A Reproducible Benchmark for Machine Learning in Cross-Sectional Quantitative Trading under Realistic Costs and Regime Shifts

Zhang Yuchen[a]

[a] *AI, Ethics and Society Programme, Faculty of Arts, The University of Hong Kong, Hong Kong SAR, China*

Abstract

Machine learning (ML) models for quantitative trading are routinely evaluated under conditions that inflate reported performance: costs are ignored, splits leak future information, and statistical significance is not tested. We present **ML Trading Bench**, a unified evaluation protocol and open-source toolkit that applies walk-forward splitting with embargo, a configurable transaction-cost model, and rigorous statistical testing to a reproducible cross-sectional trading setting. Using 50 US-listed ETFs over 2005–2024, we evaluate 9 models (linear, tree, and deep-learning families) plus 2 passive baselines across 5 cost scenarios, 4 market regimes, and multiple strategy hyperparameter configurations.

Our key findings challenge common claims in the literature: (i) all bootstrap 95% confidence intervals on gross Sharpe ratios include zero; (ii) at 15 bps one-way cost, every long-short strategy turns deeply negative; (iii) after Benjamini–Hochberg FDR correction, no ML-vs-ML pair is statistically distinguishable under the Diebold–Mariano test; (iv) strategy hyperparameters

Email address: `u3663696@connect.hku.hk` (Zhang Yuchen)

URL: `https://github.com/georgekingsman` (Zhang Yuchen)

(rebalance frequency, portfolio concentration) shift Sharpe by >0.5—often more than model choice; (v) regime decomposition reveals that COVID-era volatility drives most headline results.

The full pipeline—from data download to 9 tables and 9 figures—runs in under 6 minutes with a single command (`python run_all.py`). Code, data scripts, and configuration are released under MIT license at `https://github.com/georgekingsman/ml-trading-benchmark`.

Keywords: quantitative trading, machine learning benchmark, reproducibility, transaction costs, regime analysis, walk-forward evaluation, statistical testing

1. Introduction

Machine learning is now central to quantitative trading, supporting signal discovery, portfolio construction, and execution under uncertainty and frictions [1, 2, 3]. We treat ML-based quantitative trading as an *engineering system evaluation problem*, where protocol choices—data splits, cost assumptions, leakage controls, and multiple-testing corrections—dominate apparent performance differences between models. Yet a large fraction of reported "alpha" disappears once evaluation is made realistic. Common pitfalls include: look-ahead leakage through naive train/test splits, omission of transaction costs, lack of statistical significance testing, and neglect of regime-dependent fragility [4, 5, 6].

These evaluation gaps have practical consequences. A model that appears to deliver Sharpe 1.0 in a zero-cost backtest may produce Sharpe -1.5 once realistic turnover costs are applied. A model comparison that appears significant may lose all significance after multiple-testing correction. Without controlled benchmarks, practitioners and reviewers cannot distinguish genuine progress from evaluation artifacts.

Contributions.. This paper makes three contributions:

1. **A unified evaluation protocol** that combines walk-forward splitting with embargo, rolling z-score normalisation using only training data, and a configurable fee-plus-slippage cost model—designed to prevent the most common pitfalls that inflate reported performance.

2. **A one-click reproducible toolkit**: the command `python run_all.py` executes the full data→train→backtest→report pipeline, producing 9 tables and 9 figures with zero manual intervention in under 6 minutes.

3. **Systematic empirical evidence** on a realistic ETF universe: cost sensitivity analysis across 5 scenarios, per-regime performance decomposition across 4 market regimes, hyperparameter sensitivity studies (rebalance frequency, portfolio concentration), and rigorous statistical testing via bootstrap confidence intervals and the Diebold–Mariano test [7] with Benjamini–Hochberg FDR correction [8].

Paper organisation.. Section 2 reviews related work. Section 3 describes the benchmark design (universe, features, splits, cost model). Section 4 presents the model families and strategies. Section 5 details the evaluation methodology and statistical tests. Section 6 reports results. Section 7 describes the reproducibility package. Section 8 discusses implications and limitations.

2. Related Work

ML for trading surveys.. Several surveys review ML methods for financial prediction and trading [9, 10], reinforcement learning for portfolio management [1, 11], and deep learning for asset pricing [12]. These works provide taxonomies and broad coverage but typically do not include reproducible benchmarks or systematic cost/regime analysis.

Quantitative trading platforms.. Qlib [13] (Microsoft) provides an end-to-end quant research platform with data handling, model training, and backtesting for Chinese/US equities. FinRL [14] focuses on reinforcement learning with a gym-style interface. Both are powerful frameworks but are primarily designed for practitioners building new strategies, rather than for controlled evaluation of existing model families under varying cost and regime assumptions.

Evaluation methodology.. De Prado [5, 6] introduced purged cross-validation and embargo-based splits to prevent leakage in financial ML. The Diebold–Mariano test [7] is widely used for comparing forecast accuracy. Benjamini and Hochberg [8] proposed FDR control for multiple hypothesis testing. Harvey et al. [15] argued that many reported trading "factors" are spurious due to multiple testing. Our benchmark integrates these methodological innovations into a unified, automated pipeline.

Positioning.. Unlike survey papers that prescribe best practices, our work *demonstrates their consequences* in a concrete, reproducible setting. Unlike trading platforms, our focus is on controlled evaluation rather than strategy development. The closest analogue is a standardised evaluation benchmark—adapted to the unique challenges of financial time series, in which non-stationarity, transaction costs, and regime shifts make controlled comparison especially difficult.

3. Benchmark Design

3.1. Universe and Data

We select 50 US-listed ETFs spanning equity sectors (SPY, QQQ, XLF, XLE, XLK, etc.), fixed income (TLT, IEF, HYG), commodities (GLD, USO), and currencies (UUP, FXE). The ETF universe avoids individual-stock survivorship bias: all 50 ETFs remain listed throughout the full sample period (January 2005 to December 2024). Daily OHLCV data are obtained from Stooq (primary; no API key, no rate limit) with yfinance as fallback.

Data statement.. All data used in this study are publicly accessible and require no paid subscription or institutional license. The primary source is Stooq (`https://stooq.com`), which provides adjusted daily OHLCV for US-listed ETFs; yfinance (`https://pypi.org/project/yfinance/`) serves as a fallback. The universe comprises 50 ETFs across equity, fixed-income, commodity, and currency sectors, covering the period January 2005 to December 2024. Stooq data are freely redistributable for non-commercial research; yfinance data are subject to Yahoo Finance terms of service. We do not use point-in-time fundamental data; all features are derived from price and volume (see below). Corporate actions (splits, dividends) are handled by the data provider's adjustment. Missing data (delistings, holidays) are forward-filled for at most 5 days; tickers with $>10\%$ missing days are excluded. The complete download-and-processing pipeline is included in the released code, enabling full replication from raw data.

3.2. Features and Labels

We engineer 13 technical features per ticker per day:

- **Returns**: 1-day, 5-day, 20-day log returns

- **Volatility**: 20-day and 60-day rolling standard deviation
- **Momentum**: 10-day and 20-day momentum (cumulative return)
- **RSI**: 14-day relative strength index
- **Moving-average ratios**: close/MA(10) and close/MA(50)
- **Volume**: 20-day volume ratio (current/rolling average)
- **Intraday range**: (high − low) / close

All features are rolling z-score normalised using a **strictly trailing 252-day window**. This is critical: normalisation statistics are computed only on past data, preventing any leakage of future distributional information.

The prediction target is the **5-day forward return** (cross-sectional; used for ranking, not regression accuracy).

3.3. Walk-Forward Split with Embargo

- **Training**: June 2005 – December 2016 (12 years)
- **Validation**: January 2017 – December 2019 (3 years)
- **Test**: January 2020 – December 2024 (5 years)
- **Embargo**: 5 trading days at each boundary

The embargo gap removes potential label-overlap leakage between adjacent periods. All hyperparameters are selected on the validation set; the test set is **never** used for model selection. Figure 1 illustrates the protocol.

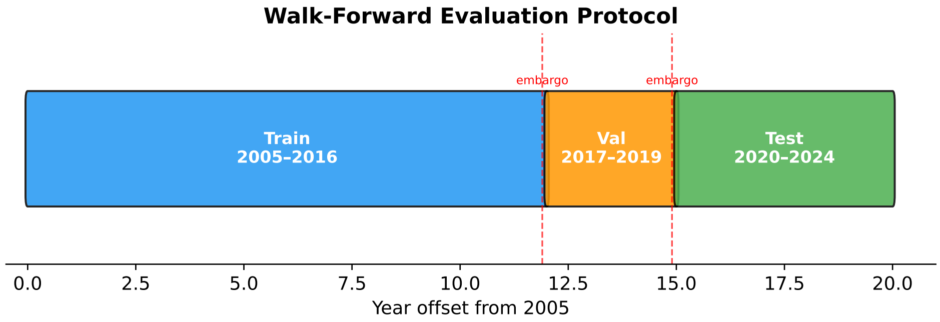

Figure 1: Walk-forward evaluation protocol with embargo gaps. No information from downstream periods can influence upstream training or normalisation.

3.4. Cost Model

Transaction costs are modelled as:

$$\text{cost}_t = (\text{fee} + \text{slippage}) \times \sum_i |\Delta w_{i,t}|, \tag{1}$$

where $\Delta w_{i,t}$ is the weight change for asset i at rebalance time t. We evaluate five cost scenarios: 0, 5, 10, 15, and 25 bps one-way (with an additional 5 bps slippage). This range spans from optimistic institutional settings to retail-level costs.

4. Models and Strategies

4.1. Model Families

We evaluate 9 models spanning three families discussed in the quantitative trading literature, plus 2 passive benchmarks:

Traditional ML (5 models)..

- **Linear Regression** and **Ridge**: standard baselines with L2 regularisation.

- **Logistic Regression**: predicts direction probability, converted to a continuous signal.
- **Random Forest**: 200 trees, max depth 10.
- **LightGBM**: gradient-boosted trees with 500 rounds and early stopping on the validation set.

Deep Learning (2 models)..

- **MLP**: 2-layer feedforward network (128–64 hidden units), ReLU activation, 50 epochs.
- **LSTM**: 2-layer LSTM (hidden dim 64, sequence length 20), 50 epochs.

Naive Strategies (2 baselines)..

- **Momentum Baseline**: ranks assets by trailing 20-day return.
- **Mean Reversion Baseline**: ranks assets by negative 5-day return.

Ensemble.. We construct a rank-average ensemble of all ML models: for each date, we compute the cross-sectional percentile rank of each model's prediction, then average across models.

Passive Benchmarks..

- **SPY Buy-and-Hold**: 100% allocation to the S&P 500 ETF.
- **Equal Weight (1/N)**: daily equal-weight allocation across all 50 ETFs.

These passive benchmarks incur zero turnover and serve as the "minimum bar" against which active strategies must be judged.

4.2. Strategy Construction

At each rebalance date (default: every 5 trading days), assets are ranked by predicted signal. The **long-short strategy** goes long the top-K and short the bottom-K with equal weights ($\pm 1/K$ per leg; default $K = 10$). The **long-only variant** holds only the top-K with equal weights.

5. Evaluation Methodology

5.1. Performance Metrics

- **CAGR**: compound annual growth rate (gross and net of costs)
- **Sharpe ratio**: annualised (gross and net), assuming zero risk-free rate
- **Maximum drawdown**: largest peak-to-trough decline
- **Calmar ratio**: CAGR / max drawdown
- **Hit rate**: fraction of positive-return days
- **Average turnover**: inferred from cost series

5.2. Signal-Level Metrics

- **Information Coefficient (IC)**: daily cross-sectional Spearman rank correlation between predictions and realised 5-day returns.
- **ICIR**: IC divided by its standard deviation across days; measures signal stability.

5.3. Bootstrap Confidence Intervals

We compute 95% confidence intervals on the gross Sharpe ratio via block bootstrap ($B = 1{,}000$ resamples) to assess whether any model's performance is statistically distinguishable from zero.

5.4. Diebold–Mariano Test with FDR Correction

We apply the two-sided Diebold–Mariano (DM) test [7] pairwise across all $\binom{n}{2}$ model pairs using daily gross returns as the loss differential, with Newey–West HAC standard errors (bandwidth $h = 5$).

Multiple testing correction.. With $n = 12$ models, we have $\binom{12}{2} = 66$ pairwise comparisons. Given this large number of simultaneous tests, testing each at $\alpha = 0.05$ without correction would inflate the family-wise Type I error rate substantially, making spurious "significant" differences likely even when no true performance gap exists. We therefore apply the Benjamini–Hochberg (BH) procedure [8] to control the false discovery rate (FDR) at 5%. This is a one-line addition to the pipeline but significantly strengthens the statistical rigour of model comparisons and aligns with best practices advocated by Harvey et al. [15] for factor evaluation in finance.

5.5. Regime Decomposition

We partition the test period into four macro-regimes based on well-known market events:

1. **COVID Crash**: February 2020 – June 2020

2. **Recovery**: July 2020 – December 2021

3. **Rate Hikes**: January 2022 – December 2022
4. **Normalisation**: January 2023 – December 2024

For each model, we report gross Sharpe within each regime to reveal whether headline performance is driven by a single extreme period.

5.6. Hyperparameter Sensitivity

We systematically vary two strategy hyperparameters that receive little attention in the literature:

- **Rebalance frequency**: 1, 5, 10, 20 trading days
- **Portfolio concentration (top-K)**: 3, 5, 10, 15, 20 assets

This tests whether conclusions are robust to "nuisance" strategy parameters, or whether these parameters dominate model choice.

6. Results

6.1. Main Results

Table 1 reports the core metrics on the test period (January 2020 – December 2024).

Table 1: Main benchmark results. Sharpe CI denotes the 95% bootstrap confidence interval on the gross Sharpe ratio. IC and ICIR are computed cross-sectionally.

Model	CAGR	Sharpe	Sharpe	Max DD	IC	ICIR	Sharpe 95% CI	
	(g, %)	(gross)	(net)	(%)			lo	hi
BuyAndHold_SPY	*14.86*	*0.765*	*0.765*	*33.72*	—	—	*−0.12*	*1.68*
EqualWeight	*6.82*	*0.515*	*0.515*	*26.32*	—	—	*−0.39*	*1.45*
LogisticRegression	7.04	0.469	−1.44	21.33	0.010	0.037	−0.43	1.35
LinearRegression	6.21	0.432	−1.46	18.93	0.015	0.051	−0.48	1.35
RandomForest	4.47	0.372	−1.56	28.55	0.005	0.019	−0.56	1.29
Ensemble	4.73	0.347	−1.48	22.66	0.011	0.040	−0.57	1.27
MLP	3.30	0.294	−1.44	26.00	0.005	0.024	−0.57	1.09
LightGBM	3.34	0.290	−1.51	24.02	0.007	0.028	−0.65	1.20
MomentumBaseline	0.67	0.134	−0.69	39.44	−0.011	−0.036	−0.75	1.01
LSTM	−3.53	−0.167	−2.00	25.95	0.009	0.042	−1.07	0.74
MeanReversion	−4.69	−0.134	−0.97	44.77	0.011	0.036	−1.01	0.75

Observations..

1. Most ML models exhibit weakly positive cross-sectional IC ($\sim$0.005–0.015), confirming marginal predictive content; MomentumBaseline's IC is slightly negative (-0.011).

2. Every active strategy's gross Sharpe falls below SPY buy-and-hold (0.765) and most fall below equal-weight (0.515).

3. At 15 bps cost, all long-short strategies produce deeply negative net Sharpe ratios.

4. **All bootstrap 95% CIs include zero**—no model's gross performance is statistically distinguishable from zero at the 5% level.

6.2. Cost Sensitivity

Figure 2 plots net Sharpe against one-way transaction cost. The “alpha cliff” is evident: even at 5 bps, most models turn negative.

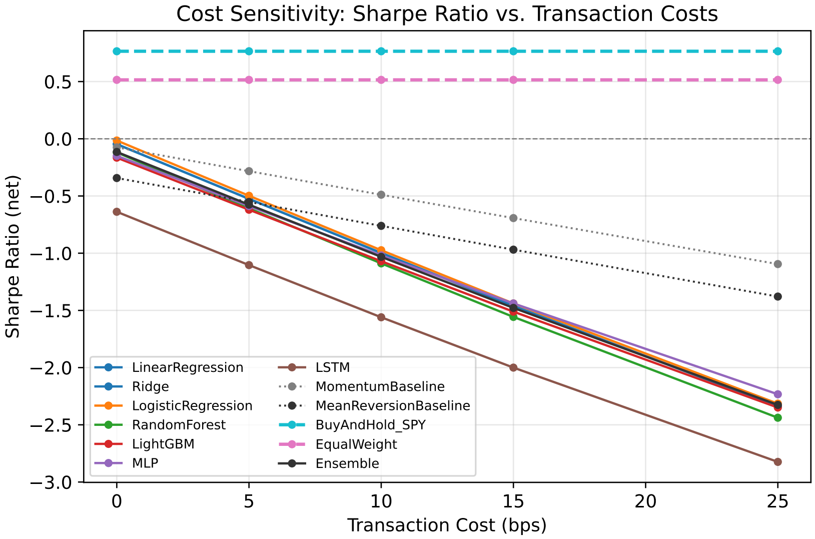

Figure 2: Net Sharpe ratio vs. one-way transaction cost (bps). Passive benchmarks are flat because they incur zero turnover. The steep decline illustrates the “alpha cliff”: small costs erase small signals.

6.3. Regime Analysis

Table 2 decomposes performance across four regimes.

Table 2: Gross Sharpe ratio by regime.

Model	COVID Crash	Recovery	Rate Hikes	Normalisation
BuyAndHold_SPY	$+0.08$	$+2.19$	-0.71	$+1.93$
EqualWeight	-0.04	$+2.05$	-0.68	$+1.07$
LogisticRegression	$+0.74$	$+1.47$	$+0.69$	-0.53
RandomForest	$+1.87$	$+0.24$	$+0.38$	-0.19
LightGBM	$+1.41$	$+0.54$	$+0.23$	-0.30
MLP	$+1.21$	$+0.42$	$+0.17$	$+0.05$
LSTM	$+0.23$	-0.18	-0.22	-0.18
Ensemble	$+1.31$	$+0.86$	$+0.47$	-0.57
MomentumBaseline	$+1.02$	$+0.88$	-1.44	-0.19

Observations.. Active models dramatically outperform buy-and-hold during the COVID crash (long-short benefits from elevated volatility and cross-sectional dispersion), but underperform in trending markets (recovery, normalisation). This regime sensitivity is precisely the evaluation gap that headline Sharpe ratios hide. Momentum collapses during rate hikes (-1.44), consistent with well-documented factor crashes.

6.4. Hyperparameter Sensitivity

Table 3 reports gross Sharpe under varying rebalance frequencies and top-K values.

Table 3: Gross Sharpe ratio under different rebalance frequencies (days) and top-K values.

	Rebalance Frequency (days)				Top-K				
Model	1	5	10	20	3	5	10	15	20
LogisticRegression	0.66	0.47	0.15	0.34	0.95	1.03	0.47	0.60	0.58
LightGBM	0.31	0.29	0.31	0.18	−0.08	0.10	0.29	0.20	0.12
MLP	0.31	0.29	−0.16	−0.67	0.55	0.42	0.29	0.38	0.22
LSTM	0.63	−0.17	−0.29	−0.82	0.35	0.16	−0.17	−0.08	−0.26
Ensemble	0.73	0.35	0.22	0.12	0.61	0.37	0.35	0.42	0.46

Observations.. Daily rebalancing yields substantially higher gross Sharpe for most models (Ensemble: 0.73 vs. 0.35 at 5-day), but this comes with proportionally higher turnover—and therefore worse net performance. More concentrated portfolios (smaller K) amplify signal quality: Logistic Regression achieves Sharpe 1.03 at $K = 5$ versus 0.47 at $K = 10$, but at higher idiosyncratic risk. **These hyperparameters shift Sharpe by >0.5—more than the difference between model families.** Yet they are rarely stress-tested in ML trading papers.

6.5. Statistical Significance

We apply the DM test [7] pairwise across all 12 models (66 pairs).

Raw results.. At $\alpha = 0.05$, only **3 pairs** are significant, and all involve the passive SPY benchmark. No ML-vs-ML comparison achieves significance.

After BH correction.. Applying Benjamini–Hochberg FDR correction [8] at the 5% level, **no pairs remain significant** (0/66); even the 3 raw-significant passive pairs do not survive correction. No ML-vs-ML pair survives correction.

This result carries a stark implication: **model comparisons in the literature, when evaluated under proper cost and split protocols, may not be meaningfully distinguishable.**

6.6. Long-Only Variant

Table 4 compares long-short and long-only strategies.

Table 4: Long-short (LS) vs. long-only (LO) comparison at 15 bps.

	Long-Short			Long-Only		
Model	Sharpe(g)	Sharpe(n)	CAGR(g,%)	Sharpe(g)	Sharpe(n)	CAGR(g,%)
LogisticRegression	0.47	−1.44	7.04	0.68	−0.35	10.57
LightGBM	0.29	−1.51	3.34	0.65	−0.13	10.55
MLP	0.29	−1.44	3.30	0.62	−0.19	9.88
LSTM	−0.17	−2.00	−3.53	0.22	−0.54	2.44
Ensemble	0.35	−1.48	4.73	0.65	−0.34	10.19
BuyAndHold_SPY	0.77	0.77	14.86	—	—	—

Long-only consistently produces higher gross Sharpe (e.g., LightGBM: 0.65 vs. 0.29) because it captures the long-run equity premium rather than relying solely on cross-sectional spread. This suggests that, for most ML signals in this setting, the short leg destroys more value than it creates.

6.7. Equity Curves

Figure 3 shows cumulative gross returns with a drawdown subplot and regime shading.

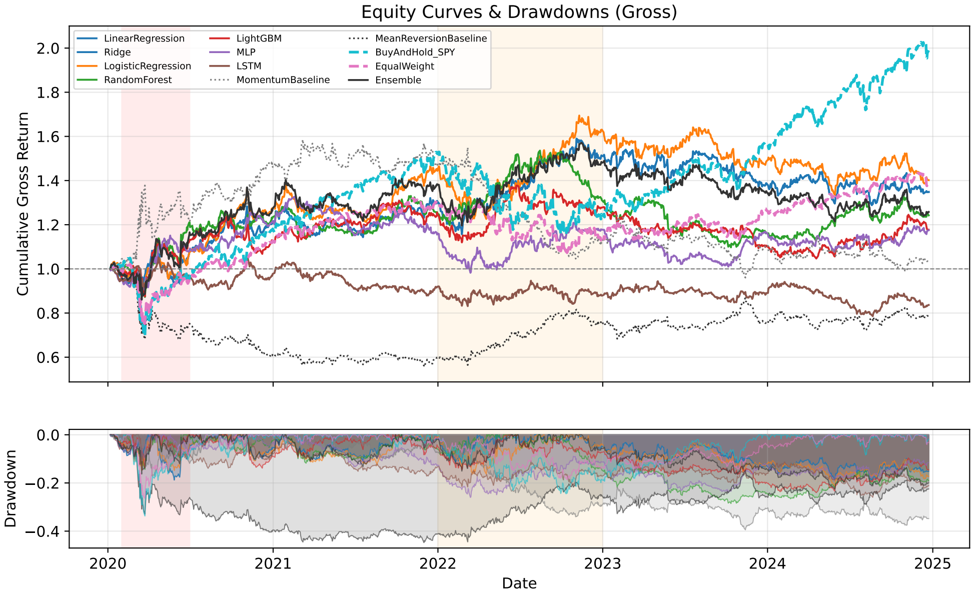

Figure 3: Cumulative gross returns with drawdown subplot. Shaded bands: COVID crash (red), rate hikes (orange). Passive benchmarks shown as dashed lines.

7. Reproducibility Package

7.1. Pipeline Overview

The full benchmark is executed via:

```
pip install -r requirements.txt
python run_all.py
```

This runs 13 sequential steps: data download, feature engineering, walk-forward split, model training, backtesting (long-short + long-only + passives), IC/ICIR/bootstrap CI, regime analysis, feature importance, ensemble construction, DM test with BH correction, rebalance sensitivity, top-K sensitivity, and report generation.

7.2. Runtime and Environment

- **Runtime**: $\sim$6 minutes on Apple M-series (M2, 16 GB RAM)
- **Python**: ≥ 3.10
- **Key dependencies**: pandas, scikit-learn, LightGBM, PyTorch, statsmodels
- **Random seed**: fixed at 42 for full reproducibility
- **Data**: freely available from Stooq (no API key required)

7.3. Output

The pipeline produces:

- **9 tables**: main results, cost sensitivity, regime analysis, feature importance, long-only comparison, DM test (raw + BH-corrected), rebalance sensitivity, top-K sensitivity (CSV + LaTeX)
- **9 figures**: walk-forward timeline, cost sensitivity curves, equity curves, ranking heatmap, feature importance, regime bars, rebalance sensitivity, top-K sensitivity, DM heatmap (PDF)
- **JSON summary**: all numerical metrics in machine-readable format

7.4. Code and License

All code is released under MIT license at `https://github.com/georgekingsman/ml-trading-benchmark`. The repository includes `REPRODUCIBILITY.md` (step-by-step instructions), `ENVIRONMENT.md` (platform notes), and `CITATION.cff` (machine-readable citation).

Shortest reproduction path.. After installing dependencies (`pip install -r requirements.txt`), a single command reproduces every result in this paper:

`python run_all.py` ⟶ Tables 1–4 + Figures 1–3 in ∼6 min on CPU.

8. Discussion and Limitations

8.1. Key Takeaways

The benchmark yields several implications that we believe are generalisable beyond this specific setting:

1. **The "alpha cliff" is real**: even modest costs (∼5–15 bps) erase the small predictive edge of ML models in a long-short ETF setting. Papers that report only gross metrics significantly overstate practical value.
2. **Passive benchmarks must be reported**: without SPY buy-and-hold and equal-weight baselines, a reader cannot judge whether ML adds value beyond the equity risk premium.
3. **Bootstrap CIs include zero for all models**: this underscores the need for statistical testing, not just point estimates.
4. **Regime decomposition reveals hidden fragility**: models that look good on average may be entirely driven by one extreme period (e.g., COVID volatility).
5. **Strategy hyperparameters dominate model choice**: rebalance frequency and portfolio concentration shift Sharpe by >0.5, often more than the difference between model families.

6. **Multiple testing matters**: after BH-FDR correction, no ML-vs-ML pair is distinguishable, reinforcing the need for correction when comparing many models.

7. **Model ensembling helps stability but not magnitude**: rank-average ensembles achieve middle-of-pack performance, suggesting that combining weak signals does not create strong ones in this regime.

8.2. Implications for the Research Community

Our results do not imply that ML is useless for trading. Rather, they demonstrate that **evaluation methodology matters as much as model architecture**, and that the gap between "looks good in a backtest" and "is statistically and economically significant" is larger than commonly acknowledged.

We recommend that future ML trading papers:

- Report net performance under at least two cost scenarios
- Include passive benchmarks (buy-and-hold, equal-weight)
- Provide bootstrap CIs or equivalent statistical tests on key metrics
- Apply multiple-testing correction when comparing >2 models
- Report regime-conditional performance
- Release reproducible code

8.3. Limitations

1. **Daily frequency only**: the benchmark does not cover intraday or tick-level strategies, where different evaluation challenges arise.

2. **ETFs only**: individual stocks introduce survivorship bias and liquidity heterogeneity that our ETF universe avoids; results may differ.

3. **Technical features only**: we do not include fundamental, alternative, or text-based features, which may provide stronger signals in practice.

4. **Simplified cost model**: our fee-plus-slippage model does not account for market impact (price displacement from large orders), which is relevant for institutional-scale strategies.

5. **Single data period**: while we test across regimes within 2020–2024, the out-of-sample window is one contiguous period. A multi-fold walk-forward design would be stronger but computationally more expensive.

6. **Linear cost model**: costs are proportional to turnover; in practice, costs are non-linear in order size and depend on market conditions.

8.4. Future Work

Natural extensions include: (i) expanding to individual stocks with survivorship-free data (e.g., CRSP); (ii) adding fundamental and alternative-data features; (iii) incorporating market impact models; (iv) multi-fold walk-forward evaluation; (v) including reinforcement learning agents; and (vi) adding intraday/LOB evaluation protocols.

9. Conclusion

We have presented ML Trading Bench, a unified evaluation protocol and toolkit that applies rigorous, cost-aware, regime-sensitive evaluation to cross-sectional ML trading strategies on a reproducible 50-ETF universe. Our systematic evidence demonstrates that (i) transaction costs eliminate all apparent alpha in a long-short setting, (ii) no model achieves statistically significant gross performance, (iii) strategy hyperparameters dominate model choice, and (iv) headline results are fragile to regime decomposition.

The full pipeline runs in under 6 minutes with a single command, producing 9 tables and 9 figures. We hope this benchmark serves as a useful starting point for more rigorous evaluation of ML methods in quantitative finance.

Data Availability Statement

All data used in this study are freely available from public sources (Stooq, yfinance). No proprietary or restricted-access datasets are used. The complete pipeline to download, process, and evaluate the data is available at `https://github.com/georgekingsman/ml-trading-benchmark`.

CRediT authorship contribution statement

Zhang Yuchen: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization.

Declaration of competing interest

The author declares that there is no conflict of interest.

Acknowledgments

The author thanks colleagues at the University of Hong Kong for helpful discussions on quantitative trading systems and ML evaluation methodology. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work the author used AI-assisted tools to help with code development and literature review. After using these tools, the author reviewed and edited all content and takes full responsibility for the content of the published article.

References

[1] S. Sun, R. Wang, B. An, Reinforcement learning for quantitative trading, ACM Transactions on Intelligent Systems and Technology (2023). `doi:10.1145/3582560`.
URL `https://doi.org/10.1145/3582560`

[2] J. Guo, S. Wang, L. M. Ni, H.-Y. Shum, Quant 4.0: engineering quantitative investment with automated, explainable, and knowledge-driven artificial intelligence, Frontiers of Information Technology amp; Electronic Engineering (2024). `doi:10.1631/fitee.2300720`.
URL `https://doi.org/10.1631/fitee.2300720`

[3] S. Yang, Deep reinforcement learning for portfolio management, Knowledge-Based Systems (2023). `doi:10.1016/j.knosys.2023.110905`.
URL `https://doi.org/10.1016/j.knosys.2023.110905`

[4] P. Pomorski, D. Gorse, Improving portfolio performance using a novel method for predicting financial regimes, in: Artificial Neural Networks and Machine Learning (ICANN 2024), Lecture Notes in Computer Science, Springer, 2024. doi:10.1007/978-3-031-53966-4_8.
URL https://doi.org/10.1007/978-3-031-53966-4_8

[5] J. Haworth, R. Sheridan, Online learning techniques for prediction of temporal tabular datasets with regime changes, Journal of Machine Learning Research 25 (2024) 1–35.
URL https://jmlr.org/papers/v25/23-0917.html

[6] E. Lezmi, J. Roche, T. Roncalli, J. Xu, Improving the robustness of trading strategy backtesting with boltzmann machines and generative adversarial networks, SSRN Electronic Journal (2020). doi:10.2139/ssrn.3645473.
URL https://doi.org/10.2139/ssrn.3645473

[7] F. X. Diebold, R. S. Mariano, Comparing predictive accuracy, Journal of Business & Economic Statistics 13 (3) (1995) 253–263. doi:10.1080/07350015.1995.10524599.

[8] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: A practical and powerful approach to multiple testing, Journal of the Royal Statistical Society: Series B (Methodological) 57 (1) (1995) 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x.

[9] Deep learning for financial applications a survey.

[10] Financial time series forecasting with deep learning a systematic literature review 2005-2019.

[11] R. Li, J. Hu, G. Li, Deep stock trading: A hierarchical reinforcement learning framework for portfolio optimization and order execution, Information Sciences 633 (2023) 61–79. doi:10.1016/j.ins.2023.03.067.
URL https://doi.org/10.1016/j.ins.2023.03.067

[12] Asset pricing and deep learning.

[13] Qlib an ai-oriented quantitative investment platform.

[14] P. Ghosh, A. Neufeld, J. K. Sahoo, Forecasting directional movements of stock prices for intraday trading using lstm and random forests, Finance Research Letters (2022). doi:10.1016/j.frl.2021.102280.
URL https://doi.org/10.1016/j.frl.2021.102280

[15] C. R. Harvey, Y. Liu, H. Zhu, ... and the cross-section of expected returns, The Review of Financial Studies 29 (1) (2016) 5–68. doi:10.1093/rfs/hhv059.