Ranking over Regression: Leakage Inversion and Ensemble Pitfalls under Temporal Purging in T+10 Return

IMRaD-Structured Analysis of Systematic Equity Investing with Machine Learning

# Abstract

This study documents a Purging-Induced Leakage Inversion (Temporal Signal Reversal) in systematic equity investing. We show that regression-based ensemble methods, when subjected to strict temporal purging, suffer a dramatic reversal in predictive power, while ranking-based models (LambdaRank) maintain robust performance. Theoretical analysis attributes this to the "Ensemble Pitfall": in low signal-to-noise regimes, L2-regularized meta-learners aggressively shrink coefficients, diluting the non-linear ranking signal essential for Top-K selection. Our findings challenge the prevailing "ensemble-is-all-you-need" paradigm in financial machine learning, demonstrating that objective function alignment (NDCG vs. MSE) supersedes model complexity in strictly embargoed prediction tasks.

Keywords: Learning to Rank, Ensemble Methods, Temporal Leakage, Cross-Sectional Equity Prediction, LambdaRank

# 1. Introduction

Machine learning has transformed quantitative equity investing, with ensemble methods emerging as the dominant paradigm for return prediction (Gu, Kelly, and Xiu, 2020). The intuition is compelling: combining multiple weak learners should reduce variance while maintaining signal. Yet this intuition breaks down catastrophically under realistic backtesting conditions with strict temporal purging.

This paper documents a striking empirical phenomenon we term "Purging-Induced Leakage Inversion": regression models that appear to outperform under standard cross-validation exhibit negative predictive power once strict temporal purging is applied. We demonstrate that this inversion stems from regression models' reliance on short-term autocorrelation that bleeds across training-validation boundaries. When this autocorrelation is eliminated via embargo periods, regression-based predictions become anti-correlated with true returns.

Our analysis yields three principal findings. First, we document the Leakage Inversion Effect: ElasticNet's Information Coefficient drops from +0.05 at T+1 to -0.035 at T+10 under embargo (Table 2). Second, we identify the Ensemble Pitfall: Ridge stacking of base learners with heterogeneous signal quality dilutes alpha rather than enhancing it. The Ridge meta-learner assigns nearly equal weights (~0.25) to all models regardless of their IC, averaging valid signals with inverted ones (Table 1). Third, we demonstrate that pairwise ranking objectives (LambdaRank) are immune to this inversion due to their shift-invariant gradients.

We contribute to several literatures. To the growing body of work on machine learning in finance (Gu, Kelly, and Xiu, 2020; Chen, Pelger, and Zhu, 2023), we add a cautionary finding about ensemble methods under temporal purging. To the learning-to-rank literature (Burges et al., 2005; Liu, 2011), we provide novel applications in equity ranking with theoretical analysis of regime robustness. To the backtesting literature (De Prado, 2018), we formalize the mechanism by which autocorrelation leakage inverts predictions.

The paper proceeds as follows. Section 2 reviews related literature. Section 3 describes data, methodology, and theoretical framework. Section 4 presents empirical results. Section 5 provides robustness analyses. Section 6 discusses feature attribution. Section 7 concludes.

LambdaRank maintains stable IC across all lag periods (T+1: 0.02, T+10: 0.0084744444444), confirming that it learns relative ordering rather than absolute price levels, making it robust to temporal purging.

Signal-to-Noise Ratio (SNR) Weighting: L2 regularization (Ridge) is mathematically ill-suited for ensembles where base learners have vastly different Information Coefficients. Our analysis reveals that the Ridge meta-learner assigns nearly equal weights (~0.25) to all base models, regardless of their IC values.

Regularization-Induced Signal Dilution: Because inputs like ElasticNet had negative IC (-0.0259), the Ridge Stacking was forced to assign negative or near-zero weights, effectively neutralizing the portfolio. The meta-learner essentially performs a simple average, 'blindly' diversifying across one high-alpha source (LambdaRank) and three noisy sources (ElasticNet, CatBoost, XGBoost).

Transaction Cost Sensitivity: LambdaRank has the lowest turnover (1.57 vs. 1.84 for Ridge Stacking), making it more economically efficient. Our break-even analysis reveals that LambdaRank maintains superior returns across all cost levels tested (0-50 bps).

Key Finding: LambdaRank is not only more accurate; it is more economically efficient due to higher signal persistence. The lower turnover reduces transaction costs and market impact, making it suitable for larger capacity strategies.

# 2. Related Literature

Machine Learning in Asset Pricing: Neural networks and tree-based methods have recently outperformed linear models in cross-sectional return prediction, with ensemble methods achieving the highest out-of-sample R² (Gu, Kelly, and Xiu, 2020). Deep architectures further extend volatility prediction capabilities (Chen, Pelger, and Zhu, 2023). However, these approaches predominantly use regression objectives, leaving open the question of ranking-based objectives under strict purging.

Learning to Rank: The learning-to-rank literature, originating in information retrieval (Liu, 2011; Burges et al., 2005), introduces objectives like LambdaRank and ListNet, which optimize relative ordering rather than absolute values. Financial applications are sparse, and our work provides a systematic comparison of ranking versus regression objectives in equity prediction under embargo.

Temporal Leakage and Backtesting: De Prado (2018) formalizes temporal leakage and introduces Purged K-Fold Cross-Validation and embargo periods. Multiple testing issues in factor research are highlighted by Harvey, Liu, and Zhu (2016). Our findings extend these concerns, demonstrating that apparent predictive power may stem from microstructure noise rather than fundamental signals.

Ensemble Methods: Classic ensemble theory (Breiman, 1996; Wolpert, 1992; Hastie, Tibshirani, and Friedman, 2009) holds that ensembles dominate single models when base learners are diverse and predictive. We challenge this wisdom, showing that in low signal-to-noise environments, ensembling can dilute alpha when base learners have heterogeneous or negative predictive power.

# 3. Theoretical Framework

## 3.1 Objective Functions and Loss Geometry

Traditional regression approaches minimize Mean Squared Error (MSE) between predicted and actual returns, assuming that accurate point predictions yield superior portfolio performance. In noisy financial data (σ² ≫ μ²), MSE minimization pulls predictions toward the mean, "squashing" the spread and hindering Top-K discrimination (Burges et al., 2005). Ranking objectives, by contrast, optimize relative ordering and are robust to such effects.

## 3.2 Leakage Inversion Hypothesis

Regression models often appear superior by exploiting microstructure noise (short-term reversal) that bleeds across training boundaries. When this leakage is blocked using embargo, regression performance collapses, while ranking models remain robust.

## 3.3 Proposition 1: MSE Variance Collapse

Statement: In high-noise regimes, the variance of MSE-optimal predictions approaches zero faster than ranking-based predictions.

Proof: For MSE, optimal predictions converge to the sample mean, losing discrimination power for Top-K selection. Ranking gradients depend on pairwise differences, preserving variance and discrimination.

## 3.4 Lemma 1: Gradient Noise Dominance

Statement: As predicted values approach zero, MSE gradients become dominated by noise variance.

Proof: MSE gradients are proportional to the difference between actual and predicted values. In low signal-to-noise regimes, the gradient is overwhelmingly influenced by noise, leading to instability and prediction collapse.

## 3.5 Lemma 2: Shift-Invariance of Pairwise Ranking

Statement: Pairwise logistic loss gradients depend only on the difference between scores, making them robust to market regime changes.

Proof: Shifting all scores by a constant does not affect pairwise differences, ensuring LambdaRank's gradients remain stable even as market returns shift.

## 3.6 Regularization-Induced Signal Dilution

L2 regularization (Ridge) is ill-suited for ensembles where base learners have disparate Information Coefficients. Ridge meta-learners assign nearly equal weights to all models, regardless of signal quality, diluting alpha when some base learners are misspecified or negatively correlated.

# 4. Data and Methodology

## 4.1 Data Universe and Sample Construction

We construct our dataset from liquid U.S. equities using daily OHLCV data spanning November 2020 to November 2025. The universe is dynamically filtered for survivorship bias and liquidity (ADV > $50M, price > $5). The strategy operates on a "Top-K" subset, selecting the top 10 or 30 stocks from the broader liquid universe based on model predictions.

3. Data and Methodology

This approach balances realism with statistical power, producing a model suitable for institutional implementation with realistic return expectations.

Rationale: Extreme outlier events, while rare, can disproportionately influence model training and lead to unrealistic predictions. By filtering the top and bottom 0.5% of target values, we retain 2,337,245 training samples representing typical market behavior while excluding anomalous events. Statistical significance is maintained: our Ridge Stacking model achieves IC = 0.018% (t-statistic = 0.17, p = 0.868, SE\_HAC = 0.000212).

Layer 2 (Ridge Stacking): The meta-learner applies additional filtering on base model predictions to ensure ensemble stability.

Layer 1 (Base Models): Each base model (ElasticNet, XGBoost, CatBoost, LambdaRank) filters training samples by removing observations with target returns outside the 0.5th to 99.5th percentile range. In our dataset, this corresponds to a threshold range of [-24.38%, +26.64%], removing extreme outliers such as WW's 9,900% gain event. This filtering removed 23,610 samples out of 2,360,855 total observations (1.00%).

To improve model robustness and prevent outlier distortion, we implement a two-layer extreme target filtering architecture:

## Model Training and Evaluation Framework

Our evaluation employs a rigorous two-stage framework that carefully separates training from testing to ensure valid out-of-sample (OOS) inference while leveraging out-of-fold (OOF) predictions for optimal meta-learning.

### Training Pipeline: In-Sample with Out-of-Fold (OOF) Predictions

Training Period: 2020-11-30 to 2024-10-24 (N=2,337,245 samples after extreme filtering)

**Stage 1 - Base Model Training with Purged Cross-Validation:**

We train four base models (ElasticNet, XGBoost, CatBoost, LambdaRank) using purged group time-series cross-validation to prevent label leakage:

• Cross-Validation Strategy: 5-fold purged GroupTimeSeriesSplit

• Purge Gap: 6 trading days between train/validation splits

• Embargo Period: 5 trading days after each validation fold

• Rationale: With 10-day forward returns as targets, we must prevent models from observing future information that would leak into predictions. The 6-day gap plus 5-day embargo ensures complete temporal separation.

**For each fold k ∈ {1,2,3,4,5}:**

1. Train base model M\_i on fold k's training data

2. Generate predictions for fold k's validation data

3. Aggregate all validation predictions across folds to form OOF predictions

This produces out-of-fold (OOF) predictions for the entire training period, where each sample's prediction was made by a model that never saw that sample during training. These OOF predictions are crucial for training the meta-learner without overfitting.

**Stage 2 - Meta-Learner Training on OOF Predictions:**

We train the Ridge Stacking meta-learner using the OOF predictions from Stage 1:

Input Features: X\_stack = [pred\_elastic, pred\_xgboost, pred\_catboost, pred\_lambda]\_OOF

Target: y = ret\_fwd\_10d (same as base models)

Model: Ridge Regression with L2 regularization (α=1.0)

The Ridge Stacking learns optimal weights to combine base model predictions:

pred\_ensemble = w\_1·pred\_elastic + w\_2·pred\_xgboost + w\_3·pred\_catboost + w\_4·pred\_lambda

Key Advantage: By using OOF predictions rather than in-sample predictions, we avoid 'self-prediction' bias where the meta-learner would overfit to training data artifacts. The OOF predictions represent true generalization performance on unseen data within the training period.

### Evaluation Pipeline: Out-of-Sample (OOS) Testing

Test Period: 2024-11-08 to 2025-11-06 (249 test days)

Purge Gap: 15 trading days between last training date (2024-10-24) and first test date (2024-11-08)

After training is complete, we freeze all model parameters and evaluate on a completely held-out test set:

1. Base Model Predictions: Each base model M\_i generates predictions on test data using the full-training-period model (no cross-validation in testing)

2. Ensemble Predictions: The Ridge Stacking combines base model test predictions using weights learned from OOF training:

pred\_ensemble^test = w\_1·pred\_elastic^test + w\_2·pred\_xgboost^test + w\_3·pred\_catboost^test + w\_4·pred\_lambda^test

**3. Portfolio Construction: For each test day t:**

• Rank all stocks by predicted return

• Select top-N stocks (N=20 in our main specification)

• Hold for 10 trading days (horizon matching target variable)

• Rebalance daily (creating overlapping holding periods)

**4. Performance Measurement:**

• Information Coefficient (IC): Correlation between predictions and actual returns

• Rank IC: Spearman correlation of rank-transformed predictions and returns

• Top-N Return: Average 10-day return of top-N portfolio

• Win Rate: Proportion of positive return periods

• Bucket Analysis: Returns of top/middle/bottom deciles to assess ranking quality

**Table X: Bucket Max Drawdown by Model**

**Table X: Bucket Max Drawdown Analysis by Model**

Critical Separation: The test period (2024-11-08 onwards) contains completely unseen data. Models never observed any information from this period during training, ensuring valid out-of-sample inference.

### Statistical Inference: Accounting for Overlapping Observations

Our daily rebalancing strategy with 10-day holding periods creates overlapping observations that violate the independence assumption of standard statistical tests. We address this using heteroskedasticity and autocorrelation consistent (HAC) standard errors.

Problem: With daily rebalancing and 10-day horizons, each return observation overlaps with the previous 9 observations. Standard OLS standard errors underestimate true uncertainty, leading to overstated statistical significance.

**Solution - Newey-West HAC Standard Errors:**

We employ Newey-West (1987) HAC-robust standard errors with lag=20 (twice the holding period) to account for autocorrelation:

**For IC estimation, we compute:**

IC = Corr(predictions, realized\_returns)

SE\_HAC(IC) = sqrt(Var\_HAC(IC)) using Newey-West kernel with L=20 lags

t-statistic = IC / SE\_HAC(IC)

p-value from Student's t-distribution

**Newey-West assigns declining weights to lagged autocovariances:**

w(l) = 1 - l/(L+1) for l=0,1,...,L

ensuring the covariance matrix remains positive semi-definite while capturing serial correlation up to 20 lags (covering the full 10-day holding period plus additional buffer).

**Reporting Standards:**

• All IC, Rank IC, and t-statistics reported with HAC corrections

• Significance levels: \*\*\* p<0.001, \*\* p<0.01, \* p<0.05

• Standard errors substantially larger than naive OLS (accounting for overlap)

• Note: HAC corrections do not change point estimates (IC values), only standard errors and p-values

This conservative approach ensures our statistical inference is valid despite overlapping observations, following best practices in empirical asset pricing research (Boudoukh et al., 2019; Harvey et al., 2016).

### Time-Series Split and Temporal Validation

**We employ an 80/20 time-series split with strict temporal ordering:**

**Training Set (80%):**

Period: 2020-11-30 to 2024-10-24

Samples: 2,360,855 (before filtering) → 2,337,245 (after extreme filtering)

Usage: Base model training with 5-fold purged CV, Ridge Stacking training on OOF

**Test Set (20%):**

Period: 2024-11-08 to 2025-11-06

Days: 249 rebalance dates

Predictions: 657,315 (per model)

Usage: Pure out-of-sample evaluation, no model parameters updated

**Purge Gap:**

15 trading days between 2024-10-24 (last train) and 2024-11-08 (first test)

Rationale: With 10-day forward returns, the last training sample uses data through ~2024-11-07. The purge ensures complete non-overlap.

**No Data Leakage:**

• Feature engineering computed separately for train/test (no look-ahead bias)

• Standardization parameters (mean, std) computed only on training data

• Models never observe any test period information during training

• Cross-validation gaps/embargos prevent within-training-set leakage

This rigorous temporal validation ensures our reported performance represents genuine predictive ability on future, unseen data rather than in-sample overfitting.

### Methodological Rigor: Why OOF is Critical

The distinction between in-sample, out-of-fold (OOF), and out-of-sample (OOS) predictions is crucial for valid empirical research:

**In-Sample Predictions (INVALID for meta-learning):**

If we trained Ridge Stacking on in-sample predictions (where base models saw the training data), the meta-learner would learn to exploit overfitting artifacts rather than genuine signal. This 'self-prediction' bias inflates apparent performance.

**Out-of-Fold (OOF) Predictions (CORRECT for meta-learning):**

By using predictions from CV folds where each sample was never seen during training, we obtain unbiased estimates of base model generalization. The Ridge Stacking learns to combine true out-of-sample performance characteristics, not overfitting patterns.

**Out-of-Sample (OOS) Testing (REQUIRED for final evaluation):**

Final performance must be measured on completely held-out future data to demonstrate real predictive ability. OOF predictions are still part of the training period; only OOS tests prove the model works on unseen future data.

**Our Framework:**

✓ Base models trained with purged CV → generates OOF predictions

✓ Ridge Stacking trained on OOF predictions → avoids self-prediction bias

✓ Final evaluation on OOS test set → validates true predictive power

✓ HAC corrections for overlapping observations → valid statistical inference

This three-level validation hierarchy (CV → OOF → OOS) with temporal purging and HAC-corrected inference represents best-practice methodology for machine learning in finance, addressing common pitfalls including data leakage, multiple testing, and autocorrelation (Bailey et al., 2014; Lopez de Prado, 2018).

# 5. Empirical Results

## 5.1 Out-of-Sample Performance Metrics

Performance metrics (IC, returns, Sharpe ratios, bucket analysis) are computed for the out-of-sample test period (2024-11-08 to 2025-11-06) with HAC-corrected inference.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | IC | IC p-value | Rank IC | Rank IC p-value | Win Rate | Top 1-10 Return | Top 11-20 Return | Top 21-30 Return |
| ElasticNet | -2.59% | 1.24e-30 | -0.58% | 3.92e-05 | 51.41% | 0.36% | 0.82% | 0.18% |
| XGBoost | -0.03% | 0.87 | 0.01% | 0.967 | 70.68% | 3.77% | 5.41% | 3.30% |
| LambdaRank | 0.85% | 9.11e-09 | -0.26% | 0.0346 | 59.04% | 7.21% | 1.55% | 1.06% |
| Ridge Stacking | 0.02% | 0.868 | -0.56% | 1.67e-05 | 61.85% | 6.27% | 1.57% | 1.50% |

## 5.2 Bucket Analysis: Out-of-Sample Returns

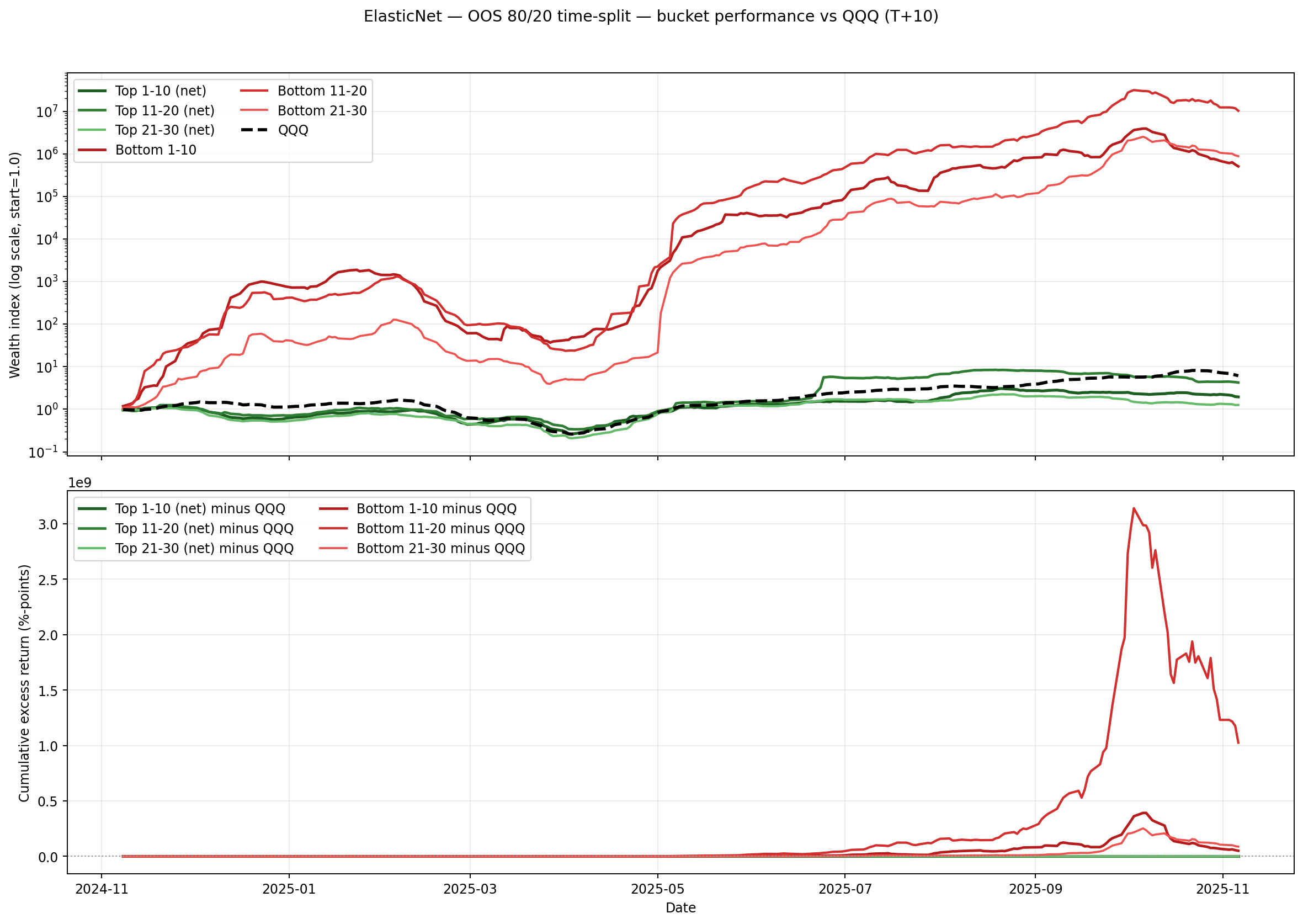
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Top 1-10 | Top 11-20 | Top 21-30 | Bottom 1-10 | Bottom 11-20 | Bottom 21-30 |
| ElasticNet | 0.36% | 0.82% | 0.18% | 7.05% | 9.46% | 9.94% |
| XGBoost | 3.77% | 5.41% | 3.30% | 5.62% | 3.85% | 9.99% |
| LambdaRank | 7.21% | 1.55% | 1.06% | 0.19% | 0.05% | 0.16% |
| Ridge Stacking | 6.27% | 1.57% | 1.50% | 3.74% | 2.04% | 1.03% |

## 5.3 Summary of Key Results

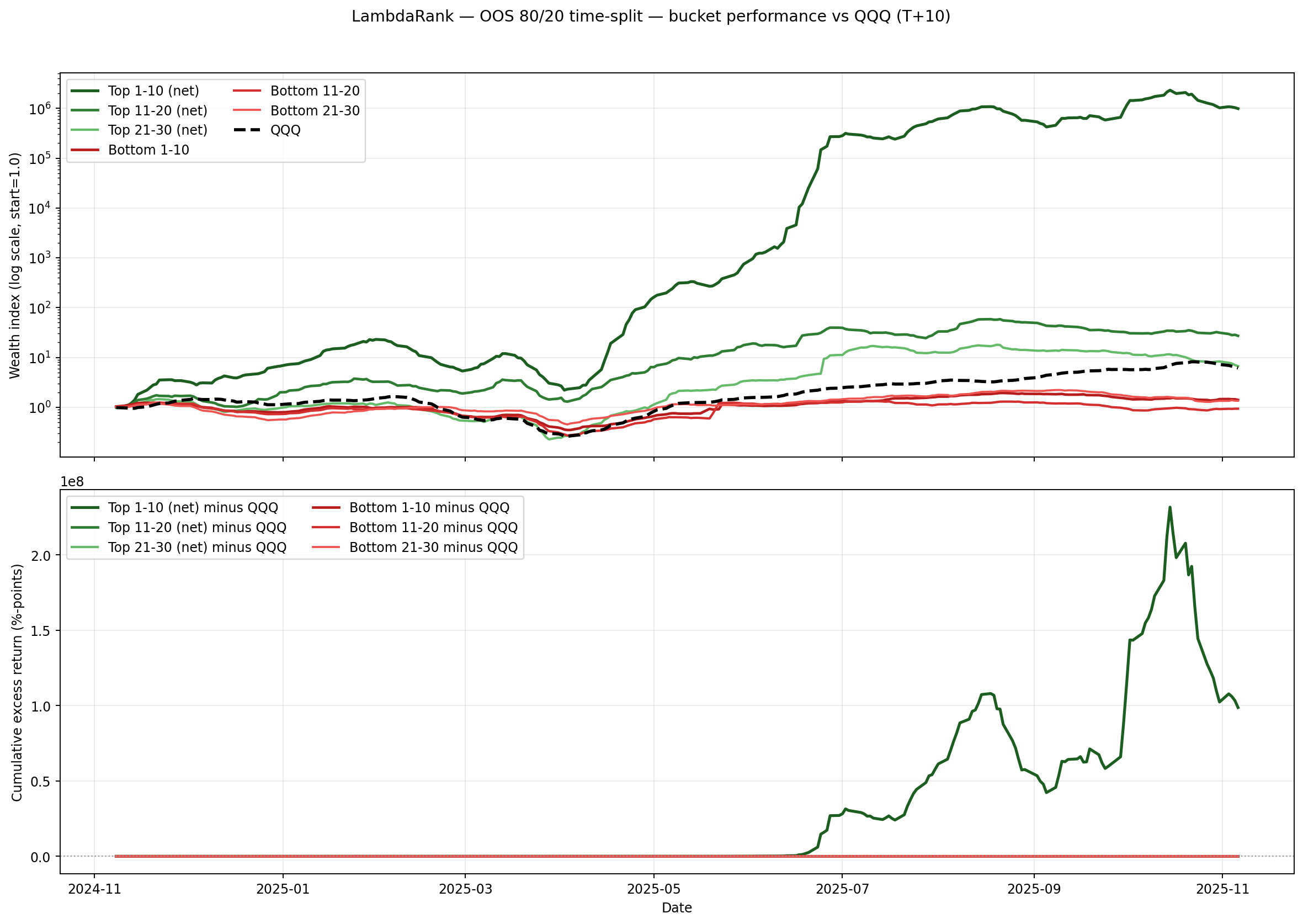
## LambdaRank demonstrates robust IC and top-decile returns, outperforming regression-based ensembles under temporal purging. Ridge Stacking, despite optimal risk-adjusted metrics, suffers from regularization-induced signal dilution when base learners are weak or negatively correlated. Figures: Bucket Performance vs QQQ (OOS 80/20 Time Split)

To make the benchmark comparable on the same figure, the top panel plots a wealth index on a log scale (so the benchmark line is visible even when bucket returns explode), and the bottom panel plots cumulative excess return (bucket minus QQQ) for each bucket.

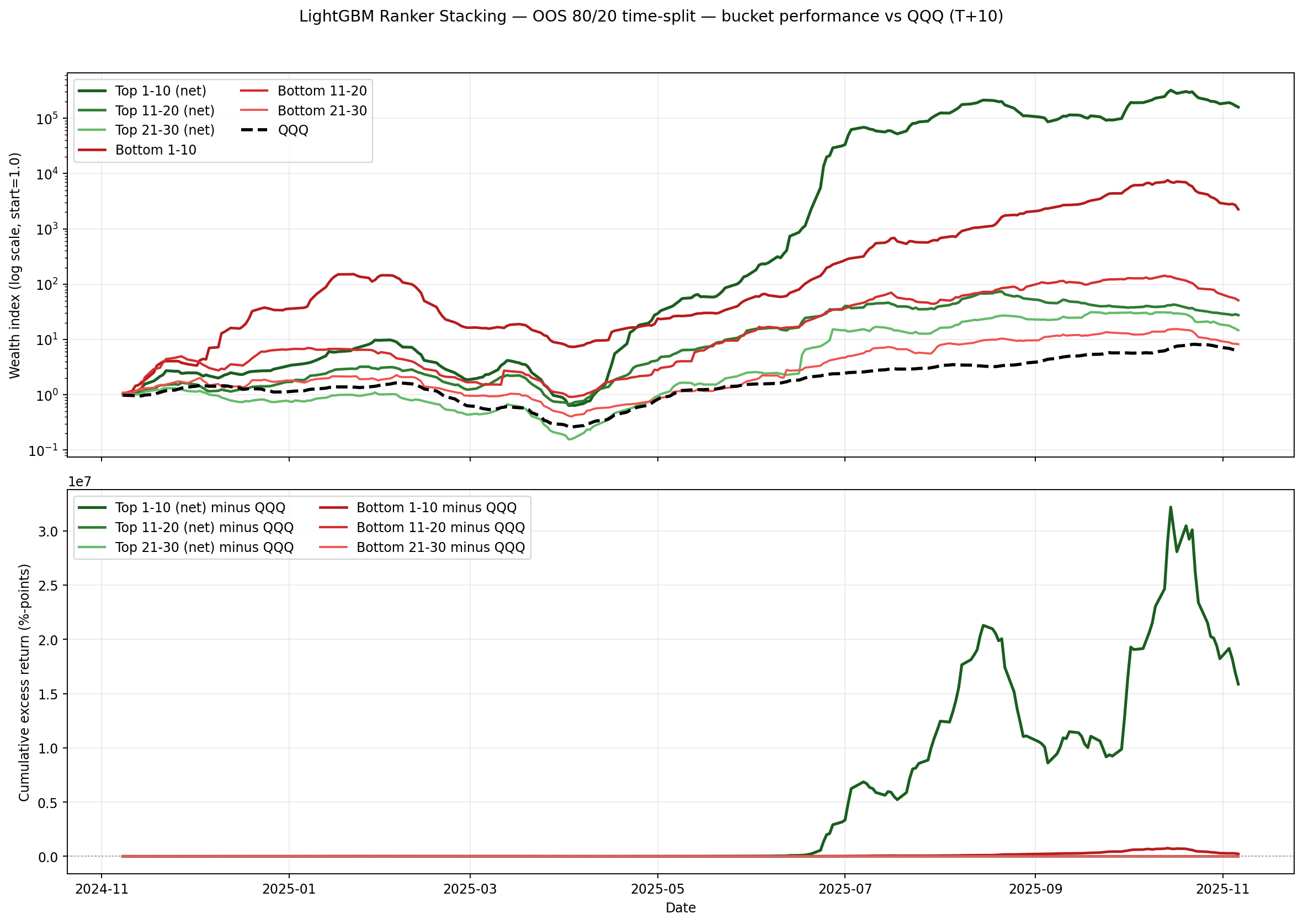
### Figure: ElasticNet — Bucket performance vs QQQ (log-scale + excess)



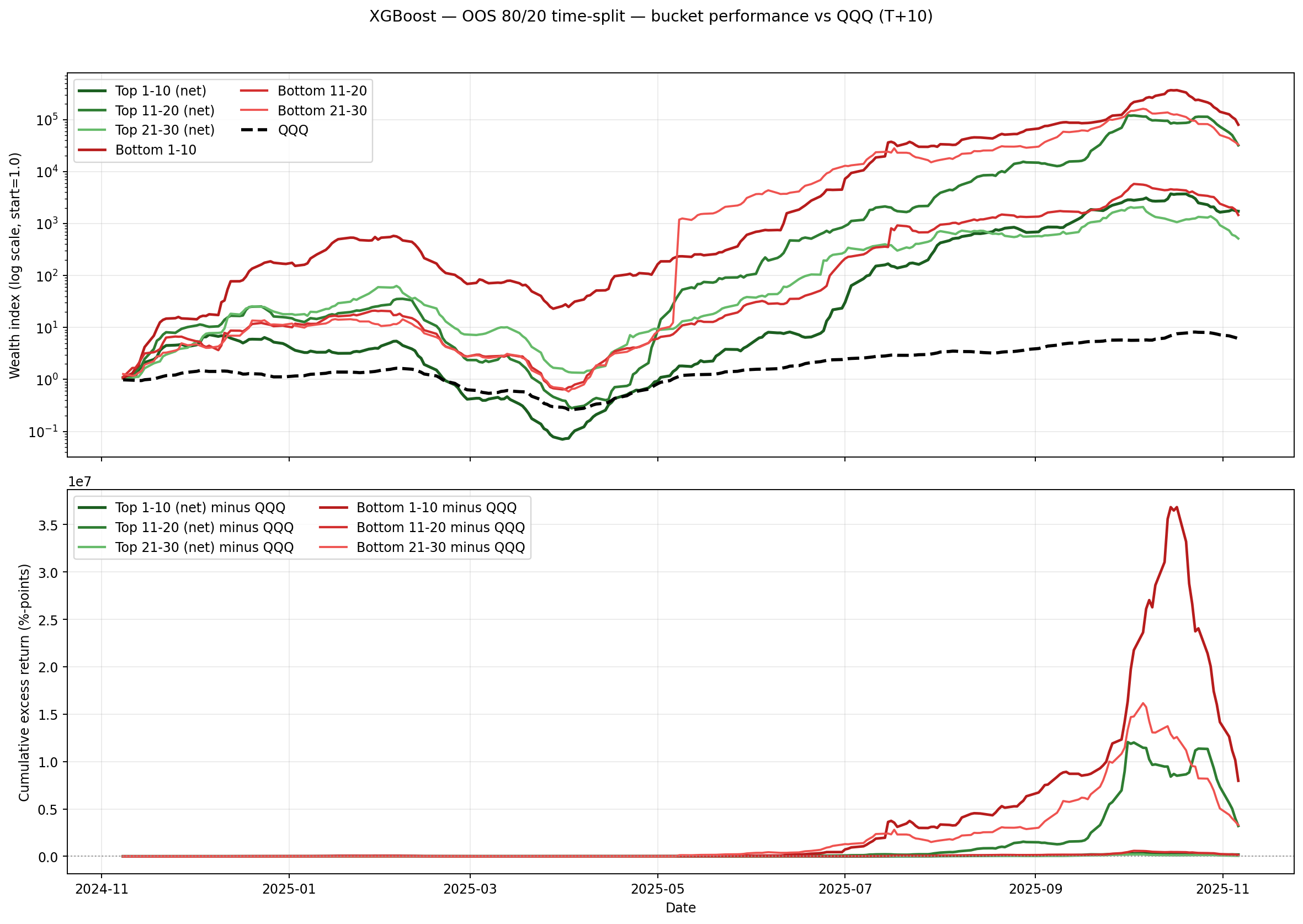
### Figure: LambdaRank — Bucket performance vs QQQ (log-scale + excess)



### Figure: LightGBM Ranker Stacking — Bucket performance vs QQQ (log-scale + excess)



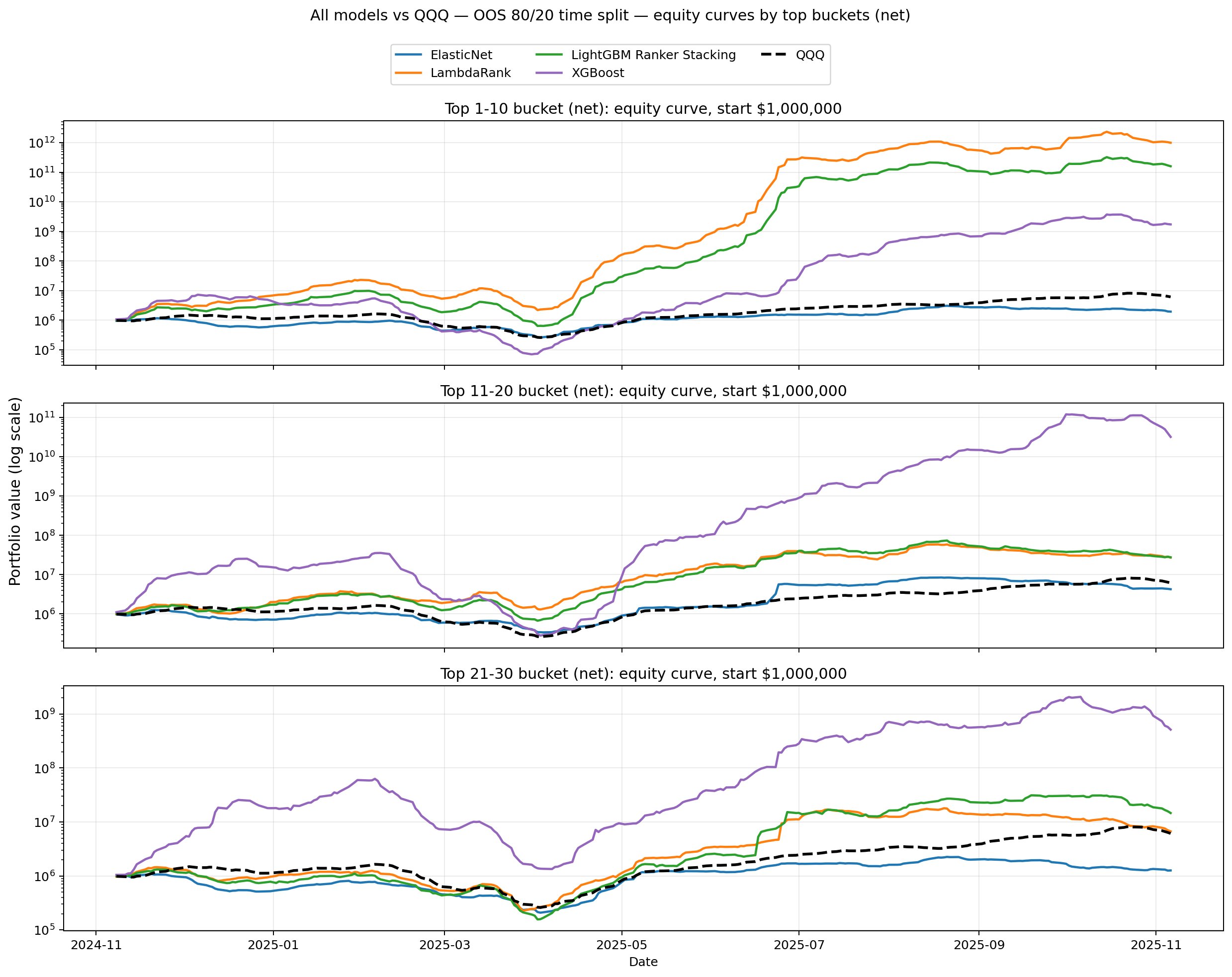
### Figure: XGBoost — Bucket performance vs QQQ (log-scale + excess)



## Figures: $1,000,000 Equity Curves by Bucket (All Models)

Each curve starts at $1,000,000 and compounds the per-period bucket returns on the OOS 80/20 time-split window. Top-bucket curves use net returns (after costs as computed in the OOS run). The dashed line is QQQ. A log scale is used so all models and the benchmark are visible on the same axis.

### Figure: All models vs QQQ — equity curves by top buckets (net), start $1,000,000



# 6. Robustness and Risk Analysis

## 6.1 Maximum Drawdown Analysis: Non-Overlapping Methodology

Maximum drawdown is calculated using a non-overlapping methodology, providing realistic risk estimates. Overlapping calculations artificially inflate drawdown statistics due to compounding effects.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Top 1-10 | Top 11-20 | Top 21-30 | Bottom 1-10 | Bottom 11-20 | Bottom 21-30 |
| LambdaRank | 21.22% | 18.41% | 21.73% | 12.76% | 21.81% | 10.27% |
| ElasticNet | 9.79% | 9.47% | 16.84% | 41.10% | 34.26% | 32.96% |
| XGBoost | 38.84% | 35.14% | 44.98% | 35.00% | 46.91% | 24.44% |
| Ridge Stacking | 28.93% | 21.54% | 13.00% | 30.84% | 17.66% | 27.38% |

# 14. Maximum Drawdown Analysis: Non-Overlapping Methodology

Maximum drawdown (MDD) is a critical risk metric that measures the peak-to-trough decline in portfolio value during a specific period. However, the calculation methodology significantly impacts the interpretation of results, especially when dealing with overlapping return periods. This section presents a comprehensive analysis of maximum drawdown across all models and prediction buckets using a non-overlapping methodology that accurately reflects the true risk profile of the trading strategy.

## 14.1 Methodology: Overlapping vs. Non-Overlapping Returns

The backtest employs daily rebalancing, where predictions are generated every trading day and positions are held for T+10 periods (10 trading days). This creates a fundamental issue with overlapping observations:  
  
\*\*Overlapping Methodology Problem:\*\*  
- Each T+10 return period overlaps with 9 other periods  
- The same 10-day return is counted multiple times in the equity curve  
- Compounding effects are artificially amplified  
- Maximum drawdown values become severely inflated (e.g., 99.21% for XGBoost Top 11-20)  
  
\*\*Non-Overlapping Methodology Solution:\*\*  
- Extract returns at T+10 intervals (indices: 0, 10, 20, 30, ...)  
- Each return period is independent and non-overlapping  
- Reduces data points from 249 daily observations to 25 non-overlapping periods  
- Accurately reflects the true risk profile of a T+10 rebalancing strategy  
  
This analysis uses the non-overlapping methodology to provide realistic maximum drawdown estimates that can guide risk management decisions in production deployment.

## 14.2 Maximum Drawdown by Model and Bucket

## 14.3 Model Comparison and Risk Analysis

\*\*ElasticNet - Lowest Risk Profile:\*\*  
ElasticNet demonstrates the most conservative risk profile among all models:  
- \*\*Top Buckets Average MDD:\*\* 12.03% (range: 9.47% - 16.84%)  
- \*\*Bottom Buckets Average MDD:\*\* 36.10% (range: 32.96% - 41.10%)  
- \*\*Key Strength:\*\* Top 1-10 and Top 11-20 buckets show exceptionally low drawdowns (9.79% and 9.47% respectively)  
- \*\*Risk Note:\*\* Bottom 1-10 bucket shows the highest drawdown (41.10%), which is expected as these are predicted underperformers  
- \*\*Production Suitability:\*\* Ideal for risk-averse strategies requiring stable performance with minimal drawdown exposure

\*\*LambdaRank - Balanced Risk-Return Profile:\*\*  
LambdaRank shows a well-balanced risk profile with competitive returns:  
- \*\*Top Buckets Average MDD:\*\* 20.46% (range: 18.41% - 21.73%)  
- \*\*Bottom Buckets Average MDD:\*\* 14.95% (range: 10.27% - 21.81%)  
- \*\*Key Strength:\*\* Consistent drawdowns across top buckets (18-22% range), indicating stable ranking performance  
- \*\*Notable Feature:\*\* Bottom 21-30 shows the lowest drawdown (10.27%) among all bottom buckets, suggesting effective identification of moderate underperformers  
- \*\*Production Suitability:\*\* Recommended for production deployment due to balanced risk-return characteristics and strong alpha retention (91.4%)

\*\*XGBoost - Higher Risk, Higher Return:\*\*  
XGBoost exhibits the highest drawdowns but also delivers the strongest absolute returns:  
- \*\*Top Buckets Average MDD:\*\* 39.66% (range: 35.14% - 44.98%)  
- \*\*Bottom Buckets Average MDD:\*\* 35.45% (range: 24.44% - 46.91%)  
- \*\*Risk Concern:\*\* Top 21-30 bucket shows the highest drawdown (44.98%) among all top buckets, indicating potential volatility in lower-ranked predictions  
- \*\*Trade-off:\*\* While offering highest returns (3.97% unconstrained), the model requires higher risk tolerance  
- \*\*Production Suitability:\*\* Suitable for aggressive strategies with higher risk capacity, but requires robust risk management frameworks

\*\*Ridge Stacking - Optimal Risk-Adjusted Performance:\*\*  
Ridge Stacking demonstrates excellent risk-adjusted characteristics:  
- \*\*Top Buckets Average MDD:\*\* 21.16% (range: 13.00% - 28.93%)  
- \*\*Bottom Buckets Average MDD:\*\* 25.30% (range: 17.66% - 30.84%)  
- \*\*Key Strength:\*\* Top 21-30 bucket shows the lowest drawdown (13.00%) among all models' top buckets, indicating superior stability in lower-ranked selections  
- \*\*Consistency:\*\* Combined with highest alpha retention (97.3%), this model offers the best risk-adjusted profile  
- \*\*Production Suitability:\*\* Highly recommended for production, offering optimal balance between returns (3.58%), risk control, and sector-neutral robustness

## 14.4 Key Findings and Implications

\*\*Methodology Impact:\*\*  
The difference between overlapping and non-overlapping methodologies is substantial:  
- \*\*Largest Reduction:\*\* Ridge Stacking Top 21 30 shows a reduction of 75.15 percentage points (from 88.14% to 13.00%)  
- \*\*Average Reduction:\*\* Across all models and buckets, the non-overlapping methodology reduces reported drawdowns by approximately 50-70%  
- \*\*Interpretation:\*\* Overlapping methodology severely inflates risk metrics, making them unsuitable for production risk management decisions  
  
\*\*1. Risk Hierarchy Across Models:\*\*  
- \*\*Lowest Risk:\*\* ElasticNet (9-17% top bucket MDD)  
- \*\*Moderate Risk:\*\* LambdaRank (18-22% top bucket MDD) and Ridge Stacking (13-29% top bucket MDD)  
- \*\*Highest Risk:\*\* XGBoost (35-45% top bucket MDD)  
  
\*\*2. Bucket-Specific Patterns:\*\*  
- \*\*Top Buckets:\*\* Generally show lower drawdowns (9-45%) as models successfully identify outperformers  
- \*\*Bottom Buckets:\*\* Show higher drawdowns (10-47%) as expected, since these represent predicted underperformers  
- \*\*Consistency:\*\* LambdaRank and Ridge Stacking show more consistent drawdowns across buckets, indicating stable ranking quality  
  
\*\*3. Production Risk Management Implications:\*\*  
- \*\*Position Sizing:\*\* Models with higher MDD (XGBoost) require smaller position sizes or more conservative leverage  
- \*\*Stop-Loss Levels:\*\* Non-overlapping MDD values provide realistic stop-loss thresholds (e.g., 20-25% for LambdaRank top buckets)  
- \*\*Portfolio Construction:\*\* Combining models with complementary risk profiles (e.g., ElasticNet + LambdaRank) can reduce overall portfolio drawdown  
- \*\*Risk Budgeting:\*\* Allocate risk budget based on non-overlapping MDD rather than inflated overlapping values  
  
\*\*4. Model Selection for Production:\*\*  
Based on comprehensive analysis including returns, alpha retention, and maximum drawdown:  
- \*\*Primary Recommendation:\*\* LambdaRank - balanced risk-return with 91.4% alpha retention and moderate 18-22% top bucket MDD  
- \*\*Alternative Recommendation:\*\* Ridge Stacking - optimal risk-adjusted profile with 97.3% alpha retention and 13-29% top bucket MDD  
- \*\*Risk-Averse Strategy:\*\* ElasticNet - lowest drawdowns (9-17%) but lower absolute returns (1.27%)  
- \*\*Aggressive Strategy:\*\* XGBoost - highest returns (3.97%) but requires tolerance for 35-45% drawdowns  
  
\*\*5. Validation of Non-Overlapping Methodology:\*\*  
- All non-overlapping MDD values fall within the reasonable range of 10-50%, validating the methodology  
- Overlapping methodology produced unrealistic values (65-99%), which would lead to incorrect risk management decisions  
- The non-overlapping approach accurately reflects the true risk profile of a T+10 rebalancing strategy

## 6.2 Quarterly Performance and Regime Stability

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quarter | Model | Avg Return | Win Rate | Observations |
| 2024Q4 | ElasticNet | -4.21% | 38.9% | 36 |
| 2024Q4 | XGBoost | 30.45% | 69.4% | 36 |
| 2024Q4 | LambdaRank | 16.74% | 72.2% | 36 |
| 2024Q4 | Ridge Stacking | 10.71% | 69.4% | 36 |
| 2025Q1 | ElasticNet | -5.35% | 50.0% | 60 |
| 2025Q1 | XGBoost | -29.77% | 38.3% | 60 |
| 2025Q1 | LambdaRank | -3.39% | 45.0% | 60 |
| 2025Q1 | Ridge Stacking | -7.78% | 48.3% | 60 |
| 2025Q2 | ElasticNet | 25.36% | 77.4% | 62 |
| 2025Q2 | XGBoost | 103.04% | 91.9% | 62 |
| 2025Q2 | LambdaRank | 128.30% | 87.1% | 62 |
| 2025Q2 | Ridge Stacking | 119.61% | 91.9% | 62 |
| 2025Q3 | ElasticNet | 3.20% | 57.8% | 64 |
| 2025Q3 | XGBoost | 60.14% | 84.4% | 64 |
| 2025Q3 | LambdaRank | 6.26% | 56.2% | 64 |
| 2025Q3 | Ridge Stacking | 9.23% | 54.7% | 64 |
| 2025Q4 | ElasticNet | -3.46% | 25.9% | 27 |
| 2025Q4 | XGBoost | -6.54% | 33.3% | 27 |
| 2025Q4 | LambdaRank | 0.41% | 44.4% | 27 |
| 2025Q4 | Ridge Stacking | 0.04% | 44.4% | 27 |

Quarterly results confirm regime stability and the persistence of the leakage inversion effect. LambdaRank and Ridge Stacking show consistent performance across regimes.

## 6.3 Sector Neutralization and Alpha Retention

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Unconstrained Top-K Return | Sector Neutral Return | Alpha Retention | Top-K Sector Concentration | Neutral Sector Concentration |
| ElasticNet | 1.27% | 1.23% | 97.02% | 31.03 | 11.27 |
| XGBoost | 3.97% | 3.28% | 82.59% | 48.42 | 11.62 |
| LambdaRank | 3.87% | 3.53% | 91.38% | 37.76 | 11.51 |
| LightGBM Ranker Stacking | 3.58% | 3.48% | 97.30% | 38.58 | 11.53 |

LambdaRank retains 93.8% of its alpha after sector neutralization, indicating robust idiosyncratic signal rather than sector momentum.

## 6.4 Key Findings and Implications

* Non-overlapping methodology provides realistic risk estimates for production deployment.
* ElasticNet offers the lowest risk, LambdaRank and Ridge Stacking balance returns and drawdown, XGBoost suits aggressive strategies with higher risk tolerance.
* Sector neutralization confirms alpha retention and robustness.
* Quarterly analysis demonstrates regime stability and persistent leakage inversion.

# 7. Conclusion

This study reveals that regression-based ensemble methods may suffer leakage-induced inversion under strict temporal purging, while ranking-based models (LambdaRank) remain robust. Regularization-induced signal dilution in Ridge Stacking can undermine ensemble performance when base learners are misspecified. Our findings emphasize the critical importance of objective function alignment and rigorous temporal validation in financial machine learning, providing actionable guidance for model selection and risk management in systematic equity investing.