\mu(c, s) - P] - \frac{Ax^2}{2\tau(c)}$. The first-order condition yields

$$\mu(c, s) - P - \frac{Ax}{\tau(c)} = 0 \quad \Rightarrow \quad x(c) = \frac{\tau(c)}{A}[\mu(c, s) - P]. \tag{32}$$

Under the generalized compute transformation function, we have $\mu(c, s) = \frac{\tau_0\bar{\theta} + f(c)\tau_s s}{\tau_0 + f(c)\tau_s}$ and $\tau(c) = \tau_0 + f(c)\tau_s$. Substituting these expressions:

$$x(c) = \frac{\tau_0 + f(c)\tau_s}{A}\left[\frac{\tau_0\bar{\theta} + f(c)\tau_s s}{\tau_0 + f(c)\tau_s} - P\right] = \frac{1}{A}\left[\tau_0\bar{\theta} + f(c)\tau_s s - P(\tau_0 + f(c)\tau_s)\right]. \tag{33}$$

■

C.2 Proof of Proposition 2

Proof. Market clearing requires $\lambda x^* + (1 - \lambda)x' = \bar{z} + \tilde{u}$. Substituting optimal demands:

$$\begin{aligned}\lambda \cdot \frac{1}{A}[\tau_0\bar{\theta} + f(c^*)\tau_s s - P(\tau_0 + f(c^*)\tau_s)] \\ + (1 - \lambda) \cdot \frac{1}{A}[\tau_0\bar{\theta} + f(c')\tau_s s - P(\tau_0 + f(c')\tau_s)] = \bar{z} + \tilde{u}.\end{aligned} \tag{34}$$

Simplifying the left-hand side:

$$\tau_0\bar{\theta} + [\lambda f(c^*) + (1 - \lambda)f(c')]\tau_s s - P[\tau_0 + \tau_s(\lambda f(c^*) + (1 - \lambda)f(c'))] = A(\bar{z} + \tilde{u}). \tag{35}$$

Using $\bar{f} = \lambda f(c^*) + (1-\lambda) f(c')$ and $\bar{\tau} = \tau_0 + \tau_s \bar{f}$:

$$\tau_0 \bar{\theta} + \tau_s \bar{f} s - P\bar{\tau} = A(\bar{z} + \tilde{u}) \quad \Rightarrow \quad P = \frac{\tau_0 \bar{\theta} + \tau_s \bar{f} s}{\bar{\tau}} - \frac{A(\bar{z} + \tilde{u})}{\bar{\tau}}. \tag{36}$$

■

C.3 Proof of Proposition 3

Proof. Substitute the equilibrium price into the demand function:

$$x^* = \frac{1}{A}\left[\tau_0 \bar{\theta} + f(c^*)\tau_s s - \left(\frac{\tau_0 \bar{\theta} + \tau_s \bar{f} s}{\bar{\tau}} - \frac{A(\bar{z} + \tilde{u})}{\bar{\tau}}\right)(\tau_0 + f(c^*)\tau_s)\right]. \tag{37}$$

Using common denominator $A\bar{\tau}$:

$$x^* = \frac{1}{A\bar{\tau}}\left[\bar{\tau}(\tau_0 \bar{\theta} + f(c^*)\tau_s s) - (\tau_0 \bar{\theta} + \tau_s \bar{f} s)(\tau_0 + f(c^*)\tau_s) + A(\bar{z} + \tilde{u})(\tau_0 + f(c^*)\tau_s)\right]. \tag{38}$$

Expanding $\bar{\tau} = \tau_0 + \tau_s \bar{f}$:

$$\bar{\tau}(\tau_0 \bar{\theta} + f(c^*)\tau_s s) = \tau_0^2 \bar{\theta} + \tau_0 f(c^*)\tau_s s + \tau_0 \tau_s \bar{f}\bar{\theta} + \tau_s^2 \bar{f} f(c^*) s, \tag{39}$$

$$(\tau_0 \bar{\theta} + \tau_s \bar{f} s)(\tau_0 + f(c^*)\tau_s) = \tau_0^2 \bar{\theta} + \tau_0 f(c^*)\tau_s \bar{\theta} + \tau_0 \tau_s \bar{f} s + \tau_s^2 \bar{f} f(c^*) s. \tag{40}$$

The difference is:

$$\tau_0 f(c^*)\tau_s s + \tau_0 \tau_s \bar{f}\bar{\theta} - \tau_0 f(c^*)\tau_s \bar{\theta} - \tau_0 \tau_s \bar{f} s = \tau_0 \tau_s (f(c^*) - \bar{f})(s - \bar{\theta}). \tag{41}$$

Therefore:

$$x^* = \frac{1}{A\bar{\tau}}\left[\tau_0 \tau_s (f(c^*) - \bar{f})(s - \bar{\theta}) + A(\bar{z} + \tilde{u})(\tau_0 + f(c^*)\tau_s)\right]. \tag{42}$$

The proof for x' is identical with c' and $f(c')$ replacing c^* and $f(c^*)$. ■

C.4 Proof of Lemma 3.1

Proof. Consider a computational capacity level $\tilde{c}$ such that $f(\tilde{c}) = \bar{f}$. By the assumptions on f (continuous, strictly increasing, mapping $[0, \infty)$ onto $[0, \infty)$), such a $\tilde{c}$ exists and is unique.

For an investor operating at capacity $\tilde{c}$, the posterior mean is

$$\mu(\tilde{c}, s) = \frac{\tau_0\bar{\theta} + f(\tilde{c})\tau_s s}{\tau_0 + f(\tilde{c})\tau_s} = \frac{\tau_0\bar{\theta} + \bar{f}\tau_s s}{\tau_0 + \bar{f}\tau_s} = \frac{\tau_0\bar{\theta} + \bar{f}\tau_s s}{\bar{\tau}}. \tag{43}$$

From the equilibrium price:

$$P = \frac{\tau_0\bar{\theta} + \tau_s\bar{f}s}{\bar{\tau}} - \frac{A(\bar{z} + \tilde{u})}{\bar{\tau}}. \tag{44}$$

Therefore:

$$\mu(\tilde{c}, s) = P + \frac{A(\bar{z} + \tilde{u})}{\bar{\tau}}. \tag{45}$$

The investor at the representative capacity level $\tilde{c}$ has posterior beliefs that differ from the price only by the risk premium term $A(\bar{z} + \tilde{u})/\bar{\tau}$. ■

C.5 Proof of Proposition 4

Proof. The certainty equivalent for an investor with computational capacity c is

$$CE(c) = W_0 + \frac{1}{2A}[\mu(c, s) - P]^2\tau(c). \tag{46}$$

The marginal benefit of compute is

$$MB(c) = \frac{\partial CE(c)}{\partial c} = \frac{1}{2A}\frac{\partial}{\partial c}\left[[\mu(c, s) - P]^2\tau(c)\right]. \tag{47}$$

We compute the relevant derivatives. First, for the posterior precision:

$$\frac{\partial \tau(c)}{\partial c} = \frac{\partial}{\partial c}[\tau_0 + f(c)\tau_s] = f'(c)\tau_s. \tag{48}$$

Second, for the posterior mean $\mu(c,s) = \frac{\tau_0\bar{\theta} + f(c)\tau_s s}{\tau_0 + f(c)\tau_s}$, we apply the quotient rule:

$$\frac{\partial \mu(c,s)}{\partial c} = \frac{f'(c)\tau_s s(\tau_0 + f(c)\tau_s) - (\tau_0\bar{\theta} + f(c)\tau_s s)f'(c)\tau_s}{(\tau_0 + f(c)\tau_s)^2} \tag{49}$$

$$= \frac{f'(c)\tau_s[\tau_0 s + f(c)\tau_s s - \tau_0\bar{\theta} - f(c)\tau_s s]}{(\tau_0 + f(c)\tau_s)^2} \tag{50}$$

$$= \frac{f'(c)\tau_s\tau_0(s - \bar{\theta})}{(\tau_0 + f(c)\tau_s)^2}. \tag{51}$$

Applying the product rule to $[\mu(c,s) - P]^2\tau(c)$:

$$\frac{\partial}{\partial c}\left[[\mu(c,s) - P]^2\tau(c)\right] = 2[\mu(c,s) - P]\frac{\partial \mu(c,s)}{\partial c}\tau(c) + [\mu(c,s) - P]^2\frac{\partial \tau(c)}{\partial c} \tag{52}$$

$$= 2[\mu(c,s) - P]\frac{f'(c)\tau_s\tau_0(s - \bar{\theta})\tau(c)}{(\tau_0 + f(c)\tau_s)^2} + [\mu(c,s) - P]^2 f'(c)\tau_s. \tag{53}$$

Therefore:

$$MB(c) = \frac{f'(c)\tau_s}{2A}\left[\frac{2[\mu(c,s) - P]\tau_0(s - \bar{\theta})\tau(c)}{(\tau_0 + f(c)\tau_s)^2} + [\mu(c,s) - P]^2\right]. \tag{54}$$

Now evaluate at $c = \tilde{c}$ where $f(\tilde{c}) = \bar{f}$. Using Lemma 3.1, $\mu(\tilde{c},s) - P = \frac{A(\bar{z} + \tilde{u})}{\bar{\tau}}$ and $\tau(\tilde{c}) = \tau_0 + f(\tilde{c})\tau_s = \tau_0 + \bar{f}\tau_s = \bar{\tau}$:

$$MB(\tilde{c}) = \frac{f'(\tilde{c})\tau_s}{2A}\left[\frac{2 \cdot \frac{A(\bar{z} + \tilde{u})}{\bar{\tau}} \cdot \tau_0(s - \bar{\theta})\bar{\tau}}{\bar{\tau}^2} + \left(\frac{A(\bar{z} + \tilde{u})}{\bar{\tau}}\right)^2\right] \tag{55}$$

$$= \frac{f'(\tilde{c})\tau_s}{2A}\left[\frac{2A(\bar{z} + \tilde{u})\tau_0(s - \bar{\theta})}{\bar{\tau}^2} + \frac{A^2(\bar{z} + \tilde{u})^2}{\bar{\tau}^2}\right] \tag{56}$$

$$= \frac{f'(\tilde{c})\tau_s}{2A\bar{\tau}^2}\left[2A(\bar{z} + \tilde{u})\tau_0(s - \bar{\theta}) + A^2(\bar{z} + \tilde{u})^2\right]. \tag{57}$$

Taking expectations over the signal s conditional on $\tilde{u}$, we use $E[s - \bar{\theta}] = 0$ since $s = \theta + \eta$

with $E[\theta] = \bar{\theta}$ and $E[\eta] = 0$:

$$E[MB(\tilde{c})] = E[E[MB(\tilde{c}) \mid \tilde{u}]] = \frac{f'(\tilde{c})\tau_s A\, E[(\bar{z} + \tilde{u})^2]}{2\bar{\tau}^2} = \frac{f'(\tilde{c})\tau_s A(\bar{z}^2 + \tau_u^{-1})}{2(\tau_0 + \tau_s \bar{f})^2}. \tag{58}$$

■

C.6 Proof of Proposition 5

Proof. We evaluate the expected marginal benefit along the compute efficient frontier at the representative computational capacity in each market. Writing this representative capacity simply as c (so that $f(c) = \bar{f}$ and $\bar{\tau} = \tau_0 + \tau_s f(c)$), Proposition 4 implies that, conditional on the supply shock,

$$E[MB(c)] = \frac{f'(c)\tau_s A\, E[(\bar{z} + \tilde{u})^2]}{2(\tau_0 + \tau_s f(c))^2} = \frac{f'(c)\tau_s A(\bar{z}^2 + \tau_u^{-1})}{2(\tau_0 + \tau_s f(c))^2}. \tag{59}$$

Define the constant $K \equiv \frac{\tau_s A(\bar{z}^2 + \tau_u^{-1})}{2} > 0$. For the marginal benefit to satisfy $E[MB(c)] = Bc^{-\gamma}$ for some $B > 0$, we require

$$\frac{Kf'(c)}{(\tau_0 + \tau_s f(c))^2} = Bc^{-\gamma}. \tag{60}$$

Consider the candidate $f(c) = \kappa c^\alpha$ for constants $\kappa > 0$ and $\alpha \in (0, 1)$. This specification satisfies our maintained assumptions: $f(0) = 0$, $f'(c) = \alpha\kappa c^{\alpha-1} > 0$, and $f''(c) = \alpha(\alpha - 1)\kappa c^{\alpha-2} < 0$, so the transformation exhibits diminishing returns. Substituting yields

$$E[MB(c)] = \frac{K\alpha\kappa c^{\alpha-1}}{(\tau_0 + \tau_s \kappa c^\alpha)^2}. \tag{61}$$

For large c, the term $\tau_s \kappa c^\alpha$ dominates τ_0 in the denominator, giving the asymptotic form

$$E[MB(c)] \sim \frac{K\alpha\kappa c^{\alpha-1}}{\tau_s^2 \kappa^2 c^{2\alpha}} = \frac{K\alpha}{\tau_s^2 \kappa} c^{-(1+\alpha)}. \tag{62}$$

This establishes sufficiency: $f(c) = \kappa c^{\alpha}$ generates a power law with exponent $\gamma = 1 + \alpha$. The power law holds exactly when $\tau_0 = 0$ and provides an excellent approximation whenever $\tau_s \kappa c^{\alpha} \gg \tau_0$. Since $\alpha \in (0, 1)$ ensures diminishing returns in the transformation function, the power law exponent satisfies $\gamma \in (1, 2)$. ■

C.7 Derivative of Expected Marginal Benefit

Under the power law specification $f(c) = \kappa c^{\alpha}$ with $\alpha \in (0, 1)$, we derive the derivative of the expected marginal benefit with respect to computational capacity.

Proof. From Proposition 4, define $K \equiv \frac{\tau_s A(\bar{z}^2 + \tau_u^{-1})}{2}$ so that

$$E[MB(c)] = \frac{\alpha \kappa K c^{\alpha - 1}}{(\tau_0 + \kappa \tau_s c^{\alpha})^2}. \tag{63}$$

Applying the quotient rule and simplifying:

$$\frac{\partial E[MB(c)]}{\partial c} = \frac{\alpha \kappa K c^{\alpha - 2} \left[(\alpha - 1)\tau_0 - (1 + \alpha)\kappa \tau_s c^{\alpha}\right]}{(\tau_0 + \kappa \tau_s c^{\alpha})^3}. \tag{64}$$

This derivative is negative for all $c > 0$. The denominator and the factor $\alpha \kappa K c^{\alpha - 2}$ are both positive. The bracketed term is the sum of $(\alpha - 1)\tau_0 < 0$ and $-(1 + \alpha)\kappa \tau_s c^{\alpha} < 0$, hence negative. The compute efficient frontier is therefore strictly downward sloping.

When $\tau_0 = 0$, the expression simplifies to

$$\frac{\partial E[MB(c)]}{\partial c} = -\frac{(1 + \alpha)\alpha A(\bar{z}^2 + \tau_u^{-1})}{2\kappa \tau_s} c^{-(2 + \alpha)}, \tag{65}$$

confirming that the rate of decline of marginal benefits itself follows a power law with exponent $2 + \alpha$. ■

C.8 Proof of Proposition 6

Proof. The computational efficiency measure is

$$\mathcal{E} \equiv \frac{(\tau_0 + \tau_s \bar{f})^2}{f'(\tilde{c})\tau_s} \tag{66}$$

where $\tilde{c}$ satisfies $f(\tilde{c}) = \bar{f}$.

To show $\mathcal{E}$ is strictly increasing in $\bar{f}$, note that implicit differentiation of $f(\tilde{c}) = \bar{f}$ yields $\partial\tilde{c}/\partial\bar{f} = 1/f'(\tilde{c})$. Differentiating $\mathcal{E}$ and simplifying:

$$\frac{\partial \mathcal{E}}{\partial \bar{f}} = \frac{(\tau_0 + \tau_s \bar{f})}{f'(\tilde{c})^2 \tau_s} \left[2\tau_s f'(\tilde{c}) - (\tau_0 + \tau_s \bar{f}) \frac{f''(\tilde{c})}{f'(\tilde{c})} \right]. \tag{67}$$

Since $f'(\tilde{c}) > 0$ and $f''(\tilde{c}) \leq 0$, we have $-(\tau_0 + \tau_s \bar{f}) \frac{f''(\tilde{c})}{f'(\tilde{c})} \geq 0$, so the bracketed term is strictly positive and hence $\partial\mathcal{E}/\partial\bar{f} > 0$.

For the limiting behavior, note that $f(\tilde{c}) = \bar{f}$ and $f(0) = 0$ with f strictly increasing implies $\tilde{c} \to 0$ as $\bar{f} \to 0$ and $\tilde{c} \to \infty$ as $\bar{f} \to \infty$. If $f'(0^+) = \lim_{c \to 0^+} f'(c)$ exists (possibly $+\infty$), then

$$\lim_{\bar{f} \to 0} \mathcal{E} = \lim_{\bar{f} \to 0} \frac{(\tau_0 + \tau_s \bar{f})^2}{\tau_s f'(\tilde{c})} = \frac{\tau_0^2}{\tau_s f'(0^+)}. \tag{68}$$

Moreover, concavity implies that f' is nonincreasing, so for any fixed $c_0 > 0$ and all sufficiently large $\bar{f}$ (equivalently, large $\tilde{c}$), we have $f'(\tilde{c}) \leq f'(c_0) < \infty$. Since $(\tau_0 + \tau_s \bar{f})^2 \to \infty$, it follows that $\mathcal{E} \to \infty$ as $\bar{f} \to \infty$.

Finally, the inverse relationship with marginal benefit follows directly from the definitions. Since $\mathcal{E} = (\tau_0 + \tau_s \bar{f})^2 / (f'(\tilde{c})\tau_s)$, substitution into the expression from Proposition 4 gives

$$E[MB(\tilde{c})] = \frac{f'(\tilde{c})\tau_s A(\bar{z}^2 + \tau_u^{-1})}{2(\tau_0 + \tau_s \bar{f})^2} = \frac{A(\bar{z}^2 + \tau_u^{-1})}{2\mathcal{E}}. \tag{69}$$

■