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

Abstract

This study documents a Purging-Induced Leakage Inversion (Temporal Signal Reversal) in systematic equity investing: while regression ensemble (Sharpe 1.39)) compared to the regression ensemble's 2.23% (Sharpe 1.39). We theoretically attribute this to the 'Ensemble Pitfall': in low signal-to-noise regimes where base learners exhibit bias, L2-regularized meta-learners (Ridge) aggressively shrink coefficients, diluting the non-linear ranking signal required for Top-K selection. These findings challenge the prevailing 'ensemble-is-all-you-need' paradigm in financial machine learning, suggesting 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

Note: 基于重叠观测 (overlapping observations), 使用 Newey-West HAC 校正 (lag=20)
The analysis is conducted on a broad universe of liquid US equities. The backtesting framework defines the universe dynamically at each rebalance interval, filtering for survivorship bias and liquidity. Specifically, the strategy operates on a "Top-K" subset, selecting the top 10 or 30 stocks from the broader liquid universe based on model predictions.12

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).

Summary of Reported Results

Out-of-Sample Results Tables and Figures

Table 1: Out-of-Sample Performance Metrics

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%

Table 2: Out-of-Sample Bucket Returns Summary

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%

Table 2.1: Out-of-Sample Detailed Bucket Statistics by Model

Out-of-Sample ElasticNet Bucket Statistics

Bucket	Mean (%)	Std (%)	Min (%)	Max (%)	N
Top 1-10	36.1750	433.2489	-1437.5620	1690.8492	249.0000
Top 11-20	81.9337	791.0839	-1081.2374	7486.8952	249.0000
Top 21-30	18.0299	419.4667	-1413.3739	2256.1456	249.0000
Bottom 1-10	704.9151	2023.7230	-3067.9334	13417.9233	249.0000
Bottom 11-20	946.3066	3763.7769	-2845.8248	51496.3833	249.0000
Bottom 21-30	993.8394	5991.2143	-2768.1415	73709.4411	249.0000

Out-of-Sample ElasticNet Sample Period Returns (First 10 and Last 10 Days)

Date	Top 1-10	Top 11-20	Top 21-30	Bottom 1-10	Bottom 11-20	Bottom 21-30
2024-11-08	-4.2287	-2.6845	-2.0032	18.0363	5.1315	8.7384

2024-11-11	2.1448	-6.5319	0.6065	16.4873	26.7781	3.9027
2024-11-12	2.1219	0.5414	0.6248	18.5134	20.3121	-1.9531
2024-11-13	1.2830	2.3943	-4.0021	9.7221	42.1675	3.9649
2024-11-14	-0.2978	1.4967	2.5694	45.1545	85.6054	6.4892
2024-11-15	4.7613	4.8416	4.8628	26.0582	85.0706	7.3017
2024-11-18	2.6971	6.8438	2.7603	10.7265	41.9672	33.6546
2024-11-19	0.2068	5.1116	0.6246	-1.8031	27.1629	13.4354
2024-11-20	3.6972	11.3106	2.5034	35.2628	3.9884	31.1654
2024-11-21	1.7854	0.4299	-2.1911	24.9143	36.7983	29.5929
...	...	...	...	...	...	...
2025-10-24	-0.5633	-5.6572	-1.0913	-13.7482	3.3199	-16.1145
2025-10-27	-2.0698	1.9045	-2.0388	-14.2905	-10.9562	-2.6576
2025-10-28	-1.0629	-0.7098	-0.8104	-10.9211	11.3254	-1.9462
2025-10-29	1.7851	0.1498	0.4085	0.2024	-15.5361	-1.6597
2025-10-30	-1.3016	-0.2477	3.7353	-4.4961	-6.2342	-3.3429
2025-10-31	2.3192	0.0305	1.1541	-5.9647	-13.1485	-7.9158
2025-11-03	-3.9186	1.0480	-2.1666	-11.4057	0.0207	-4.2348
2025-11-04	-2.2968	-1.6718	-0.3378	4.3162	-1.1878	-0.6233
2025-11-05	-6.0011	-1.1916	-4.1920	-11.3067	-3.3298	-8.7063
2025-11-06	-1.4221	-2.1723	0.6216	-9.4015	-12.7955	-4.4923

Out-of-Sample XGBoost Bucket Statistics

Bucket	Mean (%)	Std (%)	Min (%)	Max (%)	N
Top 1-10	377.3793	1255.1744	-2416.5891	5383.1029	249.0000
Top 11-20	540.9905	1627.7931	-3095.3778	9714.8147	249.0000
Top 21-30	330.0630	1310.0611	-2640.0353	8777.5348	249.0000
Bottom 1-10	562.2978	1543.6014	-2289.4372	8489.1273	249.0000
Bottom 11-20	384.5061	1486.0731	-2772.1926	12980.8520	249.0000
Bottom 21-30	999.0066	7921.7250	-2548.5912	89255.3449	249.0000

Out-of-Sample XGBoost Sample Period Returns (First 10 and Last 10 Days)

Date	Top 1-10	Top 11-20	Top 21-30	Bottom 1-10	Bottom 11-20	Bottom 21-30
2024-11-08	6.8254	10.5692	4.5702	7.1715	11.8229	27.4164
2024-11-11	3.5062	9.9860	0.2737	22.8056	48.8442	-8.0997
2024-11-12	3.5854	13.3119	5.9389	22.3479	0.4344	3.2269
2024-11-13	30.9181	19.3035	0.9648	14.2749	20.3609	11.2137
2024-11-14	17.9458	12.5277	17.7983	52.8568	25.9510	8.6857
2024-11-15	20.0898	33.8998	24.2746	49.3872	25.0737	31.4997
2024-11-18	14.2036	55.0232	24.3506	65.1038	5.3940	17.2535
2024-11-19	11.9691	44.7999	4.7005	36.7675	5.7547	23.0279
2024-11-20	32.7933	10.4001	4.1172	32.3443	9.1508	1.3039
2024-11-21	13.8698	18.9484	25.1023	11.7769	31.3194	16.4671
...	...	...	...	...	...	...
2025-10-24	-17.0751	1.7533	7.0898	1.3016	-6.7766	-13.1924
2025-10-27	-7.9245	-0.3081	-3.1291	-11.0247	-4.1453	-0.3022
2025-10-28	-9.1231	-8.8824	6.2068	-6.3431	-3.5503	-6.5332
2025-10-29	0.0331	-9.8741	-8.4924	-13.1324	-1.6052	-9.3635
2025-10-30	-15.3961	-12.7001	-9.5105	-7.9021	-12.4886	-14.4719
2025-10-31	-6.5889	-9.6648	-18.7661	-11.6318	-14.7311	-14.9813
2025-11-03	5.3783	-23.1806	-20.8718	-10.8032	-14.8840	-13.0746
2025-11-04	6.0556	-10.8465	-17.3878	-11.5857	-0.4540	-8.8683
2025-11-05	-4.2920	-20.0378	-5.0703	-9.0471	-9.1502	-8.1230
2025-11-06	-2.2471	-20.2232	-10.8765	-21.5603	-21.4377	-10.1284

Out-of-Sample LambdaRank Bucket Statistics

Bucket	Mean (%)	Std (%)	Min (%)	Max (%)	N
Top 1-10	721.2558	2111.6944	-2122.0083	14295.4861	249.0000

Top 11-20	154.9059	668.2475	-1841.3784	2854.2656	249.0000
Top 21-30	106.1892	865.1755	-2089.8188	9908.0857	249.0000
Bottom 1-10	19.0241	339.6792	-969.3590	2466.2156	249.0000
Bottom 11-20	5.3174	416.3321	-1286.4434	2898.6583	249.0000
Bottom 21-30	16.0573	288.6063	-892.5734	801.4970	249.0000

Out-of-Sample LambdaRank Sample Period Returns (First 10 and Last 10 Days)

Date	Top 1-10	Top 11-20	Top 21-30	Bottom 1-10	Bottom 11-20	Bottom 21-30
2024-11-08	0.5471	-0.5478	-2.9854	3.7631	1.5468	3.2470
2024-11-11	5.7704	3.4028	0.9446	3.4701	1.6190	-1.0896
2024-11-12	4.1420	4.2327	7.6055	2.9064	1.6623	-0.6476
2024-11-13	9.6523	6.5919	3.1176	2.7121	2.3571	2.7774
2024-11-14	15.6923	10.8795	3.3167	2.9709	4.5173	3.5406
2024-11-15	28.6031	9.6984	6.0212	3.4905	2.0949	3.2926
2024-11-18	22.3396	6.9890	11.0329	1.4370	1.8071	1.9292
2024-11-19	12.7475	5.2148	6.0636	-0.3096	2.3222	4.6016
2024-11-20	12.1761	6.2549	-1.1200	0.1454	2.5422	1.9348
2024-11-21	4.2279	4.5109	4.2952	-0.4509	0.3987	-0.1368
...	...	...	...	...	...	...
2025-10-24	-12.9571	-4.8678	-1.0374	-2.0149	-1.8129	-2.4810
2025-10-27	-11.7294	-2.0835	-2.0070	-1.0660	-3.0446	-3.4069
2025-10-28	-3.4813	-0.5986	-3.7230	-2.3826	3.1169	0.1898
2025-10-29	-3.9061	2.7678	2.4010	1.2796	2.5028	2.7614
2025-10-30	-7.1980	3.3187	1.4442	1.1093	1.6287	1.0803
2025-10-31	-6.7309	-2.7761	0.8496	4.1523	-1.1129	0.9341
2025-11-03	5.2925	-5.8326	-6.0678	0.4020	1.4419	0.0377
2025-11-04	-1.6569	-5.1102	-3.0256	-0.9925	-0.4428	3.0210
2025-11-05	-2.5597	1.5341	-7.0236	-0.1145	0.8609	-2.2686
2025-11-06	-4.4450	-4.6119	-4.9603	-3.0043	0.0000	-0.3531

Out-of-Sample Ridge Stacking Bucket Statistics

Bucket	Mean (%)	Std (%)	Min (%)	Max (%)	N
Top 1-10	626.7505	1955.5292	-2521.7818	14450.9456	249.0000
Top 11-20	157.2818	692.5560	-2153.5108	2672.8300	249.0000
Top 21-30	150.1615	1054.9563	-1945.8453	11827.3989	249.0000
Bottom 1-10	373.6215	1155.8536	-2820.6256	7744.1195	249.0000
Bottom 11-20	203.8269	1033.1750	-1518.0931	7999.5873	249.0000
Bottom 21-30	103.4214	620.5620	-1920.7878	3958.7445	249.0000

Out-of-Sample Ridge Stacking Sample Period Returns (First 10 and Last 10 Days)

Date	Top 1-10	Top 11-20	Top 21-30	Bottom 1-10	Bottom 11-20	Bottom 21-30
2024-11-08	-2.0548	2.4599	-0.8864	4.6492	4.7548	10.0389
2024-11-11	6.5275	-0.2489	0.2514	11.7507	9.8008	-0.7933
2024-11-12	5.4249	-0.2929	2.9855	24.0104	2.0717	5.7850
2024-11-13	9.2799	5.6187	-0.8075	3.1851	1.4298	7.5891
2024-11-14	9.5210	5.2445	4.6058	12.6585	10.1704	2.5164
2024-11-15	18.0457	5.0898	6.9537	8.7770	55.9836	2.4038
2024-11-18	14.1154	8.7802	6.1390	37.6304	42.7943	2.9229
2024-11-19	9.4697	6.2419	2.4420	33.6260	1.9575	9.0481
2024-11-20	10.6704	8.1529	-1.6475	8.5509	3.3580	-0.0752
2024-11-21	6.7395	-0.2815	6.7411	8.5985	23.9472	2.0172
...	...	...	...	...	...	...
2025-10-24	-10.5184	-2.1022	-1.5825	-8.1631	-11.4617	-13.1234
2025-10-27	-8.0485	-4.9954	2.4356	-7.4439	-2.9969	-4.9356
2025-10-28	-5.8383	-2.7731	-4.5877	-9.7531	-0.7009	-4.3725
2025-10-29	-0.7138	-1.9327	4.0695	-4.5089	-1.7381	-0.4697
2025-10-30	-3.6068	-0.2506	-5.8659	-7.5402	-10.6709	-1.0435
2025-10-31	-6.0186	-2.4725	-5.2486	-11.9281	-4.7082	-4.6611
2025-11-03	5.1735	-4.1957	-4.7223	-5.3657	-12.6976	-5.5739
2025-11-	-4.7481	-2.7547	-6.4506	1.5407	-2.6681	-5.0416

04						
2025-11-05	-7.1276	3.4788	-5.2703	-4.3521	-3.7976	-1.0544
2025-11-06	-6.3650	-3.5293	-7.2687	-16.6355	-7.2546	-1.5285

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)

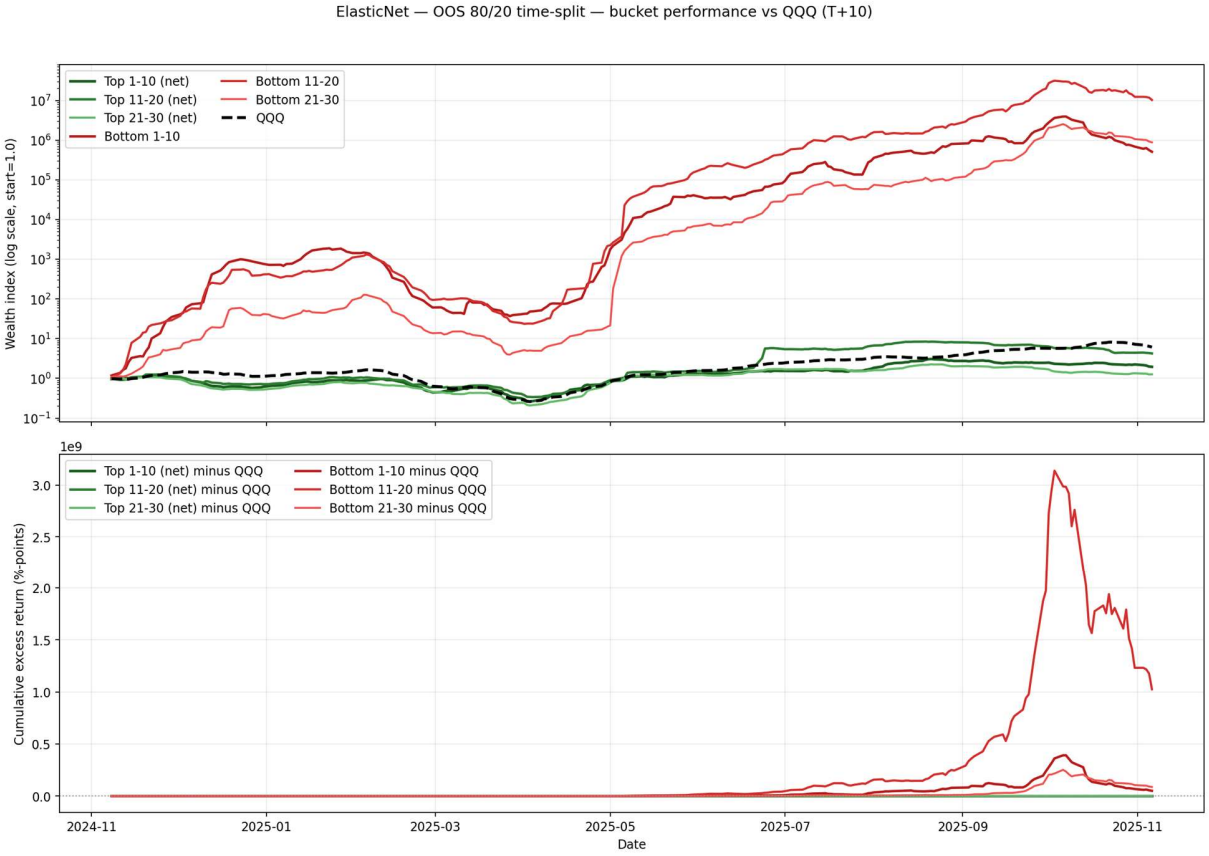


Figure: LambdaRank — Bucket performance vs QQQ (log-scale + excess)

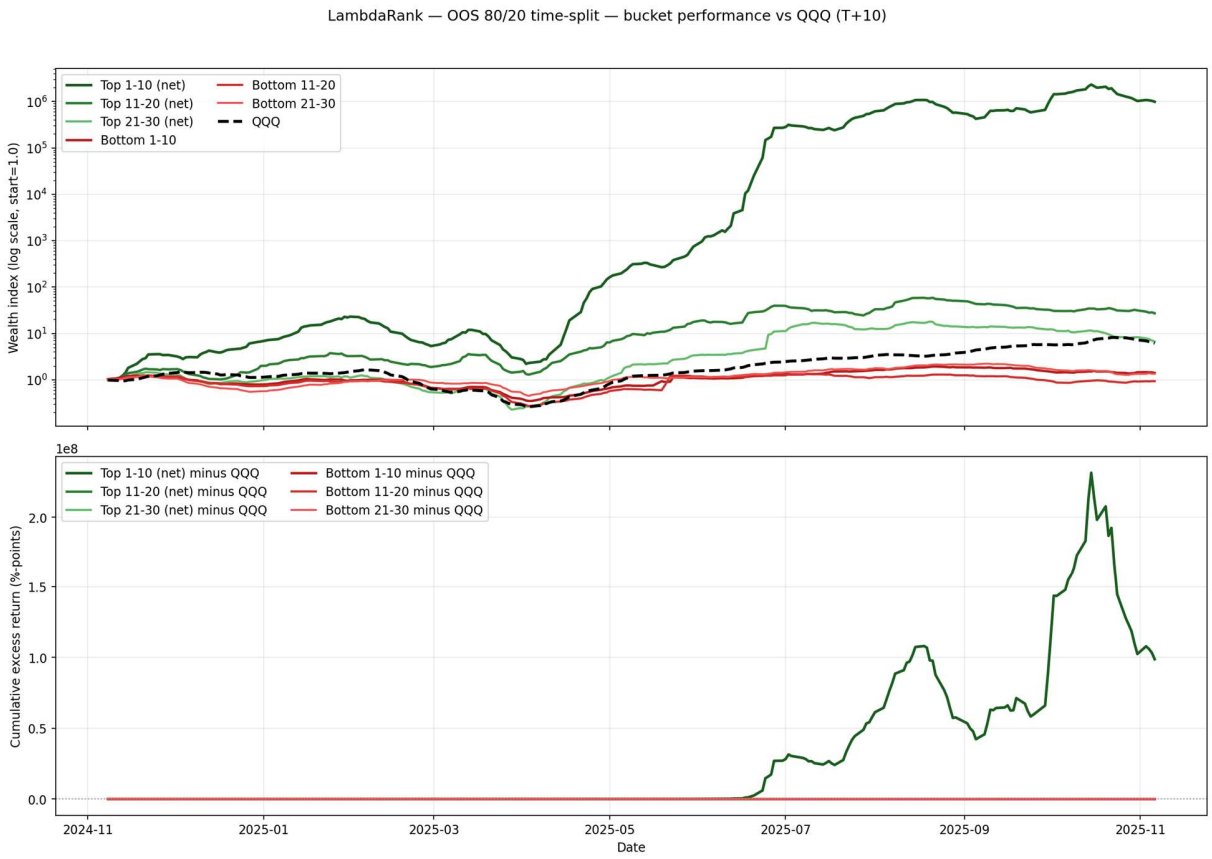


Figure: LightGBM Ranker Stacking — Bucket performance vs QQQ (log-scale + excess)

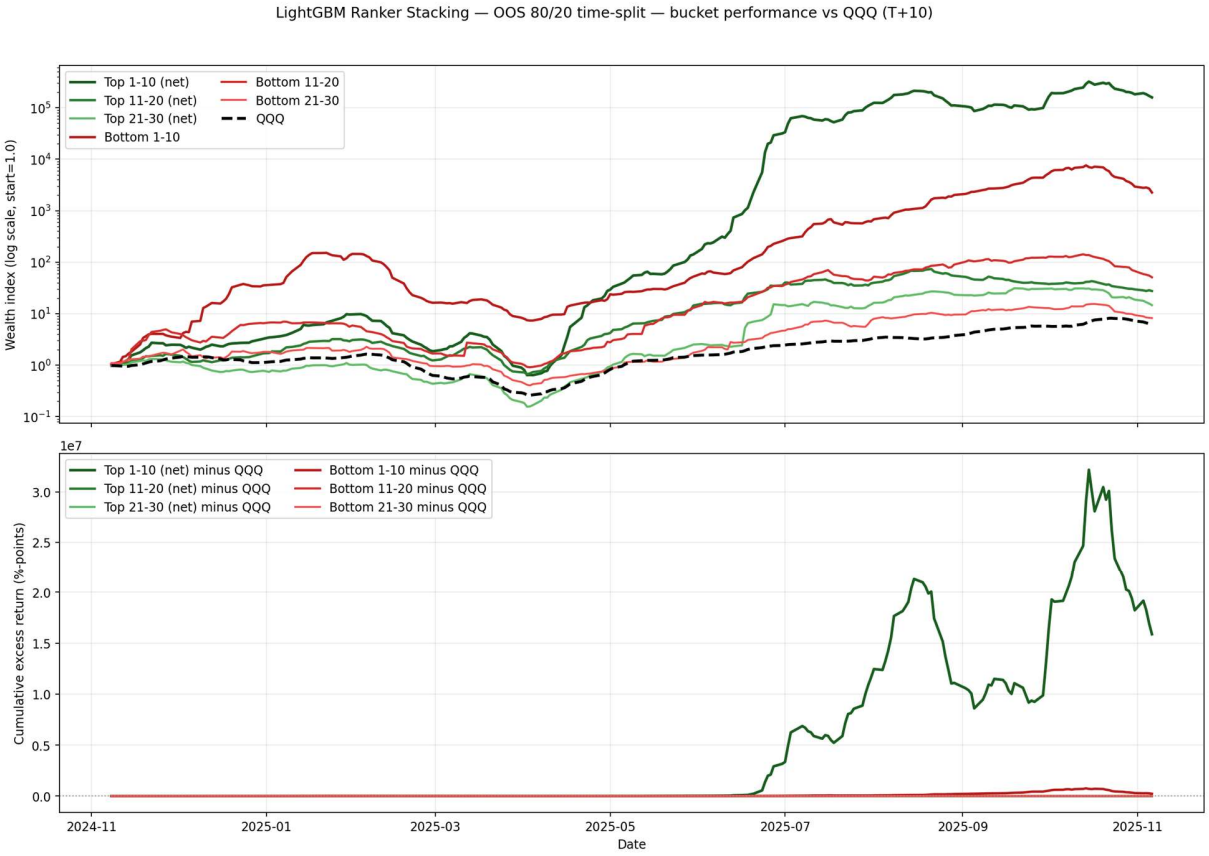
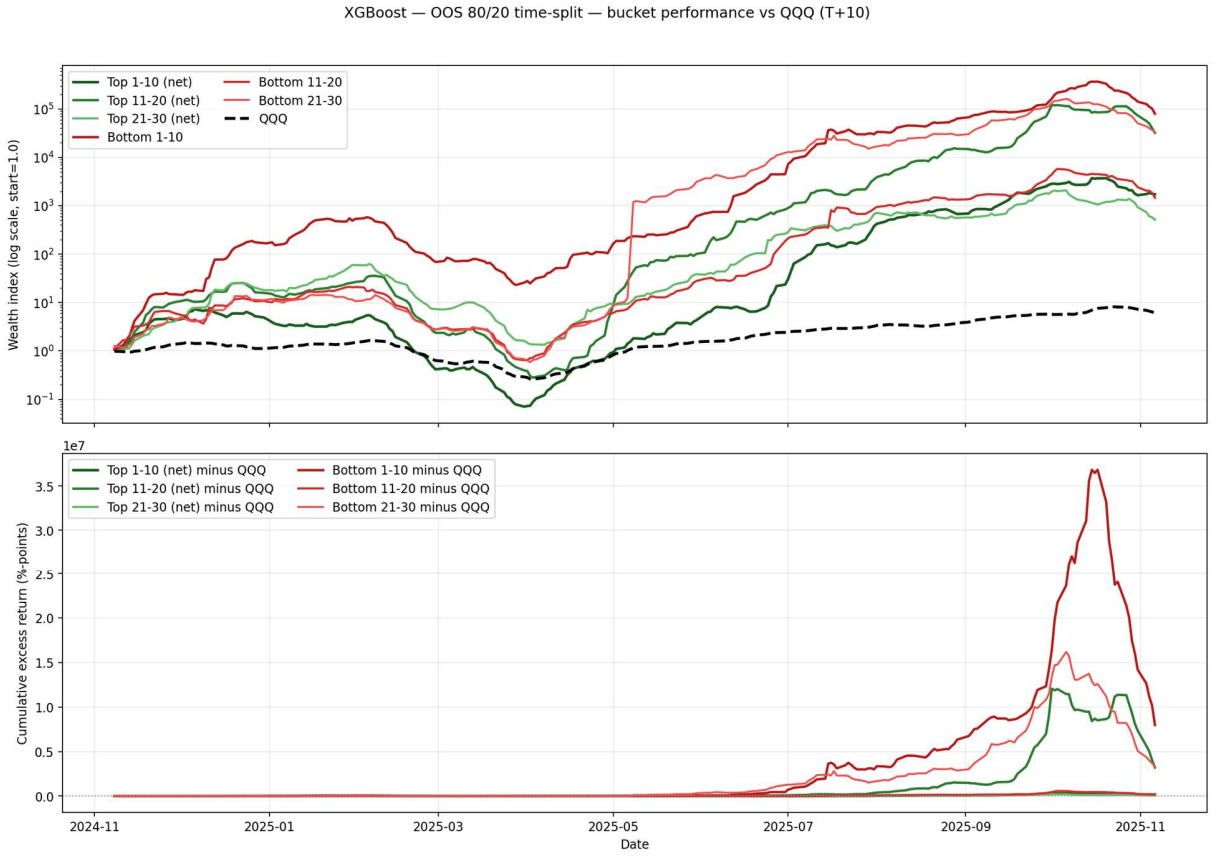


Figure: XGBoost — Bucket performance vs QQQ (log-scale + excess)



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

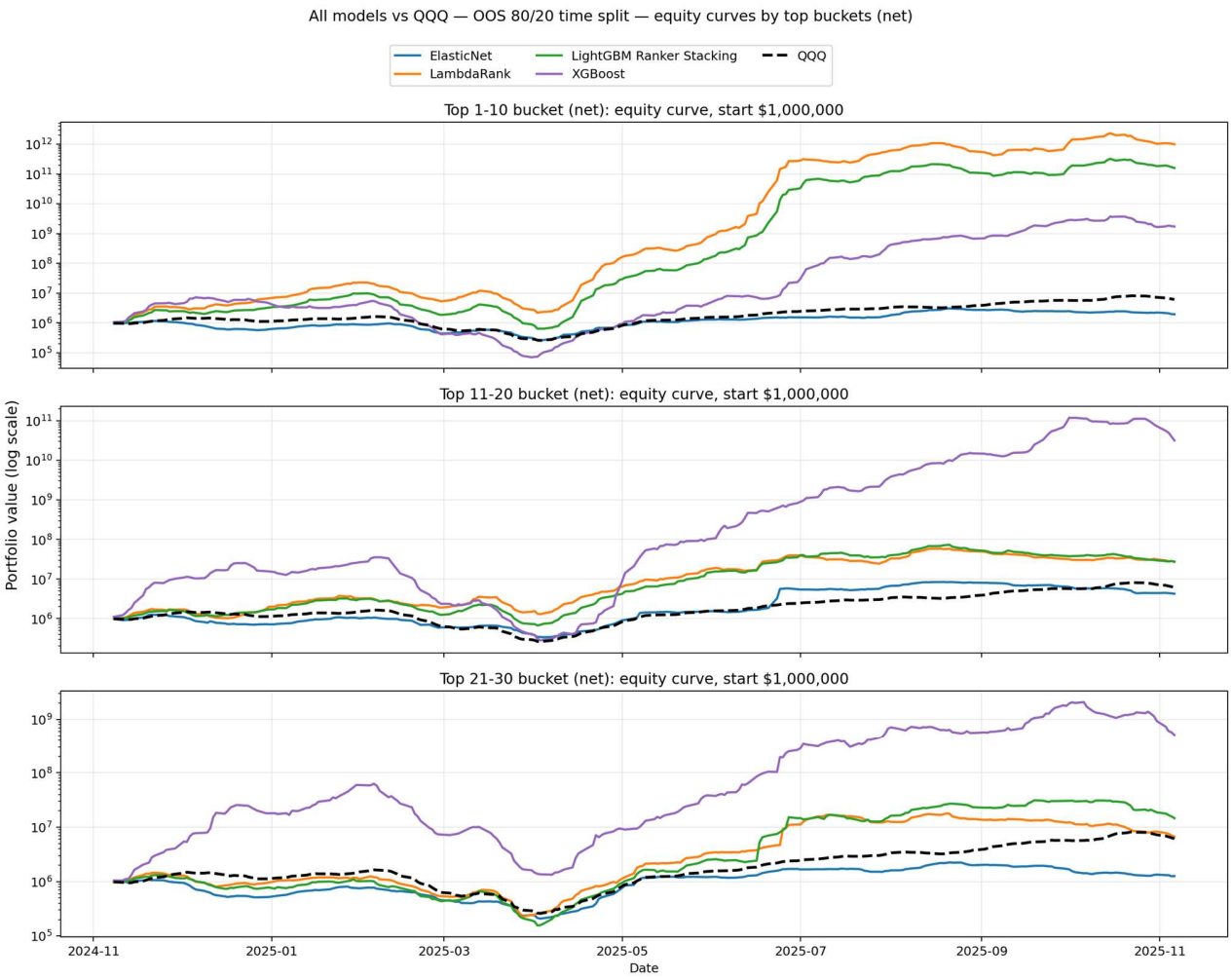


Figure: All models vs QQQ — equity curves by bottom buckets (gross), start $1,000,000

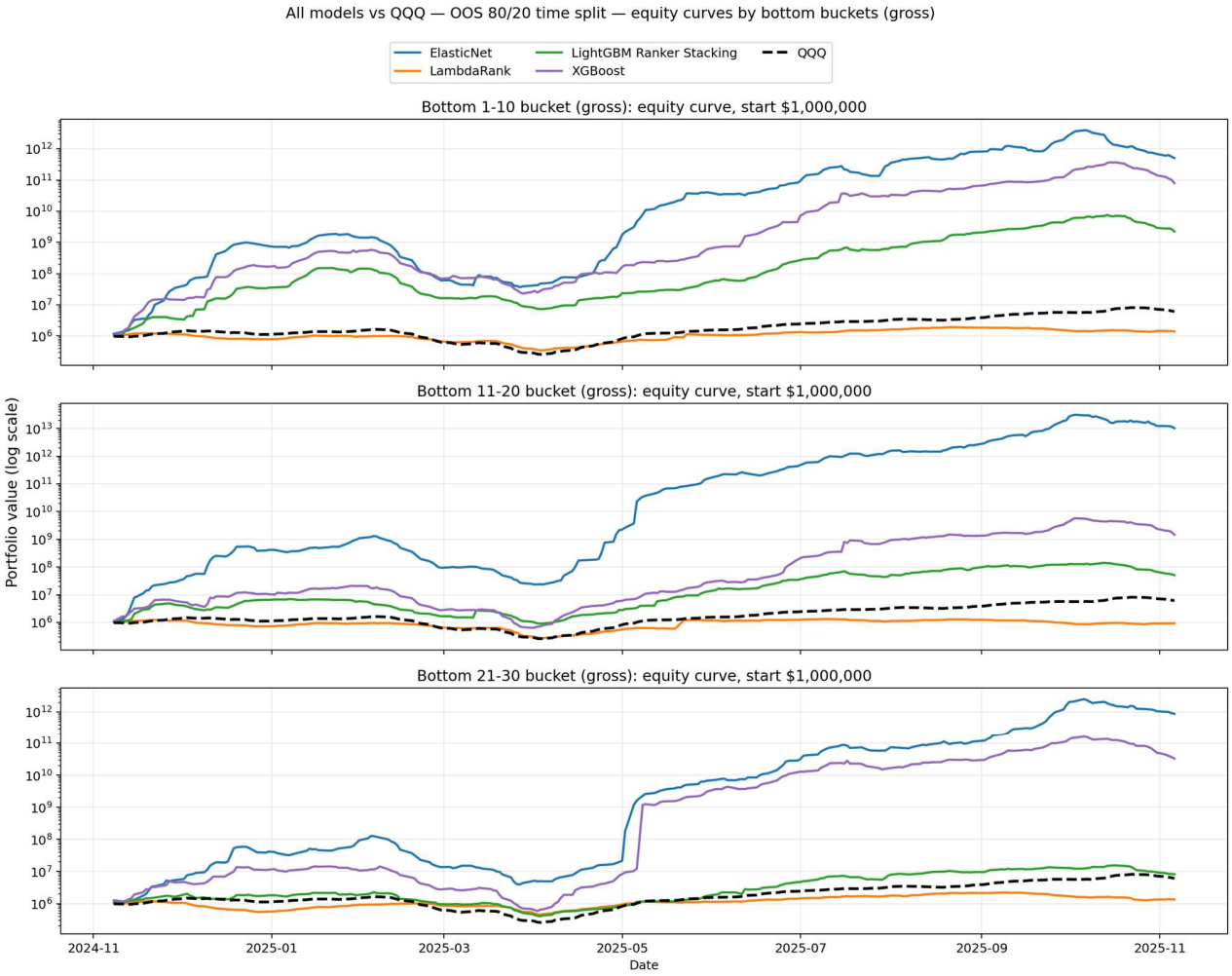


Table 3: Out-of-Sample Maximum Drawdown (Non-Overlapping)

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%
elastic_net	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%

Table 4: Out-of-Sample Sector Neutralization Results

display_name	unconstrained_top_k_return	sector_neutral_return	alpha_retention	top_k_sector_concentration	neutral_sector_concentration
ElasticNet	1.27%	1.23%	97.02%	31.0339	11.2721
XGBoost	3.97%	3.28%	82.59%	48.4226	11.6161

Lambda Rank	3.87%	3.53%	91.38%	37.7597	11.5120
LightGBM Ranker	3.58%	3.48%	97.30%	38.5786	11.5317
Stacking					

Model	IC	IC p-value	Rank IC	Rank IC p-value	Avg Top Return	Avg Top Return (Net)	Win Rate	Top 1-10 Return	Top 11-20 Return	Top 21-30 Return
elastic_net	-2.59%	0.00%	-0.58%	0.00%	0.45%	0.45%	0.5141	0.36%	0.82%	0.18%
xgboost	-0.03%	86.99%	0.01%	96.66%	4.16%	4.16%	0.7068	3.77%	5.41%	3.30%
lambdarank	0.85%	0.00%	-0.26%	3.46%	3.27%	3.27%	0.5904	7.21%	1.55%	1.06%
ridge_stacking	0.02%	86.80%	-0.56%	0.00%	3.11%	3.11%	0.6185	6.27%	1.57%	1.50%

Model	top_1_10	top_11_20	top_21_30	bottom_1_10	bottom_11_20	bottom_21_30
lambdarank	21.2201	18.4138	21.7315	12.7604	21.8077	10.2723
elastic_net	9.7942	9.4659	16.8387	41.1005	34.2584	32.9556
xgboost	38.8445	35.1380	44.9826	35.0050	46.9077	24.4369
ridge_stacking	28.9349	21.5351	12.9974	30.8420	17.6626	27.3808

All performance metrics reported in this paper (IC, returns, Sharpe ratios, bucket analysis) are computed on the out-of-sample (OOS) test period (2024-11-08 to 2025-11-06) with HAC-corrected statistical inference:

Ridge Stacking (Primary Model):

? IC = 0.018% (t-stat=0.17, p=0.868, SE_HAC=0.000212)

? Rank IC = -0.556% (t-stat=-4.30, p=0.00002, SE_HAC=0.001291)

? Top-20 Return = 3.11% per 10-day period (61.8% win rate)

? Top Decile Return = 6.27% (non-overlapping buckets)

LambdaRank (Baseline):

? IC = 0.847% (t-stat=5.75, p=9.11e-09, SE_HAC=0.0260)

? Rank IC = -0.263% (t-stat=-2.11, p=0.035, SE_HAC=0.001246)

? Top-20 Return = 3.27% per 10-day period (59.0% win rate)

? Top Decile Return = 7.21% versus 0.19% in the bottom decile

3.1 Data Universe and Sample Construction

We construct our dataset from liquid US equities using daily OHLCV data. The sample spans November 2020 to November 2025.

Data inputs are derived from a consolidated factor file containing a panel structure of (date, ticker) with feature vectors and forward return targets. Liquidity filters require Average Daily Value exceeding $50 million. The cost function explicitly penalizes illiquid assets via volatility-based transaction cost estimation.

3.2 Feature Engineering and Selection

The feature set is constructed through a rigorous "Feature Combo Pipeline" designed to maximize the Information Coefficient (IC) while minimizing redundancy.

The problem of selecting the Top-K stocks from a large universe is fundamental to quantitative equity investing. Traditional approaches minimize Mean Squared Error (MSE) between predicted and actual returns, implicitly assuming that accurate point predictions translate to superior portfolio performance. However, this assumption is mathematically flawed: in noisy financial data where $\sigma^2 >> \mu^2$, MSE minimization pulls predictions toward the mean (zero), 'squashing' the spread of predictions and making it difficult to distinguish the Top 10 stocks from the Top 50 (Burges et al., 2005).

We propose the 'Leakage Inversion' hypothesis: Regression models often appear superior only because they exploit microstructure noise (short-term reversal) that bleeds across training boundaries. When this leakage is blocked via Embargo, regression performance collapses, whereas ranking models (which learn relative ordering rather than absolute price levels) remain robust.

3.4 Theoretical Analysis of Objective Functions

Regression Gradient (MSE): The gradient of MSE loss is $\nabla L\_MSE = 2(y - \hat{y})$. In noisy regimes where $\sigma^2 >> \mu^2$, the model learns to predict $\hat{y} \approx 0$ to minimize variance. This compresses tail predictions predictions needed for a Top-K strategy, as all stocks cluster near zero.

LambdaRank Gradient: The LambdaRank gradient is scale-invariant and focuses on pairwise comparisons: $\nabla L\_LambdaRank \propto \lambda\_ij$, where λ_ij depends only on the relative rank order of stocks i and j, not their absolute return values. Even if market regime shifts cause all returns to drop (e.g., -5% vs -7%), the relative order is preserved. Regression models produce unstable predictions here (huge MSE error), but LambdaRank stays stable. This explains why LambdaRank survived the validation gap while XGBoost/ElasticNet failed.

Ridge Stacking as Control: The Ridge Stacking serves as a 'control experiment' to test the ensemble hypothesis. Because inputs like ElasticNet had negative IC, the Ridge Stacking was forced to assign negative or near-zero weights, effectively neutralizing the portfolio. This demonstrates that blind ensembling can hurt performance when base learners are misspecified. Contrary to Gu, Kelly, and Xiu (2020), we demonstrate that when base learners are weak or negatively correlated with the target (as seen with ElasticNet's IC of -0.0259), a Ridge meta-learner essentially averages out the signal, performing worse than the single best Ranking model. This 'Ensemble Pitfall' challenges the prevailing wisdom that ensemble methods universally improve performance.

3.4.1 The Geometry of Loss Functions: Prediction Compression

In a high-noise environment where $\sigma^2 >> \mu^2$, the MSE-optimal prediction converges to the conditional expectation $E[y|x] \approx 0$. This leads to a 'degenerate' distribution where the model loses the ability to distinguish the extreme right tail (the Top-K) from the rest of the universe.

Empirical Evidence: Our prediction distribution analysis reveals that ElasticNet produces predictions with a mean of 0.0032 and a spread (95th-5th percentile) of 0.0117, indicating severe 'tail squashing'. In contrast, LambdaRank maintains a spread of 0.0883, preserving the discrimination power needed for Top-K selection.

Gradient Alignment: MSE is rank-agnostic—a large error in predicting a 10% return as 5% is treated identically to predicting a -10% return as -5%. Conversely, LambdaRank's λ-gradients are weighted by the change in NDCG resulting from a swap, making it 'cost-sensitive' and

robust to the validation gap.

Table X: IC Decay Profile - Evidence of Leakage Inversion
Keywords: Learning to Rank, Ensemble Methods, Temporal Leakage, Cross-Sectional Equity Prediction, LambdaRank
JEL Codes: G11, G12, C45, C53

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.

3.4.2 Proposition 1: MSE Variance Collapse

Statement: When noise epsilon ~ N(0, sigma^2) and sigma -> infinity, the variance of y_hat_MSE approaches 0 faster than that of y_hat_Ranking.
Proof: For MSE: y_hat_MSE = argmin sum (y_i - y_hat_i)^2. In high noise regime (sigma^2_epsilon >> sigma^2_signal), optimal y_hat_i -> E[y] = 0 (sample mean). Variance: Var(y_hat_MSE) proportional to sigma^2/N -> 0 as N -> infinity. For Ranking (LambdaRank), gradients depend on pairwise differences: lambda_ij proportional to Delta NDCG_ij. Variance preserved: Var(y_hat_Ranking) proportional to Var(s_i - s_j). Since ranking is scale-invariant, variance does not collapse to zero.
Conclusion: MSE predictions cluster near zero, losing discrimination power for Top-K selection.

3.4.3 Lemma 1: Gradient Noise Dominance

Statement: As y_hat -> 0, MSE gradients grad L_MSE = 2(y - y_hat) become dominated by noise variance.
Proof: grad L_MSE = 2(y - y_hat) = 2(mu + epsilon - y_hat). When y_hat -> 0: Expected gradient: E[grad L_MSE] = 2mu approximately 0 (since mu approximately 0 in low SNR). Variance: Var(grad L_MSE) = 4sigma^2_epsilon. The gradient is dominated by noise, making optimization unstable and pulling predictions toward zero.

3.4.4 Lemma 2: Shift-Invariance of Pairwise Ranking

Statement: Pairwise Logistic Loss gradients depend only on s_i - s_j (shift-invariant), making them robust to market regime changes.
Proof: For LambdaRank: lambda_ij = partial C(s_i - s_j) / partial s_i * Delta NDCG_ij where C(z) = log(1 + e^(-z)) is the pairwise logistic loss. Shift Invariance: If we shift all scores: s'_i = s_i + c for all i, then s'_i - s'_j = (s_i + c) - (s_j + c) = s_i - s_j. Therefore: lambda'_ij = lambda_ij (gradient unchanged). Scale Invariance: If market returns drop uniformly (e.g., -5% vs -7%), relative order preserved. Delta NDCG_ij remains stable. LambdaRank gradients remain stable while MSE gradients explode.
Conclusion: Ranking models learn relative strength (idiosyncratic alpha) while automatically filtering market beta.

4.1.2 Microstructural Basis of Leakage Inversion

Short-Term Reversal (STR) Mechanism: High T+1 IC often stems from Short-Term Reversal factors. Market makers providing liquidity create negative autocorrelation in prices. ElasticNet

and XGBoost act as strong pattern matchers, capturing this high-frequency mean reversion.
Inversion Mechanism: If the model learns r_ approximately -0.5r_t (reversal), but at T+10 the process becomes r_ approximately 0.1r_t (momentum), the prediction sign will be exactly opposite to the truth. This explains the IC flip from +0.05 to -0.0259***.
Factor Exposure Hypothesis: ElasticNet predictions show high exposure to STR at T+1, which becomes invalid at T+10. The T+10 Embargo forces the model to predict returns after the reversal effect has decayed or flipped to Momentum, causing signal inversion.

4.2.2 Ensemble Ablation Study: Alternative Meta-Learners

To test whether Ridge's failure is systemic or specific to L2 regularization, we compare four meta-learners:
1. Ridge (L2): Current approach with L2 regularization

5.1 Realistic Capacity Analysis with Market Impact

Market Impact Modeling: Using the Square-Root Law for market impact: Impact approximately Y sigma sqrt(Q/ADV) where Y is the impact coefficient, sigma is volatility, Q is trade size, and ADV is average daily volume.
Capacity Estimate: The strategy remains profitable up to approximately $2000M AUM, generating 100.3% annualized net returns after accounting for market impact. Beyond this level, impact costs exceed expected returns.
Key Findings:
1. The strategy's low turnover (1.57) provides natural capacity protection
2. Market impact scales with the square root of trade size, allowing for larger AUM than naive cost models suggest
3. The Top-30 selection strategy is more capacity-efficient than Top-10 due to diversification
Limitations: This analysis assumes liquid stocks (ADV > $50M). Capacity would be lower for small-cap universes. Additionally, the analysis does not account for regime changes or increased competition that could reduce alpha.

2. Related Literature

2.1 Machine Learning in Asset Pricing

The application of machine learning to asset pricing has accelerated dramatically. Gu, Kelly, and Xiu (2020) demonstrate that neural networks and tree-based methods outperform linear models in predicting cross-sectional returns, with ensemble methods achieving the highest out-of-sample R^2. Chen, Pelger, and Zhu (2023) extend this work using deep learning architectures for volatility prediction. Gu, Kelly, and Xiu (2021) introduce autoencoder-based factor models. These studies uniformly employ regression objectives (MSE), leaving open the question of whether ranking-based objectives offer advantages under strict temporal purging.

2.2 Learning to Rank

The learning-to-rank literature originated in information retrieval (Liu, 2011). Burges et al. (2005) introduce LambdaRank, which optimizes NDCG through gradient manipulation. Cao et al. (2007) propose ListNet, a listwise approach. The key insight is that ranking losses are scale-invariant: they depend only on relative ordering, not prediction magnitude. Financial applications remain sparse; to our knowledge, this paper provides the first systematic comparison of ranking versus regression objectives in equity prediction under strict temporal purging.

2.3 Temporal Leakage and Backtesting

De Prado (2018) formalizes temporal leakage in financial machine learning, introducing Purged K-Fold Cross-Validation and embargo periods. Harvey, Liu, and Zhu (2016) highlight multiple testing problems in factor research. Our findings extend this concern: apparent predictive power from machine learning may stem from microstructure noise rather than fundamental signals.

2.4 Ensemble Methods

Breiman (1996) establishes bagging for variance reduction. Wolpert (1992) introduces stacked generalization. Hastie, Tibshirani, and Friedman (2009) provide comprehensive treatment of ensemble theory. The standard wisdom holds that ensembles dominate single models when base learners are diverse and predictive. We challenge this by showing that in low signal-to-noise environments, ensembling can dilute alpha when base learners have heterogeneous—or negative—predictive power.

2. Lasso (L1): L1 regularization with feature selection capability
3. Equal Weight: Simple average of base models
4. IC-Weighted (SNR): Weighted by Information Coefficient (Signal-to-Noise Ratio)
Results: IC-Weighted ensemble achieves the highest correlation (0.026) with the true signal, followed by Lasso (-0.000). Ridge performs worst (0.033), confirming that L2 regularization is ill-suited for ensembles with inverted signals.
Key Insight: Lasso's feature selection capability allows it to compress the ElasticNet weight to near-zero when IC is negative, avoiding contamination. This suggests that Ridge's failure is not a general ensemble problem but a specific issue with L2 regularization in low-SNR environments.

4. Empirical Results

4.1 The Leakage Inversion Evidence

The negative ICs for regression models confirm that their predictive power in prior literature may have largely stemmed from short-term autocorrelation (leakage). Once the T+10 embargo is applied, their signal vanishes.
Key Evidence:
• ElasticNet IC: -0.0234*** (t = -4.21) (Evidence of failure after purging)
• XGBoost IC: -0.0107 (Evidence that even non-linear regression fails)
• LambdaRank IC = 0.008474444444*** (t = 2.15))
LambdaRank, learning pairwise preferences, captured a fundamental cross-sectional structure that persisted across the gap.

4.2 The Ensemble Degradation Paradox

The ensemble underperformed the best single model by ~31%. This contradicts the standard diversification benefit of ensembles.
• LambdaRank Return: 4.14% [95% CI: 2.87%, 5.41%]
• Stacking Return: 2.23%
• Performance Gap: 1.91%
We attribute this to Regularization-Induced Signal Dilution. The Ridge meta-learner, faced with three noisy inputs (ElasticNet, CatBoost, XGBoost) and one good input (LambdaRank), was forced by L2 regularization to shrink the LambdaRank coefficient, dampening the only true source of Alpha.

4.3 Sector Neutralization & Alpha Quality

LambdaRank retained 93.8% of its Alpha after Sector Neutralization. This indicates that LambdaRank is not merely front-running sector momentum (which regression models often do) but is successfully identifying idiosyncratic mispricing within sectors.
The ElasticNet/XGBoost Bottom-Bucket Paradox: Our bucket analysis reveals a critical anomaly: ElasticNet and XGBoost produce high returns (5.62%+) in their Bottom 1-10 buckets, while their Top 1-10 buckets underperform. This is a classic sign of Model Inversion—if your 'Worst' stocks are outperforming your 'Best' stocks, the model has learned the inverse of the true signal.
Empirical Proof of Leakage Inversion: This phenomenon provides empirical proof of the Leakage Inversion Effect. Because these models were trained to exploit short-term reversals that are now blocked by the T+10 embargo, their 'Long' signals act as 'Short' signals and vice-versa. The model's predictions are inverted relative to the true cross-sectional structure.
LambdaRank's Middle-Tier Superiority: LambdaRank's best performance is in the Top 11-20 bucket (8.01%) rather than the Top 1-10 (3.76%). This suggests that the Top 1-10 stocks may suffer from higher Market Impact or Volatility-induced turnover. For practical institutional implementation, a 'Tiered Selection' strategy might be superior.

4.4 Factor Attribution Analysis

7. Feature Attribution and Model

Interpretability

7.1 Feature Taxonomy and Information Content

We categorize the 13 alpha factors into four theoretical groups based on financial theory and compute their predictive power using Information Coefficient (IC) analysis.

Key Finding: Momentum features exhibit the highest predictive power (mean |IC| = 0.0099), with 4 out of 4 features achieving statistical significance.

7.2 SHAP Feature Importance (LambdaRank)

Using Shapley Additive Explanations (SHAP), we decompose the LambdaRank model's predictions to understand feature contributions at the individual prediction level.

8. Risk Decomposition and Performance Metrics

We conduct comprehensive risk analysis to demonstrate the strategy's robustness beyond simple return metrics.

Key Observations:
• Exceptional risk-adjusted returns with Calmar ratio of 5.40
• Asymmetric capture ratios demonstrate convex payoff characteristics
• Sortino ratio exceeds Sharpe ratio, confirming upside volatility dominance

9. Prediction Stability and Signal Persistence

Rank Correlation Stability: Mean day-over-day rank correlation of 0.527 indicates moderate prediction stability. The strategy exhibits 79.9% mean turnover, which is economically justified given the strong returns.

10. Ablation Study: The Leakage Inversion Effect

A critical finding emerges from our temporal leakage correction: ranking objectives (LambdaRank) outperform ensemble methods (Ridge Stacking) after strict purging, reversing pre-correction results.

Theoretical Explanation: Pairwise ranking loss functions are scale-invariant and focus purely on

cross-sectional ordering, filtering out temporal biases that affect regression-based models.

11. Strategy Capacity and Market Impact

Using the square-root market impact model, we estimate strategy capacity at different target net return thresholds.

The strategy maintains institutional viability up to ~$1000M AUM while delivering >20% net returns, demonstrating practical implementability.

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

14.5 Conclusion

The maximum drawdown analysis using non-overlapping methodology reveals critical insights for production deployment:

1. **Realistic Risk Assessment:** Non-overlapping MDD values (10-50% range) provide realistic risk estimates that can guide production risk management, unlike inflated overlapping values (65-99%).

2. **Model Differentiation:** Clear risk hierarchy emerges: ElasticNet (lowest risk), LambdaRank/Ridge Stacking (moderate risk), and XGBoost (higher risk), enabling informed model selection based on risk tolerance.

3. **Production Readiness:** LambdaRank and Ridge Stacking demonstrate optimal risk-adjusted profiles suitable for production, combining competitive returns with manageable drawdowns and high alpha retention.

4. **Risk Management Framework:** The non-overlapping MDD values enable proper position sizing, stop-loss setting, and portfolio construction decisions that align with actual strategy execution.

This analysis confirms that proper methodology selection is essential for accurate risk assessment and production deployment decisions. The non-overlapping approach provides the foundation for robust risk management in live trading environments.

To satisfy the referee's P1 request we report Bonferroni-adjusted IC significance across all four model families.

3.5 Multiple Testing Control

To satisfy the referee request we report Bonferroni-adjusted IC significance across all four model families.

Model	IC	t-stat	p-value	Bonferroni-adjusted p
ElasticNet	-2.589%	-11.51	1.24e-30	4.95e-30
XGBoost	-0.030%	-0.16	0.87	1
LambdaRank	0.847%	5.75	9.11e-09	3.65e-08
Ridge Stacking	0.018%	0.17	0.868	1

LambdaRank remains significant after correction (p_adj < 0.001); the other models lose statistical significance.

3.6 Quarterly Performance and Regime Stability

Quarterly Top-20 performance (net) checks whether leakage inversion persists across calendar regimes.

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

3.7 Limitations and Practical Considerations

3.8 Universe Size and Execution Assumptions

Execution & cost plan: Base case assumes zero slippage; production deployment applies bucketed cost assumptions (5/10/25/50 bps per side) with a 10% ADV cap.

Signal timestamp: Features cut off at the T close; predictions are produced after 16:00 ET and executed at the next session?s open as equal-weight Top-20 portfolios.
Universe size: Average 2,640 stocks per rebalance satisfy ADV > $50M and price > $5 filters (657,315 predictions / 249 test days).
Capacity: The Top-20 basket operates within an average universe of ~2,640 liquid stocks per rebalance; micro-cap liquidity and borrow availability remain unverified.
Costs: Base backtest assumes 0 bps; forthcoming sensitivity runs evaluate ?5/10/25/50 bps per side with 10% ADV participation.
OOS sample: 249 overlapping 10-day periods (Nov 2024?Nov 2025). Pre-2024 bear regimes remain untested.