

Machine Learning Meets Markowitz

Campbell R. Harvey
Duke University and NBER

Joint Work With



Machine Learning Meets Markowitz

PBCSF-NIFR Research Paper

85 Pages • Posted: 22 Dec 2025 • Last revised: 18 Apr 2026

[Yijie Wang](#)

Tongji University - School of Economics and Management

[Hao Gao](#)

Tsinghua University - PBC School of Finance

[Campbell R. Harvey](#)

Duke University - Fuqua School of Business; National Bureau of Economic Research (NBER)

[Yan Liu](#)

Tsinghua University - Tsinghua University School of Economics and Management

[Xinyuan Tao](#)

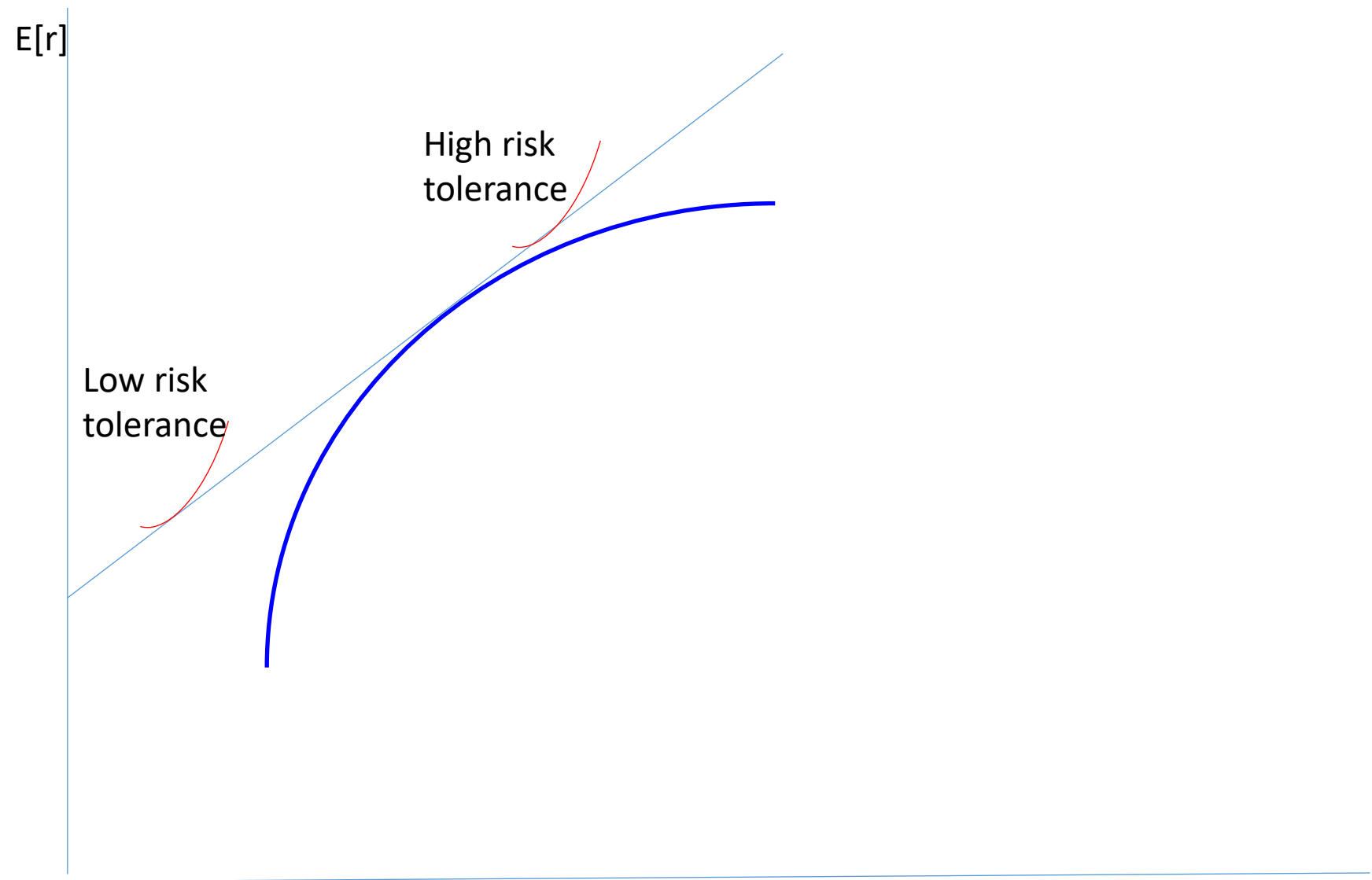
New Jersey Institute of Technology

Date Written: April 17, 2026

Abstract

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

Finance 101



Finance 101?



Defying Convention

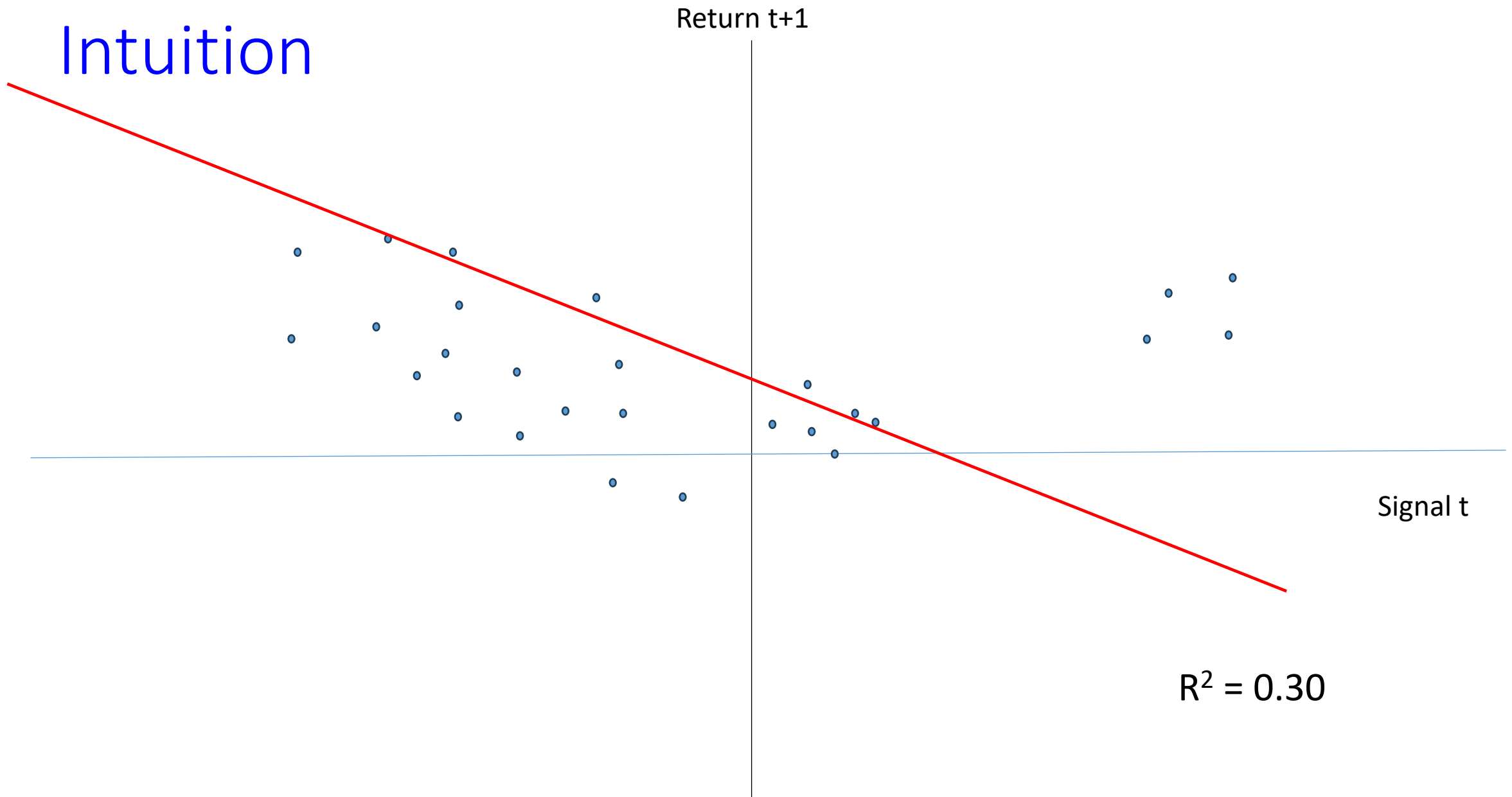
The usual approach

- Generate (forecast) expected returns, covariances
- Plug into optimizer and select portfolio based on risk aversion

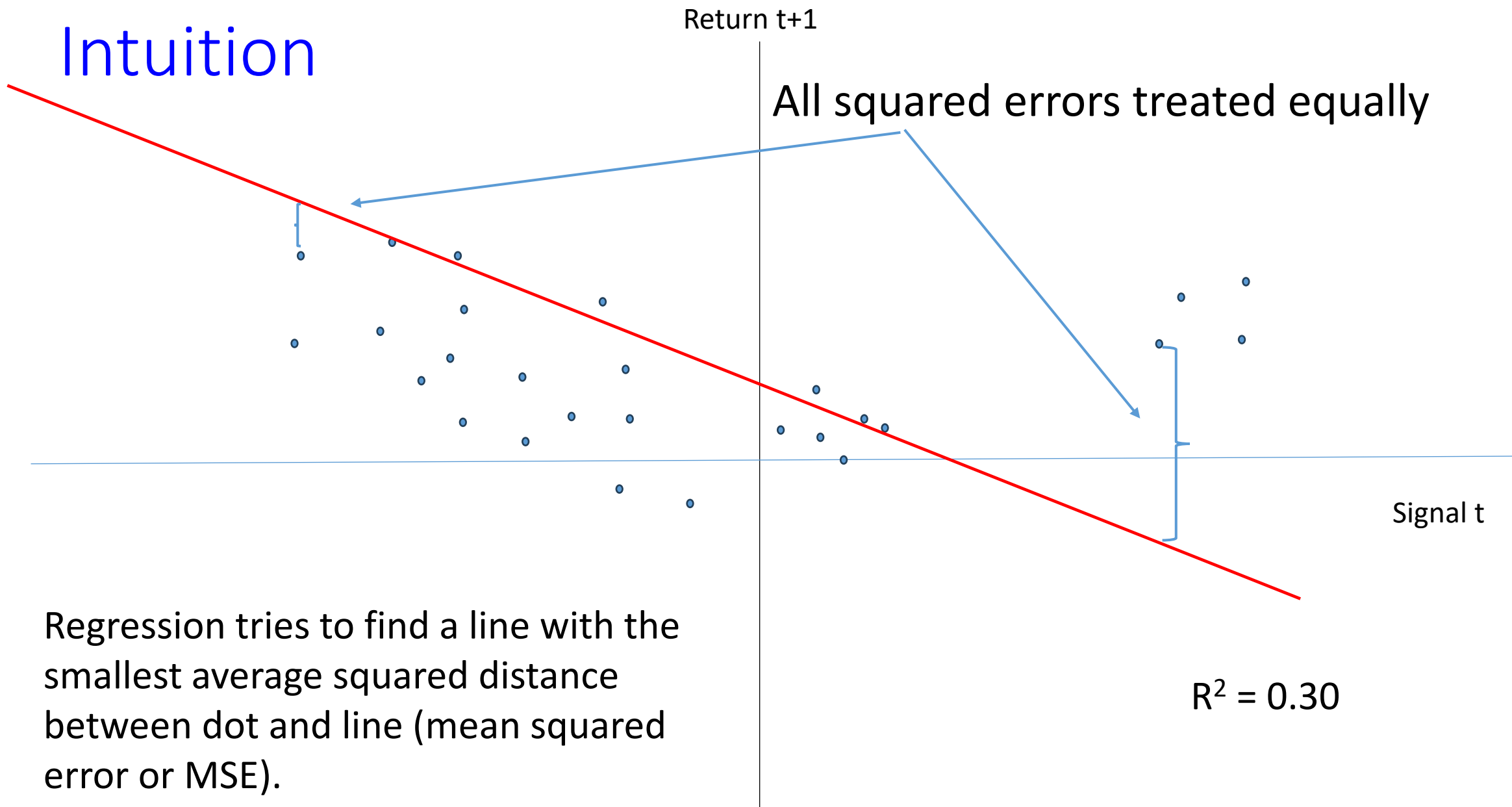
The two-step ‘predict then optimize’ is sub-optimal because the first stage treats all asset prediction errors as equally important.

Investors prefer more prediction precision for assets more likely to end up in their portfolio.

Intuition



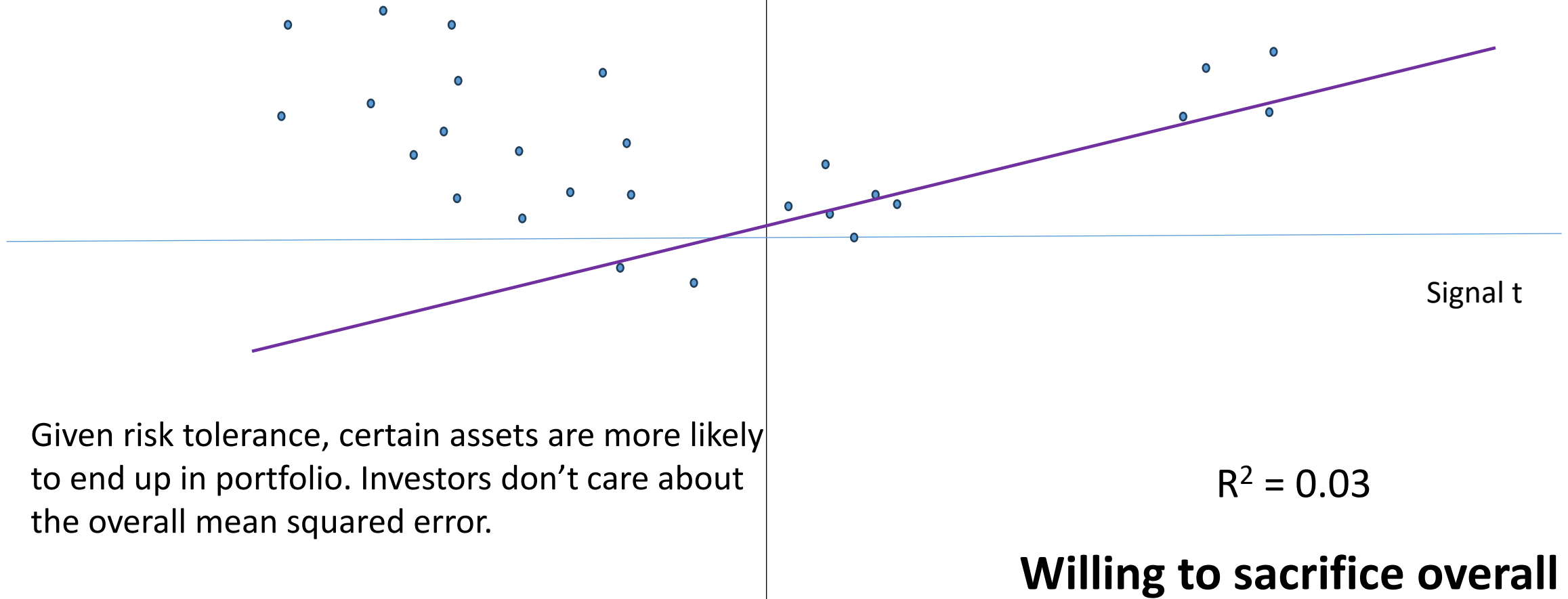
Intuition



Intuition

Return t+1

Investors want maximum precision for the assets most likely to be in their portfolio.



The Standard Two-Stage Paradigm

The standard approach in both academia and industry is a **two-stage “predict, then optimize”** process.

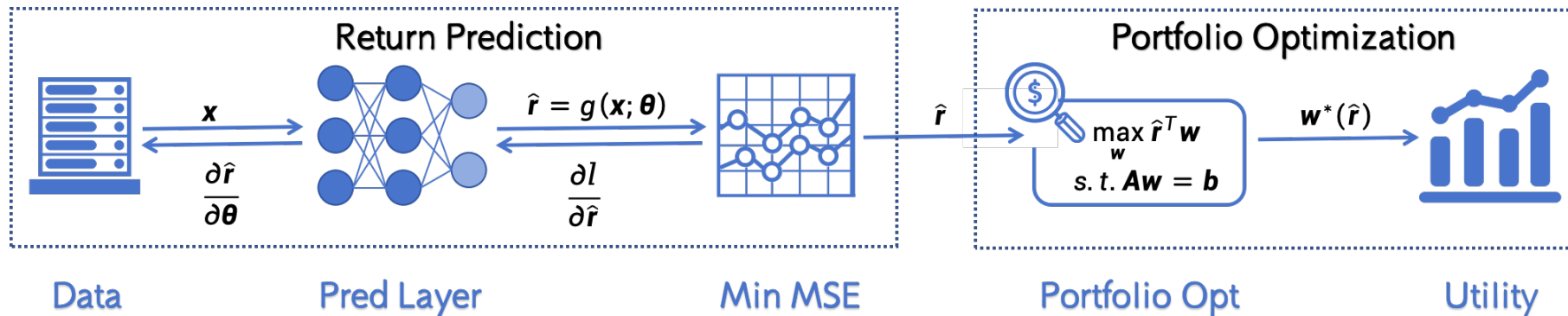
Stage 1: Predict

- Given firm characteristics x , train a prediction model $\hat{r} = g(x, \theta)$ parameterized by θ .
- Objective: Minimize prediction error (e.g., MSE).



Stage 2: Optimize

- Use the predictions $\hat{r}$ as input.
- Objective: Find portfolio weights w^* that maximize utility $u(w^*, \hat{r})$.



The Standard Two-Stage Paradigm

The standard approach in both academia and industry is a **two-stage “predict, then optimize”** process.

Stage 1: Predict

- Given firm characteristics x , train a prediction model $\hat{r} = g(x, \theta)$ parameterized by θ .
- Objective: Minimize prediction error (e.g., MSE).



Stage 2: Optimize

- Use the predictions $\hat{r}$ as input.
- Objective: Find portfolio weights w^* that maximize utility $u(w^*, r)$.

The Consequence: A Critical Disconnect

This separation creates a fundamental gap between asset pricing (prediction) and portfolio construction (decision-making), leading to several problems:

- Misaligned objective
- High transaction costs
- Ignoring investment constraints

Example: A Four Asset Investment Problem (1/3)

- Consider a **Mean-Variance** investor solving:

$$\max_w \quad w^\top \hat{r} - \eta w^\top \Sigma w \quad \text{s.t.} \quad w \geq 0, \quad w^\top e = 1$$

- Setup:** Four assets driven by two characteristics: *Yield* and *Growth*.
 - **Large-cap (i=1,2):** Expected returns driven more by *Yield*.
 - **Small-cap (i=3,4):** Expected returns driven more by *Growth*.

$$r_{i,t+1} = \begin{cases} \text{Yield}_{i,t} + 0.2 \text{Growth}_{i,t} & \text{Large-cap} \quad (i = 1, 2) \\ 0.2 \text{Yield}_{i,t} + \text{Growth}_{i,t} & \text{Small-cap} \quad (i = 3, 4) \end{cases}$$

- The characteristics matrix (x) and covariance matrix (Σ):

$$x = \begin{bmatrix} 0.3 & 0.1 \\ 0.35 & 0.2 \\ 0.1 & 0.9 \\ 0.2 & 1.0 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.0064 & 0.00288 & 0 & 0 \\ 0.00288 & 0.0144 & 0 & 0.0084 \\ 0 & 0 & 0.36 & 0.294 \\ 0 & 0.0084 & 0.294 & 0.49 \end{bmatrix}$$

Yield *Growth*

Example: Evaluating Predictive Models (2/3)

We compare three distinct linear prediction models, $g(x, \theta)$:

- **Overall (O):** The standard approach. It strictly minimizes the global Mean Squared Error across all assets.

$$g_o(x, \theta) = 0.652Yield_{i,t} + 0.927Growth_{i,t} \quad (1)$$

- **Large (L):** Adjusted to emphasize Earnings Yield. It has a **worse** overall MSE but predicts large-caps more accurately.

$$g_L(x, \theta) = 1.05 \times 0.652Yield_{i,t} + 0.95 \times 0.927Growth_{i,t} \quad (2)$$

- **Small (S):** Adjusted to emphasize Growth. It has a **worse** overall MSE but predicts small-caps more accurately.

$$g_S(x, \theta) = 0.95 \times 0.652Yield_{i,t} + 1.05 \times 0.927Growth_{i,t} \quad (3)$$

Which model generates the best portfolio performance (realized utility)?

Example: Results & Realized Utility (3/3)

We evaluate the realized utility of the resulting portfolios for two different investors.

Investor Type	O	L	S
Risk Averse ($\eta = 10$)	31.97%	32.01%	31.90%
Risk Tolerant ($\eta = 1$)	62.84%	62.53%	62.90%

Conclusion

- ▶ The MSE model **fails** to maximize utility for either investor, despite having the best statistical fit.
- ▶ Risk-averse investors achieve higher utility when the model is tuned for lower-risk assets (Pro-Large).
- ▶ Risk-tolerant investors achieve higher utility when the model is tuned for higher-risk assets (Pro-Small).

Ignoring Economic Restrictions

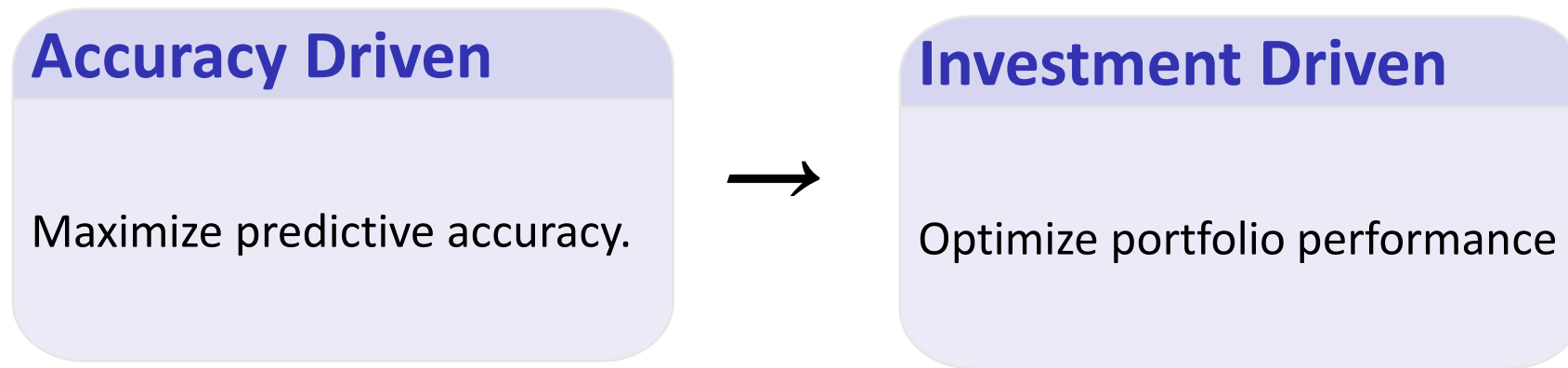
The prediction model generates “raw” return forecasts without considering if they are **feasible or practical** to implement [Avramov et al., 2023, Wang, 2025] .

- This leads to implied portfolios that are often unrealistic:
 - **High Turnover:** Frequent trading leads to substantial **transaction costs** that erode gross returns.
 - **Extreme Positions:** The model might suggest allocating **50% of capital to a single stock**, violating risk mandates.
 - **Difficult-to-Trade Stocks:** Profitability is often concentrated in **small-cap or illiquid stocks** where it is hard to deploy significant capital.
 - **Constraint Violations:** The predictions don't respect market rules like **short-selling restrictions**.
- **The Bottom Line:** High statistical performance on paper may fail to translate into implementable, net-positive returns.

Our Contribution: A Decision-Aware Framework

We propose a systematic and practical solution to bridge this gap.

- We shift the focus



This allows the model to learn a prediction function that is explicitly **aware** of investor objectives and constraints.

First, let's consider existing approaches.

Integrated Learning via SDF and Tangency Portfolio (I)

- Estimating the stochastic discount factor (SDF) that minimizes pricing errors is equivalent to finding the conditional mean–variance efficient (tangency) portfolio.
- Recent ML work therefore estimates the SDF directly, aiming for a one-step portfolio construction rather than predict-then-optimize:
 - Kozak et al. (2020): Shrinkage on SDF weights in the factor zoo.
 - Gu et al. (2021): Autoencoder latent factors spanning the frontier.
 - Didisheim et al. (2024): Virtue of complexity for recovering the tangency portfolio
 - Chen et al. (2024): Deep nonlinear SDF with no-arbitrage in the loss.
 - Feng et al. (2026): Deep long–short factor + market → deep tangency portfolio.

Integrated Learning via SDF and Tangency Portfolio (II)

- A one-step path to the maximum Sharpe / tangency portfolio—economically elegant and empirically strong in frictionless settings.
- Implementation gaps:
 - **Frictionless assumption:** The target is the unconstrained tangency portfolio.
 - **Extreme positions:** Implied portfolios often require high leverage or aggressive shorting—difficult in practice [Avramov et al., 2023].
 - **Missing mandates:** Generally, do not embed long-only, position caps, or transaction costs.

Parametric Portfolio Policy (PPP)

PPP learns a linear policy $w = w_0 + \alpha x$ from characteristics to portfolio weights by maximizing the utility function, where w_0 is a benchmark portfolio (e.g., equal weighted) [Brandt et al. , 2009]:

$$\max_w u(w, r) = \max_{\alpha} u(w_0 + \alpha x, r)$$

- Due to the lack of a constrained optimization step, the learned parametric policy could be infeasible for some characteristic realizations.

Example

Suppose the learned policy function is $w = 0.5 + 0.5x$, and the portfolio manager has constraints $\{0 \leq w_1 \leq 0.6, 0 \leq w_2 \leq 0.6, w_1 + w_2 \leq 1\}$, then consider two characteristic realizations:

Sample 1: $x = [0.2, -0.2] \rightarrow w = [0.6, 0.4]$ (feasible)

Sample 2: $x = [0.4, -0.4] \rightarrow w = [0.7, 0.3]$ (Infeasible!)

Deep Reinforcement Learning: AlphaPortfolio

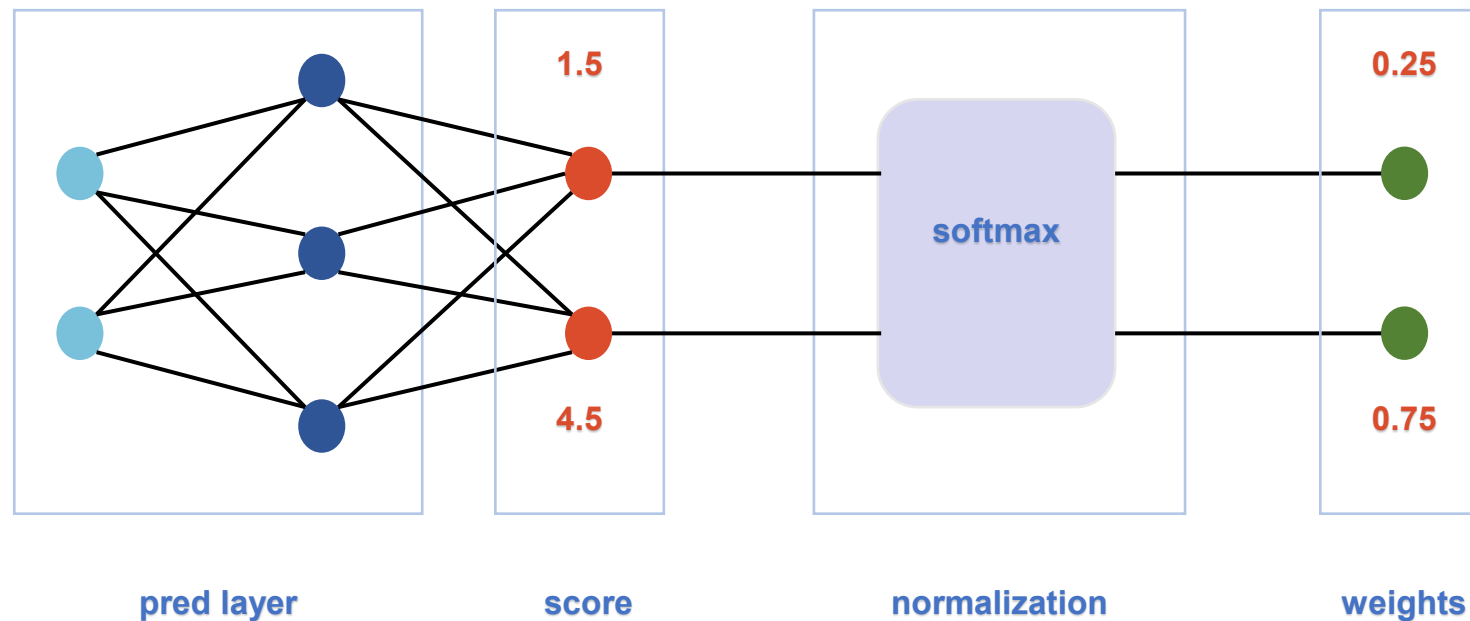
AlphaPortfolio shares the same philosophy as PPP, training portfolio policies directly on investment performance (utility/rewards) rather than on statistical prediction error. [Cong et al. , 2021].

- It Use a deep neural network to capture nonlinear mappings from characteristics to asset scores.
- Convert scores into portfolio weights with a normalization layer (e.g., softmax).
- Optimize the policy end-to-end for realized investment objectives.

Deep Reinforcement Learning: AlphaPortfolio

Limitation:

- Final weights are fully determined by score outputs. There is no constrained optimization step, so institutional rules cannot be enforced exactly.
- For example, under constraints such as $w \leq 0.6$, the following softmax-implied weights (0.75 and 0.25) can be infeasible.



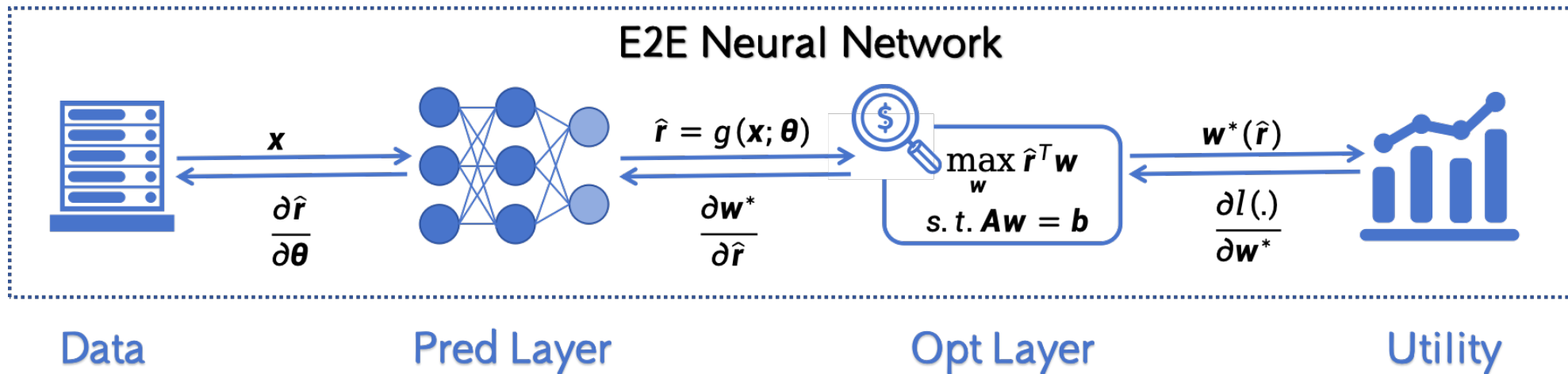
Interpretable Decision-Aware Learning: AlphaGlass

AlphaGlass [Bell et al. , 2026] also trains end-to-end on investment objectives (e.g., Sharpe ratio). It maps firm characteristics into additive signals, then convert scores into portfolios via a differentiable rank-and-mask layer that approximates conventional quantile sorting.

- However, the portfolio map is built for long–short (high-minus-low) quantile portfolios, not a general constrained optimizer.
- The model cannot precisely enforce institutional mandates such as position caps, exposure restrictions, or other hard feasibility constraints.

Our Method: The E2E Neural Network

Unlike PPP and AlphaPortfolio, our E2E framework integrates a formal **portfolio optimization layer** — making feasibility part of the learning architecture, not an afterthought.



Forward Pass

- Input factors x
- Predict returns $\hat{r}$
- Solve for optimal portfolio w^*
- Evaluate realized utility

Backward Pass

Gradients from the final loss flow back to update the prediction layer's parameters

The E2E Framework: A Bilevel Perspective

Bilevel Optimization Formulation

$$\begin{aligned} \max_{\theta} \quad & u(\{w_t^*, r_{t+1}\}_{t=1}^T) \\ \text{s.t.} \quad & w_t^* = \operatorname{argmax}_{w_t \in \mathcal{W}} u_t(w_t, g(x_t; \theta)) \quad \forall t \in [T] \end{aligned}$$



prediction function

Upper-level (Leader)

Return prediction

- Train the prediction function parameters θ .
- Goal: Maximize the overall utility function.

Lower-level (Follower)

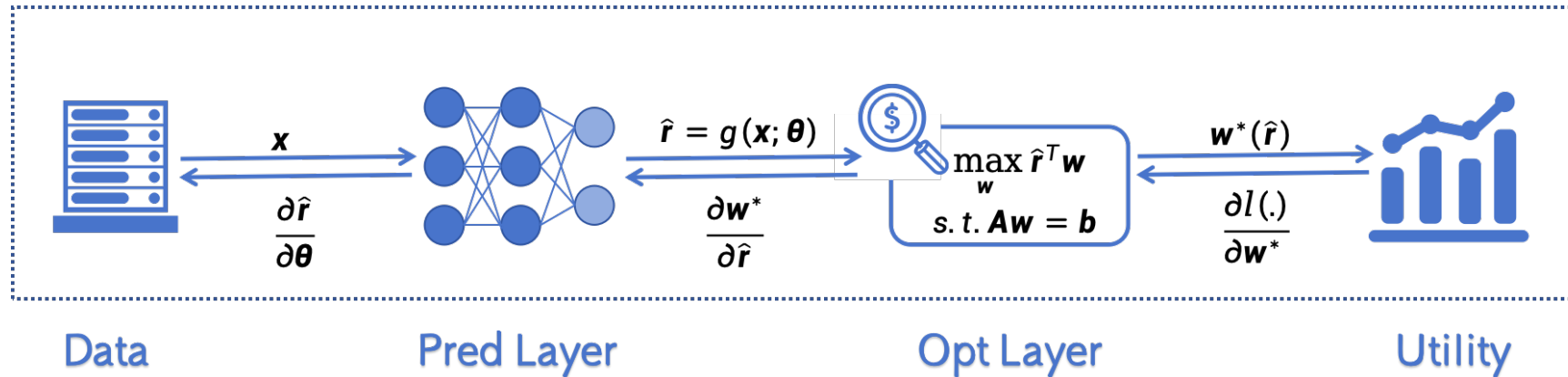
The Portfolio Manager

- Reacts to the model's return predictions $\hat{r} = g(x_t, \theta)$.
- Goal: Construct the optimal portfolio w^* according to the predicted return $\hat{r}$.

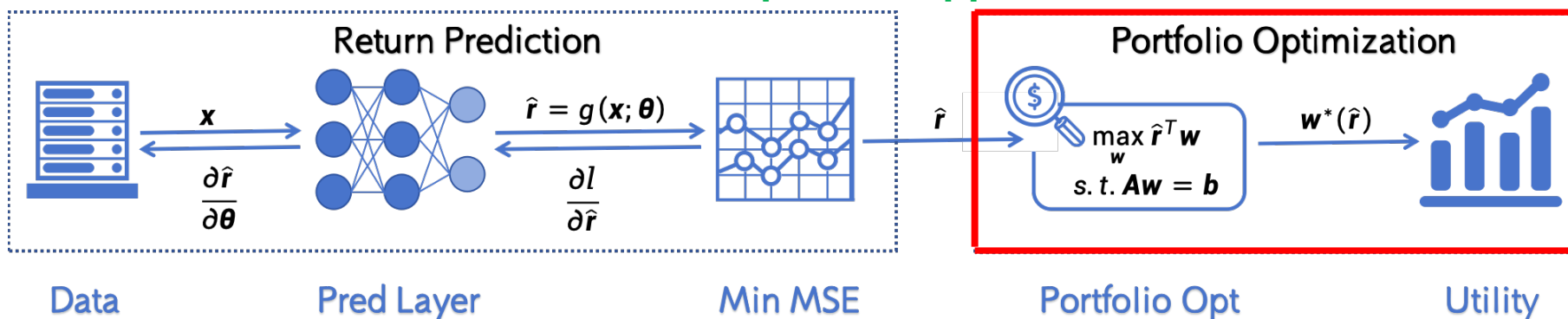
Our Method: The E2E Neural Network

Unlike PPP and AlphaPortfolio, our E2E framework integrates a formal **portfolio optimization layer** — making feasibility part of the learning architecture, not an afterthought.

Our E2E Neural Network Approach

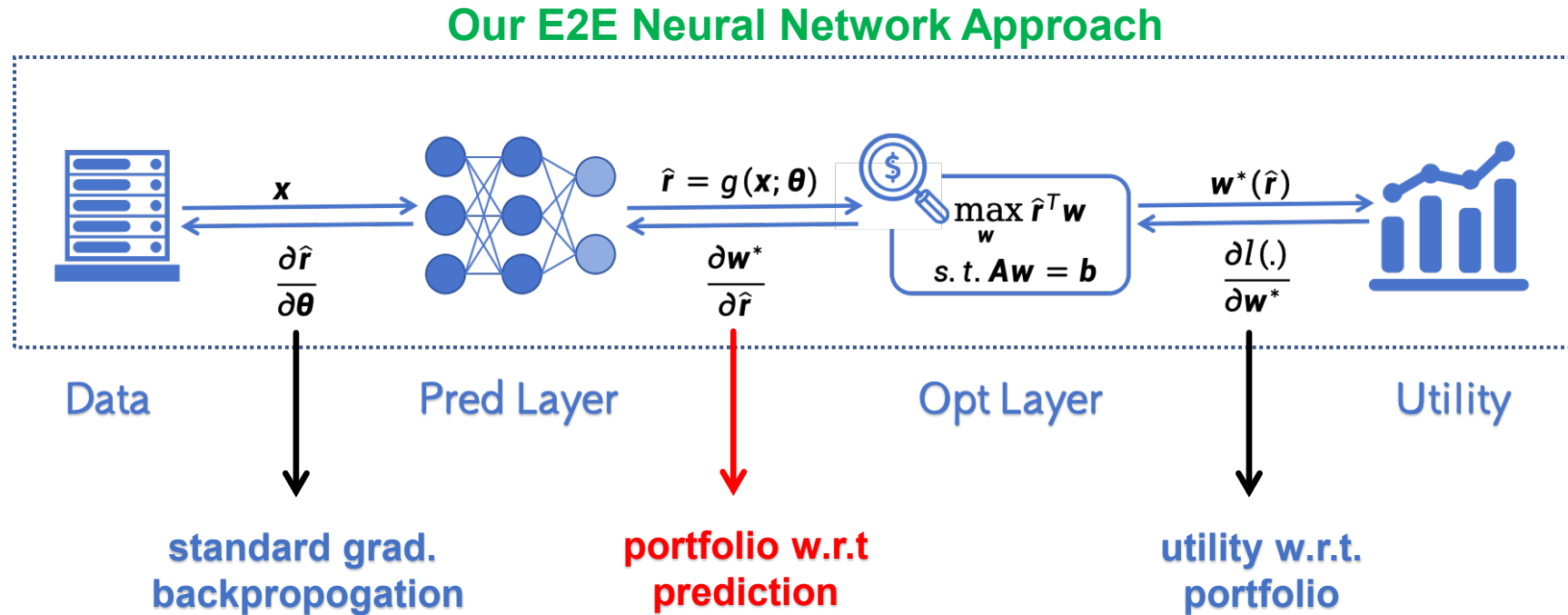


Predict then Optimize Approach



Challenge: The Backward Pass

The gradient of the predictive parameter θ can be calculated via the chain rule.



- We build on recent advances in CS to measure how portfolio weights respond to return forecasts, so the prediction model can be trained from final portfolio performance. On top of this foundation, we add several designs for practical portfolio construction.

Data: A Large-Scale, Higher-Frequency Dataset

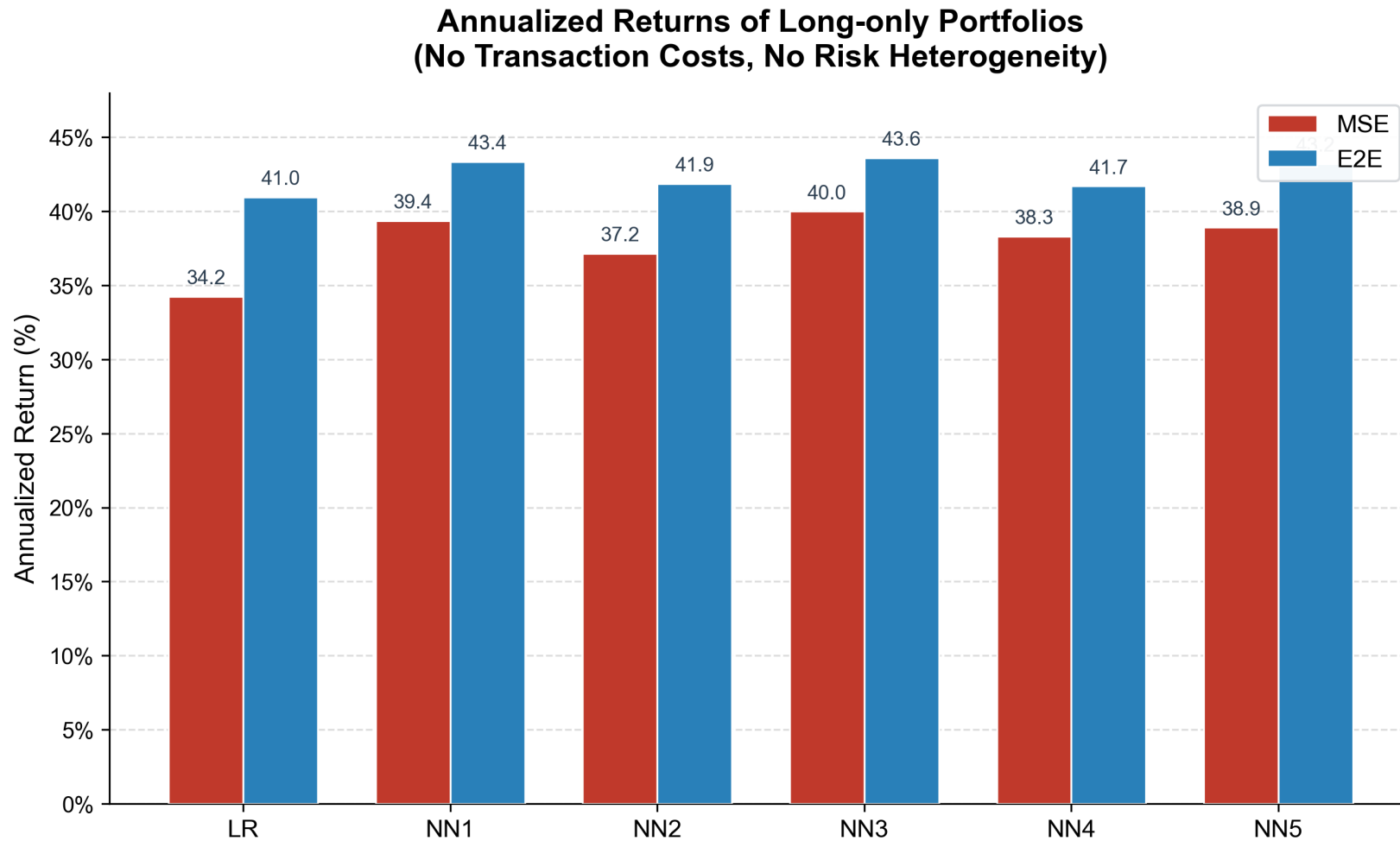
Dataset Specifications

- **Market:** China A-share market
- **Time Period:** January 2010 – December 2023 (14 years)
- **Universe:** 5,489 stocks
- **Characteristics:** We employ a large collection of 415 firm-level characteristics.

Scale of Analysis

The combination of a large stock universe and the time frame results in approximately **6 million stock-day observations**.

Test 1: Performance under Long-Only Constraints



Interpretation

- Stocks are sorted into 10 deciles based on predicted returns.
- We assign equal weights to the top decile as long-only portfolios.
- We compare MSE vs. E2E for Linear Regression (LR) and five Neural Network (NN1-NN5) models.

E2E Appears to Deliver Superior Performance

Quantifying the Improvement

- Across the six models, E2E improves the average annualized portfolio return of the High group.
- E2E achieves a **10% relative improvement** in both average returns and Sharpe ratios.

Statistical Significance

The outperformance of E2E is **statistically significant** at the 5% level or better across all six models.

Test 2: Transactions Costs

High portfolio turnover leads to higher transaction costs (TCs), eroding profitability. To evaluate our framework under these frictions, we compare three strategies.

Framework 1: MSE-Top

The Baseline. A long-only portfolio formed directly from the raw predictions of the standard MSE model, without any cost optimization.

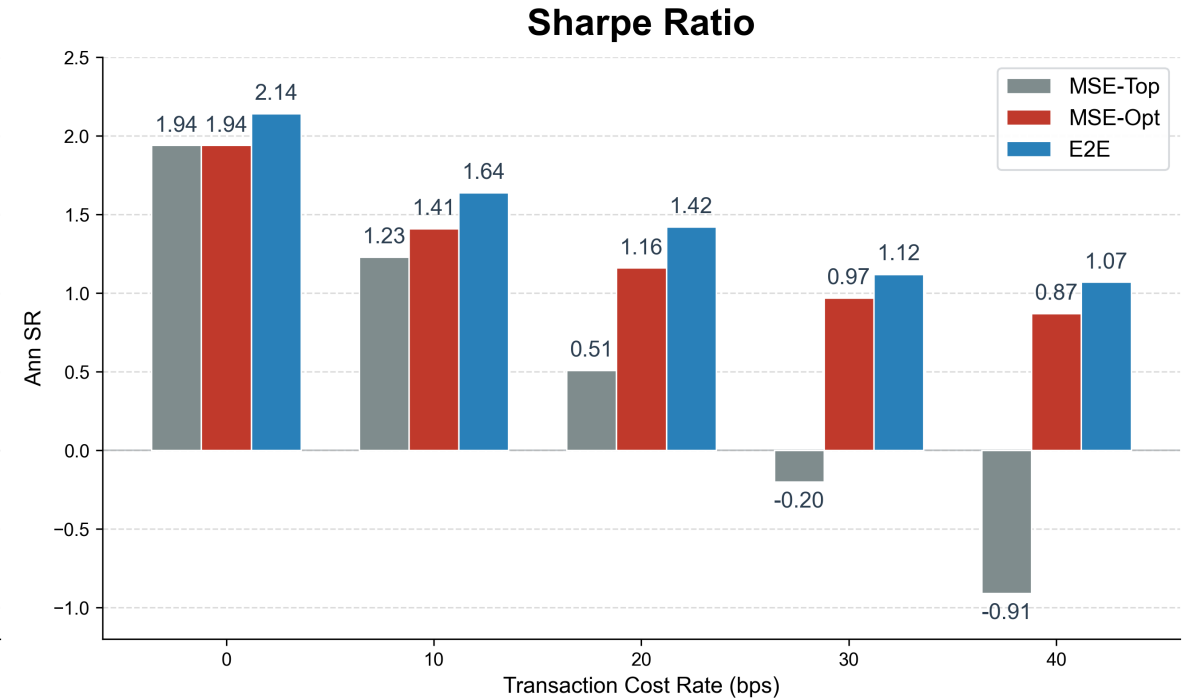
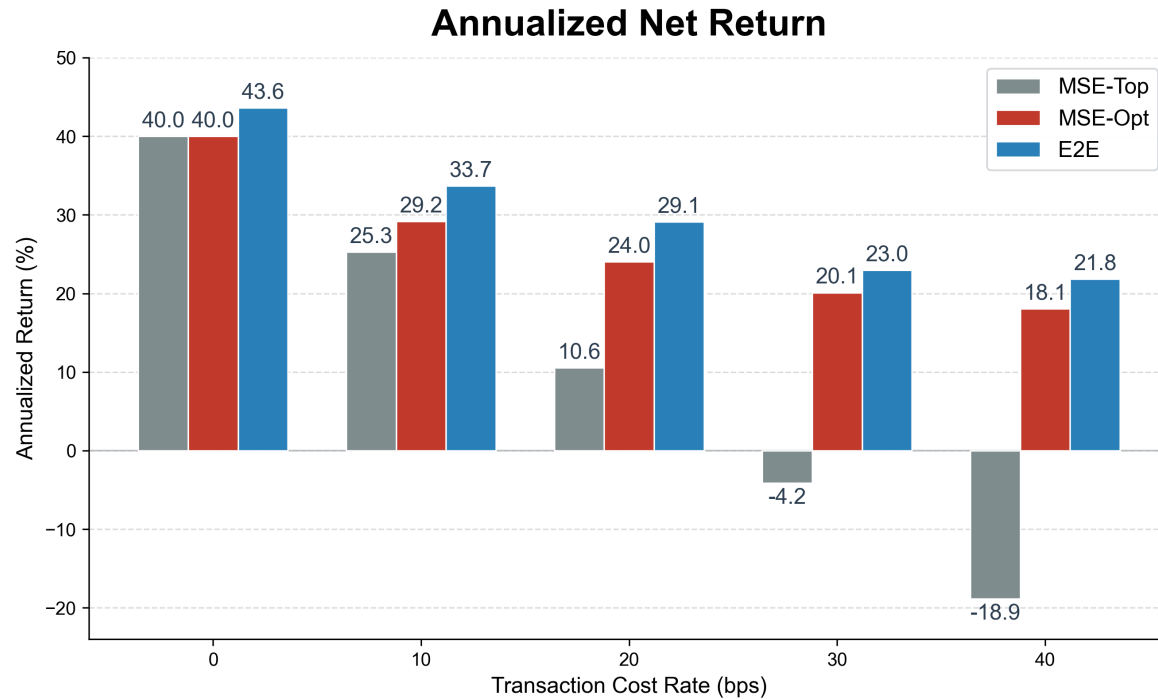
Framework2: MSE-Opt

The Industry Standard. The MSE signal is followed by a second-stage optimization that penalizes turnover to manage transaction costs. This mimics common industry practice.

Framework 3: E2E (Our Model)

The Integrated Approach. Our proposed end-to-end framework, which incorporates cost-awareness *directly into the training* of the prediction function.

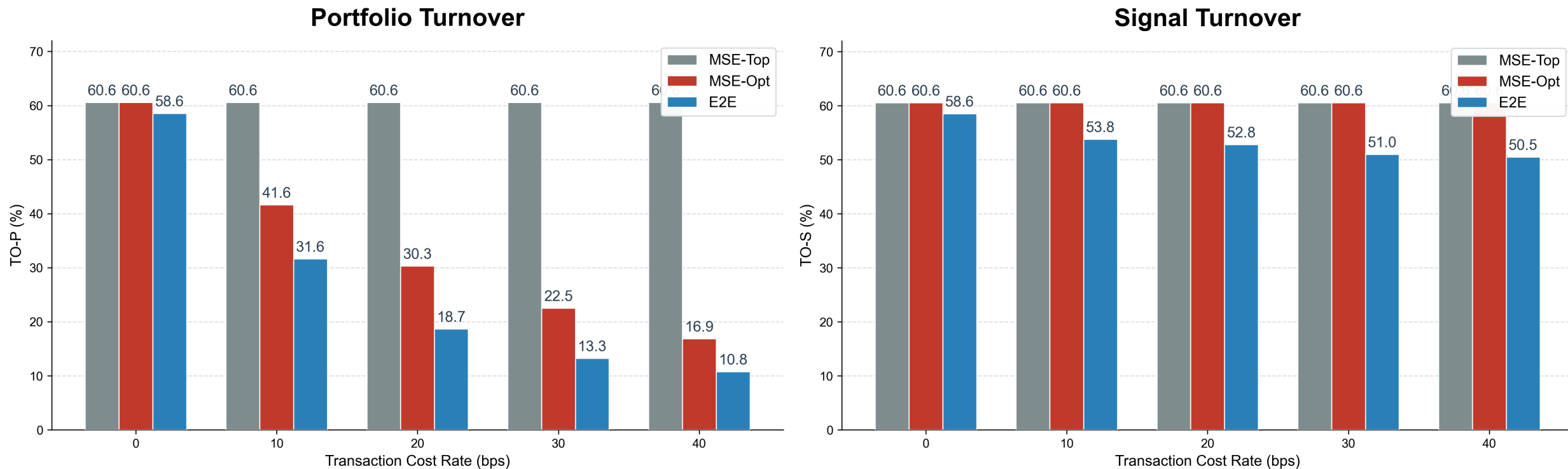
Test 2: Transactions Costs



E2E Consistently Outperforms in Backtest

The E2E framework delivers higher net returns and Sharpe ratios than **both** MSE-based benchmarks across **all** cost settings.

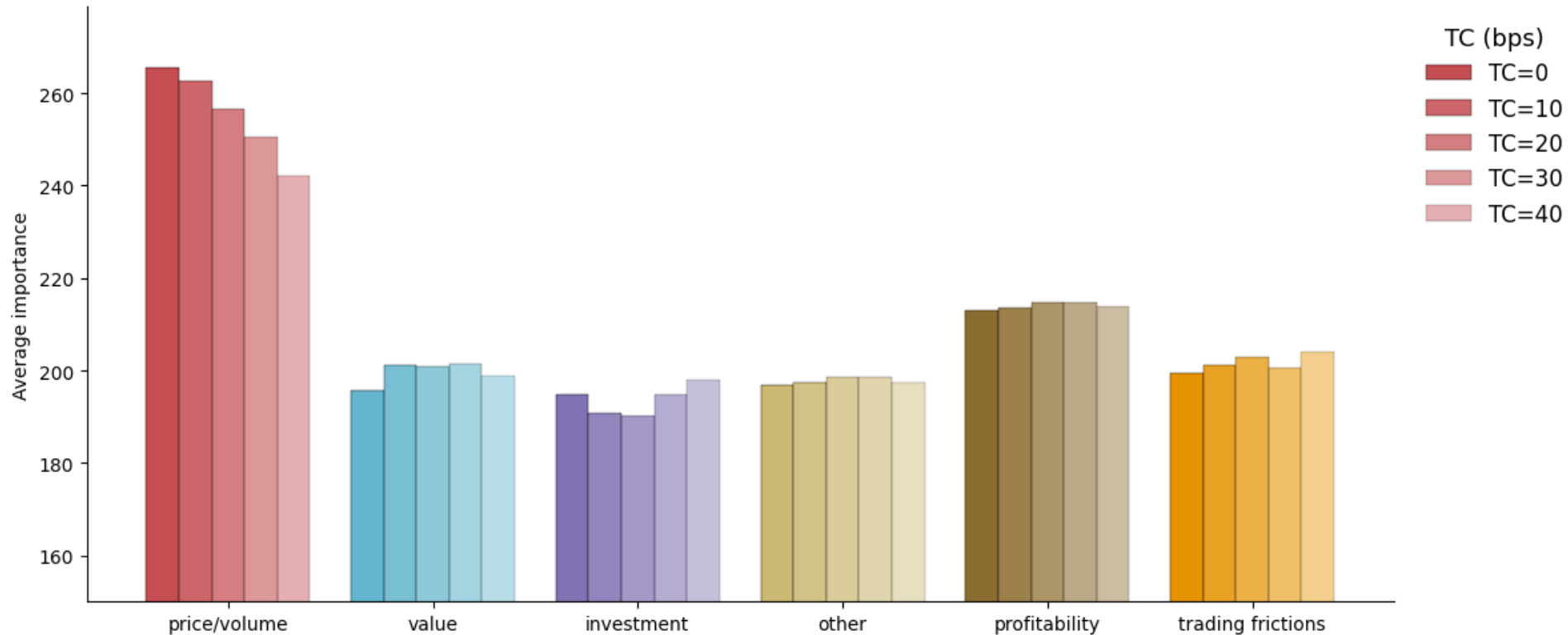
Test 2: Transactions Costs



A Smarter Adaptation Mechanism

- As TCs rise, both MSE-Opt and E2E reduce portfolio turnover (*actions*).
- However, only E2E also reduces *signal* turnover. It learns to generate **lower-turnover signals** from the start, because it is aware of the downstream.

Test 2: Factor Importance



Adaptive Factor Importance

E2E model down-weights high-turnover Price/Volume factors and shifts weight toward more stable fundamental signals.

Test 3: Heterogeneous Risk Preferences

The traditional two-stage approach assumes a **single** prediction model for all investors. Individual risk preferences are only applied in the final, separate optimization step.

- This overlooks the fact that risk preferences can influence not only the weighting of assets, but also how return signals are interpreted.

Our Solution: The Risk-Aware E2E Model

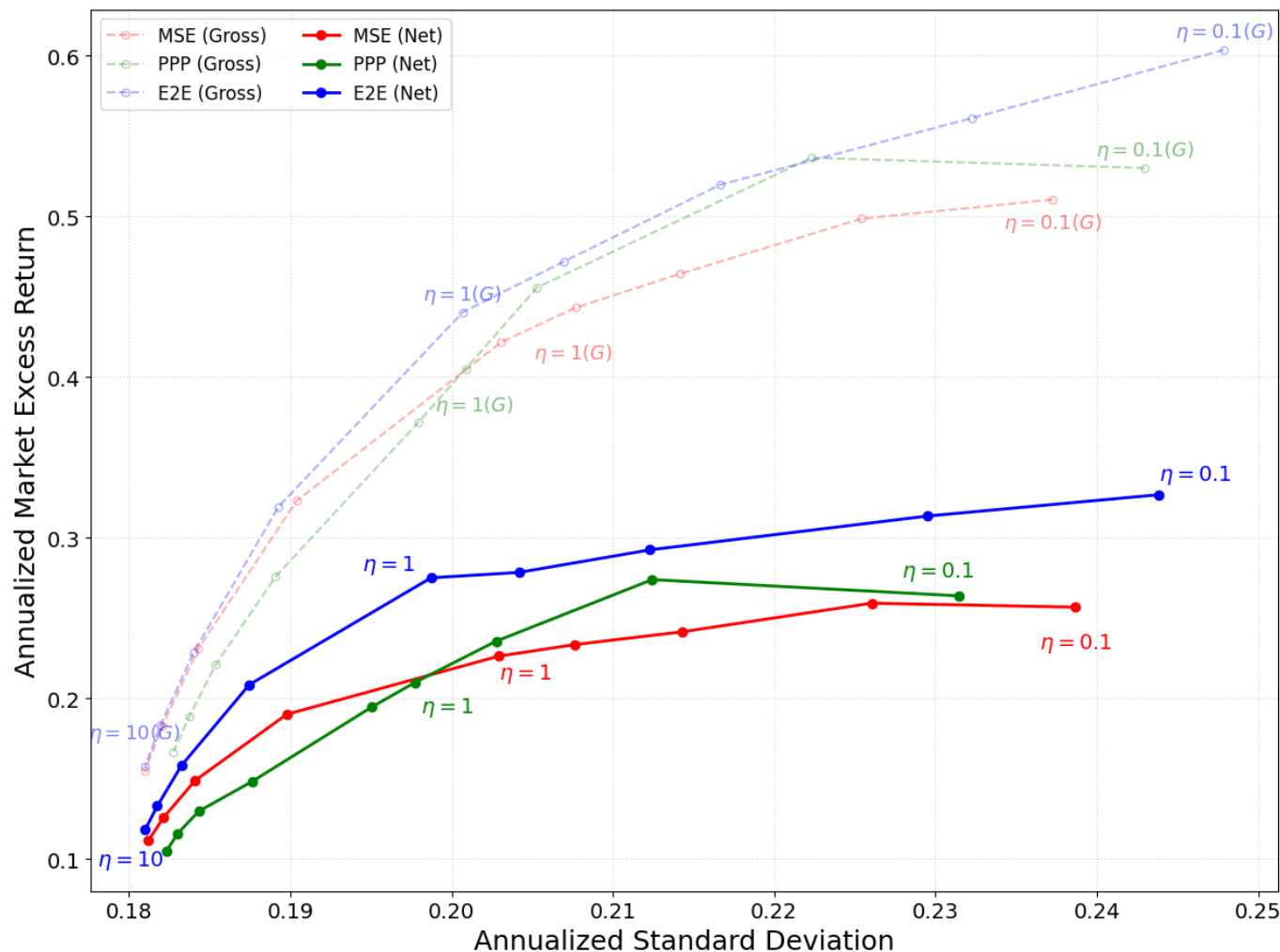
- We further embed risk preference (η) into the E2E framework. The model learns a tailored pricing function that aligns with the investment utility

$$w^T r - \eta \sqrt{w^T \Sigma w}$$

- We examine different risk preference level from 0.1 to 10.
- We adopt a diagonal covariance estimation for illustration only.

A New Efficient Frontier

Training prediction models with different risk-aversion levels (η), we trace out the realized efficient frontiers for the MSE, PPP, and E2E.



Future Directions: Richer Risk and Preferences

Two natural extensions would sharpen the risk model and broaden the preference specification.

Full Covariance Matrix

- Our current setup adopts a diagonal covariance matrix.
- A full covariance (or a structured estimator: factor, shrinkage) may capture cross-asset dependence more accurately.

Alternative utility function

Current training centers on mean–variance utility. The same E2E architecture can host other objectives:

- Sharpe ratios
- CVaR-style preferences

Future Directions: Faster Algorithms for E2E Learning

In our E2E framework, each forward pass must solve a constrained portfolio optimization problem. Relative to standard MSE training, this makes each iteration more expensive.

- Design solvers / architectures that accelerate the optimization layer without abandoning hard constraints, for example:
 - warm starts and incremental re-optimization across adjacent periods
 - parallelization and better batching of cross-sectional solves
- Goal: keep the E2E - feasibility written into the model - while making training and inference scalable.

Future Directions: Security-Specific Transaction Costs

Trading frictions enter the E2E objective through a *uniform* transaction cost rate applied to all stocks.

- In practice, costs depend on:
 - liquidity
 - size (impact)
 - volatility
 - trading intensity.
- Replace the uniform rate with **security-specific** costs
- Enables the model to learn both what to predict and what is cheap to trade.



E-mail comments to cam.harvey@duke.edu (thanks!)