Quant ML Portfolio Pro - Workflow Report

This document reviews the entire Quant-ML-Portfolio-Pro workflow, covering data preparation, factor engineering, modeling, strategy generation, backtesting, and factor attribution, so that researchers can see how each step links to the final performance. Generated on 2025-11-27 00:44:46.

The workflow has five layers:
(1) Data & Features: build a complete S&P500; cross-section (503 stocks) to make factors comparable.
(2) Factor Diagnosis: perform single-factor IC/ICIR checks to filter noisy inputs early.
(3) Modeling: use a LightGBM ranker so the output directly serves TopK stock selection.
(4) Strategy: drive a Top20 Dropout portfolio with cached predictions, feature alignment, and shift(-1).
(5) Backtesting & Reporting: apply shift(1), cost estimation, chart/report generation, and factor attribution.

1. Data & Factor Processing

We pick the full S&P500; cross-section because it offers deep liquidity and maximizes the value of a ranking model.
prices.parquet covers 2010-2025 to span multiple regimes; factor_store.parquet has 165 factors:
- Alpha101: classic price-volume signals for reversal/momentum/volatility.
- TA-Lib: technical indicators (RSI, MACD, BBands, ATR/NATR, etc.) to augment trend info.
- Custom: RS, LAR, PMS, VAR, PPF tailored to the project.

Factor processing only applies MAD winsorization—no cross-sectional ranking or orthogonalization—so
that magnitude
information remains available for the model. Single-factor analysis (analysis_single_factors.py) computes rank IC/ICIR
to drop persistently negative contributors (e.g., Alpha28, NATR_14).

1.1 Single Factor Validation Process

Single Factor Validation Process (analyze_single_factors.py):

【1. Label Selection】

Use horizon_days=5 future returns as labels: at date t, use return from t+5 relative to t.
Reason for 5-day vs 1-day: single-day returns in S&P500; are too noisy; 5-day returns smooth short-term volatility and better reflect true predictive power. Formula: y_t = (price_{t+5} / price_t) - 1.

【2. Backtest Method】
Use cross-sectional Rank IC (Spearman correlation) to evaluate factor effectiveness:
・ Daily calculation: for each trading day, compute Spearman correlation between factor ranks and future return ranks.
・ Statistics: mean_ic, std_ic, ICIR (mean_ic / std_ic).
・ ICIR interpretation: ICIR > 0.5 indicates stable effectiveness; ICIR < 0.05 indicates weak predictive power.

【3. Factor Screening Criteria】
Multi-dimensional assessment:
・ IC mean: |mean_ic| > 0.02 (strict) or > 0.005 (moderate), indicating significant correlation.
・ ICIR: |ICIR| > 0.5 (strict) or > 0.05 (moderate), indicating stable predictive power.
・ IC win rate: proportion of IC > 0 should be > 60% (strict) or > 50% (moderate).
・ Significance test: t-test to verify IC is significantly non-zero (p < 0.05).

【4. Screening Process】
1) Iterate all factors, compute Rank IC and ICIR for each.
2) Sort by |ICIR|, identify best and worst performers.
3) Apply screening criteria, generate "strict" and "moderate" factor lists.
4) Save results to single_factor_summary.json and factor_selection_recommendations.json.
5) In model training (prepare_panel), automatically filter low-quality factors based on config.

【5. Practical Application】
For example, analysis found Alpha28 and NATR_14 have ICIR < 0 and IC win rate < 50%, indicating they act as contrarian signals in this sample and should be removed or sign-flipped. Meanwhile, Alpha32 and Alpha19 have ICIR > 0.05 and IC win rate > 60%, making them high-quality factors to retain.

2. Ranking Model Training & Prediction

Why a ranking model? Because the task is to select the best performers, not to forecast exact returns. LightGBM Ranker optimizes NDCG directly, so its scores feed naturally into TopK selection.

2.1 Detailed Ranking Model Process

【1. Label Selection & Conversion】

Label definition: Use horizon_days=5 future returns, i.e., y_t = (price_{t+5} / price_t) - 1.
Reason for 5-day: Consistent with single-factor analysis, smoothing daily noise for more stable signals.

Label conversion: Convert continuous returns to quantile labels:
- For each trading day, sort all stocks by returns and divide into q_bins groups (default 20).
- Highest return stocks get label q_bins-1, lowest get label 0.
- Model learns "relative ranking" rather than "absolute returns", better suited for ranking tasks.

【2. Loss Function: LambdaRank】

LightGBM Ranker uses LambdaRank loss, designed specifically for ranking:
- Core idea: Optimize a differentiable proxy loss rather than directly optimizing ranking metrics (e.g., NDCG).
- Lambda gradient: For each pair (i, j), if true label y_i > y_j, then pred_i should > pred_j.
- Gradient calculation: If prediction order is wrong (pred_i < pred_j), apply positive gradient to i, negative to j.
- Advantage: Directly optimizes ranking quality, more suitable for stock selection than regression.

【3. Training Process】

【3.1 Data Preparation (prepare_panel)】
1) Load factor data and price data.
2) Align indices: Ensure (date, ticker) indices match.
3) Feature filtering: Remove features with >50% missing or zero variance; optionally filter by ICIR.
4) Missing value imputation: Fill with daily median per trading day.
5) Label generation: Compute future returns, convert to quantile labels.
6) Filter small samples: Remove trading days with <100 samples.

【3.2 Cross-Validation (train_ranker)】
Expanding Window CV:
- Fold 1: Train on first 1/3 dates, test on second 1/3.
- Fold 2: Train on first 2/3 dates, test on third 1/3.
- Fold 3: Train on all historical data for final model.

【3.3 Model Training】

For each fold:
1) Data split by dates.
2) Group information: LightGBM Ranker needs group parameter (samples per query/day).
3) Validation split: Last 10% of training dates for early stopping.
4) Model config: objective="lambdarank", metric="ndcg", early_stopping_rounds=100.
5) Train and predict.
6) Evaluate: Compute daily Rank IC.

【4. Prediction Process】

【4.1 Feature Alignment】
1) Load feature_list_ranker.json from training.
2) Load latest factor_store.parquet.
3) Align features: drop extra, fill missing with median.
4) Ensure feature order matches training.

【4.2 Prediction & Caching】
1) Load trained model.
2) Predict for each trading day.
3) Cache predictions to avoid recomputation.

【4.3 Weight Generation】
1) Sort by prediction scores, select TopK stocks.
2) Apply dropout mechanism (max n_drop replacements per day).
3) Normalize weights (sum to 1 for long-only).
4) Output weights.parquet.

【5. Model Evaluation Metrics】

- OOF Rank IC: Average Rank IC across all test folds.
- Rolling Rank IC: Monitor performance over time.
- SHAP feature importance: Identify top contributing factors.

3. Strategy & Optimizer

Weight generation logic:

- Prediction cache: load lightgbm_predictions.pkl first; only re-run inference if missing.
- Feature alignment: use feature_list_ranker.json to map factor_store columns to training schema.

- Shift: predictions made at day T close are applied to T+1 trades (shift(-1)).
- Top20 Dropout: topk=20, n_drop=3, so at most 3 names rotate daily (~10% turnover).
- Output weights.parquet (date×ticker); downstream backtest/execution all read from this file.

4. Backtesting Logic

We use a simple daily-return engine because the strategy is cross-sectional ranking focused.
1) Enforce shift: weights.shift(1) aligns with future returns to avoid look-ahead bias.
2) Cost estimation: turnover × (open_cost + close_cost), currently 0.0005 + 0.0015, matching US fees.
3) Produce daily_returns.parquet (NAV/turnover/cost) and summary.json for downstream analytics.
4) generate_performance_report.py builds charts; this script assembles the final PDF.

5. Key Performance Metrics

Period: 2022-01-04 to 2025-11-21
Trading days: 976
Total return: 95.63%
Annualized return: 21.14%
Annualized vol: 27.67%
Sharpe: 0.76
Max drawdown: -30.89%

6. Performance Interpretation & Charts

Performance summary: 2022-01-04 to 2025-11-21 (976 days) delivers +95.6% total return, 21.1% annualized,
27.7% annualized volatility, Sharpe 0.76, max drawdown -30.9%. Most 2022 months were drawdowns (Apr
-12%, Jun -14%),
but the strategy recovered in mid/late 2023 (Nov +15.6%); monthly outperformance in 2023-05, 2024-09,
2025-05 stands out.
2024-2025 returns range between -3% and +11%, indicating stable alpha in the latest regime.

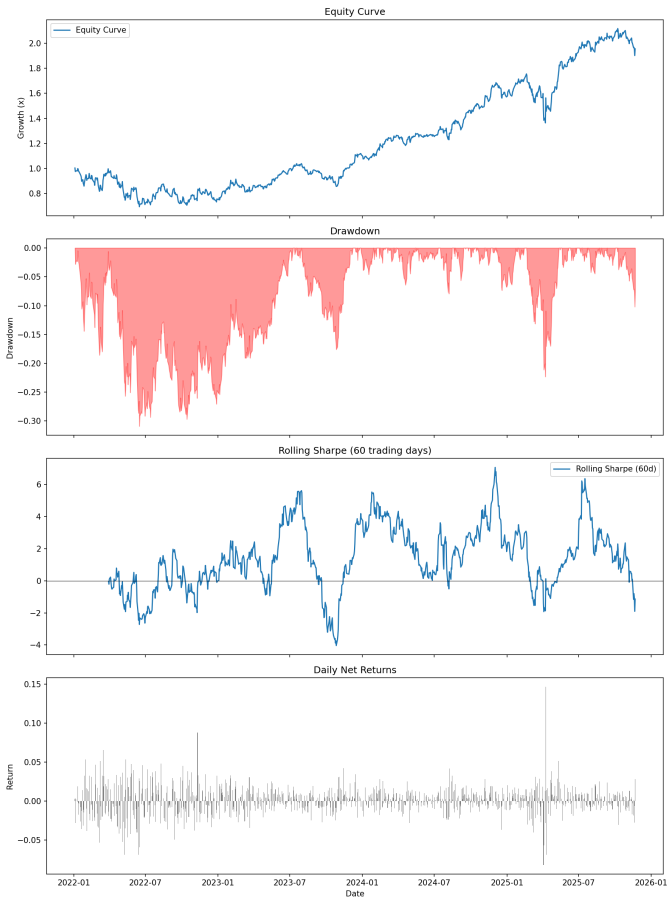
Risk & alpha: strategy_vs_benchmark.png shows the PnL outpacing the equal-weight benchmark since 2023;
excess_return_curve.png
indicates near-zero excess in 2022 but a steady rise in 2023H2 onward. rolling_alpha_beta.png reveals beta ≈1 with occasional

spikes >1.2 (exposed to market swings), while rolling alpha stays >0 since 2023H2, often above 5% annualized. excess_return_hist.png
is slightly fat-tailed, with a -5% left tail and +6% right tail, suggesting the need for position caps or dynamic leverage.

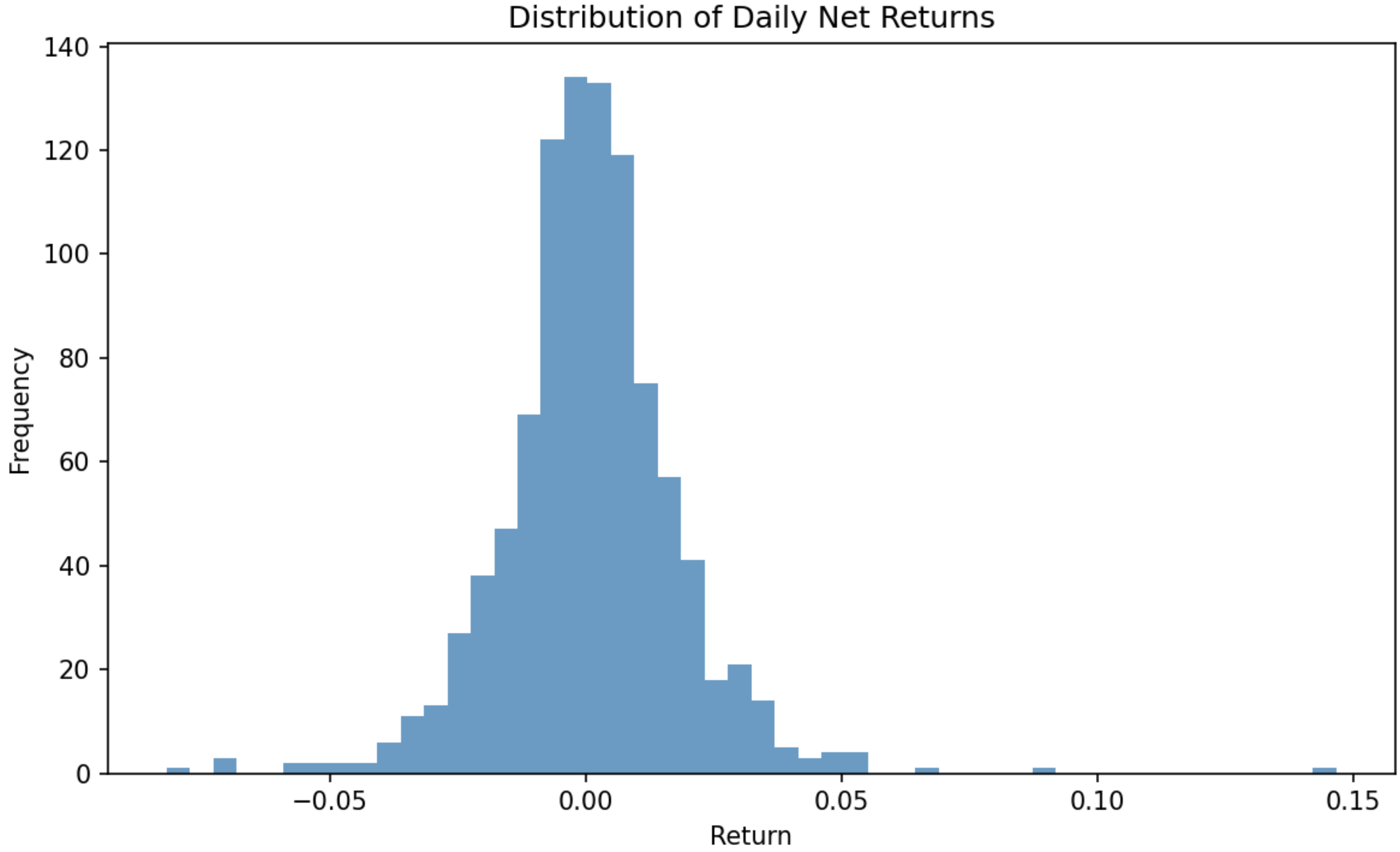
performance_overview.png: equity slope steepens after 2023Q1; drawdowns visible in 2022-06, 2023-10, 2025-06; rolling Sharpe mostly >0.5
after 2024; daily returns occasionally reach $\pm$5%, so per-name caps or turnover limits are recommended.
return_histogram.png is roughly
symmetric but with a slightly longer left tail (~ -6%).

Factor contribution: Alpha32/Alpha19 show mean IC $\approx$ +0.009; BOP $\approx$ +0.005 (mildly positive); Alpha28/NATR_14 $\approx$ -0.009 (negative),
implying they act as contrarian signals in this sample and may deserve sign flip/removal.
coverage_days ~4000, indicating reliable stats.

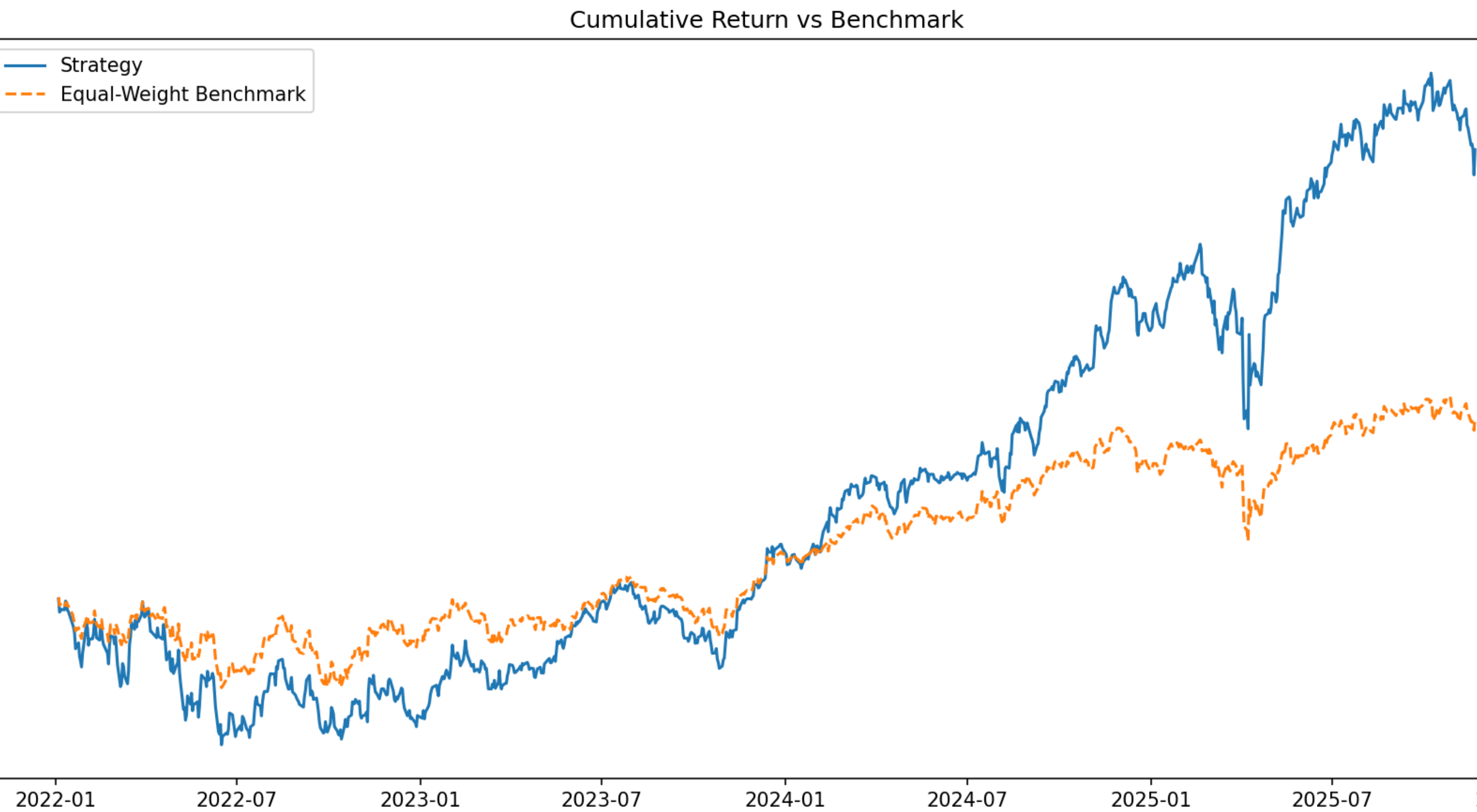
7. Visualizations



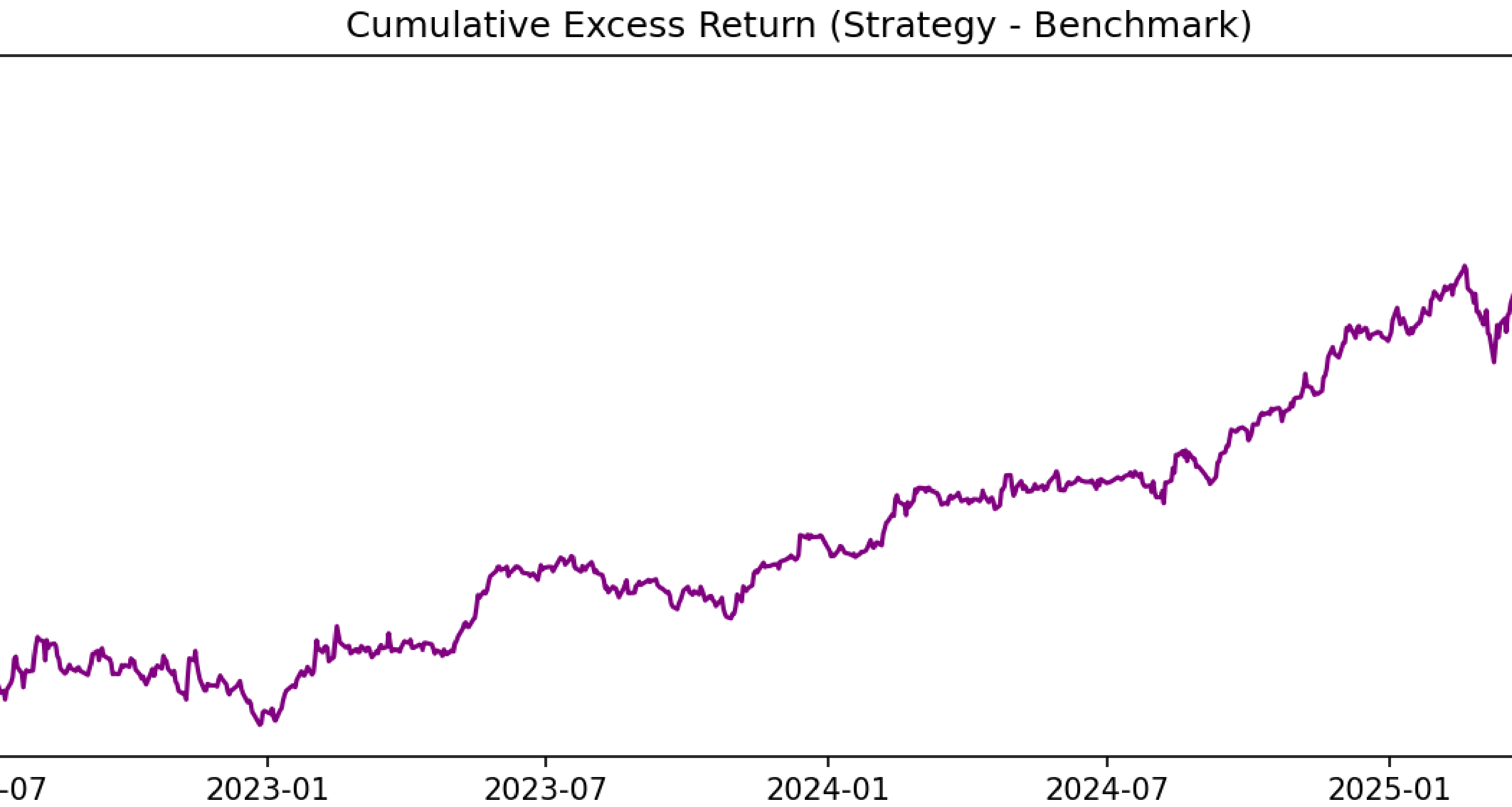
Performance Overview (Equity, Drawdown, Rolling Sharpe, Daily Returns)



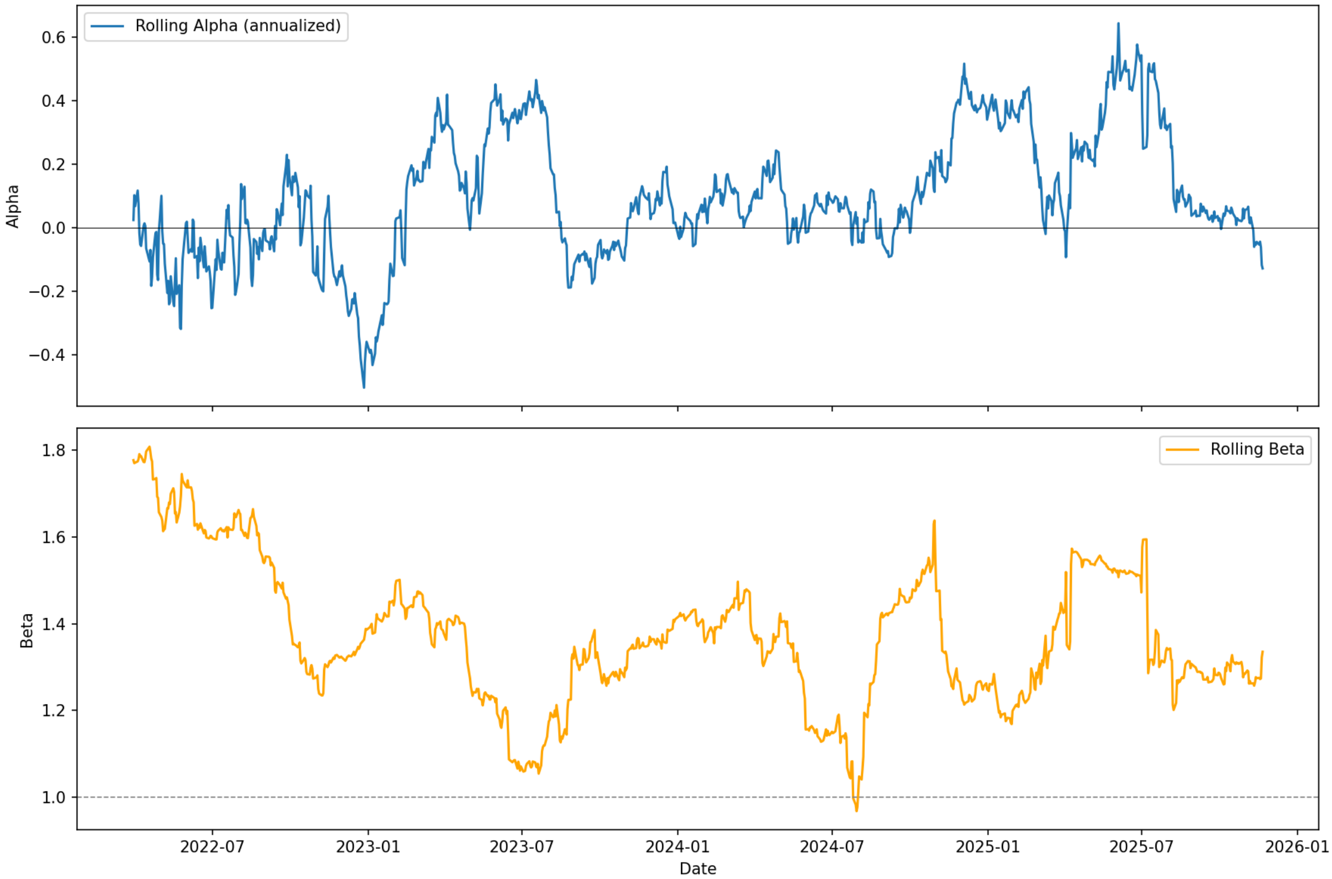
Distribution of Daily Net Returns



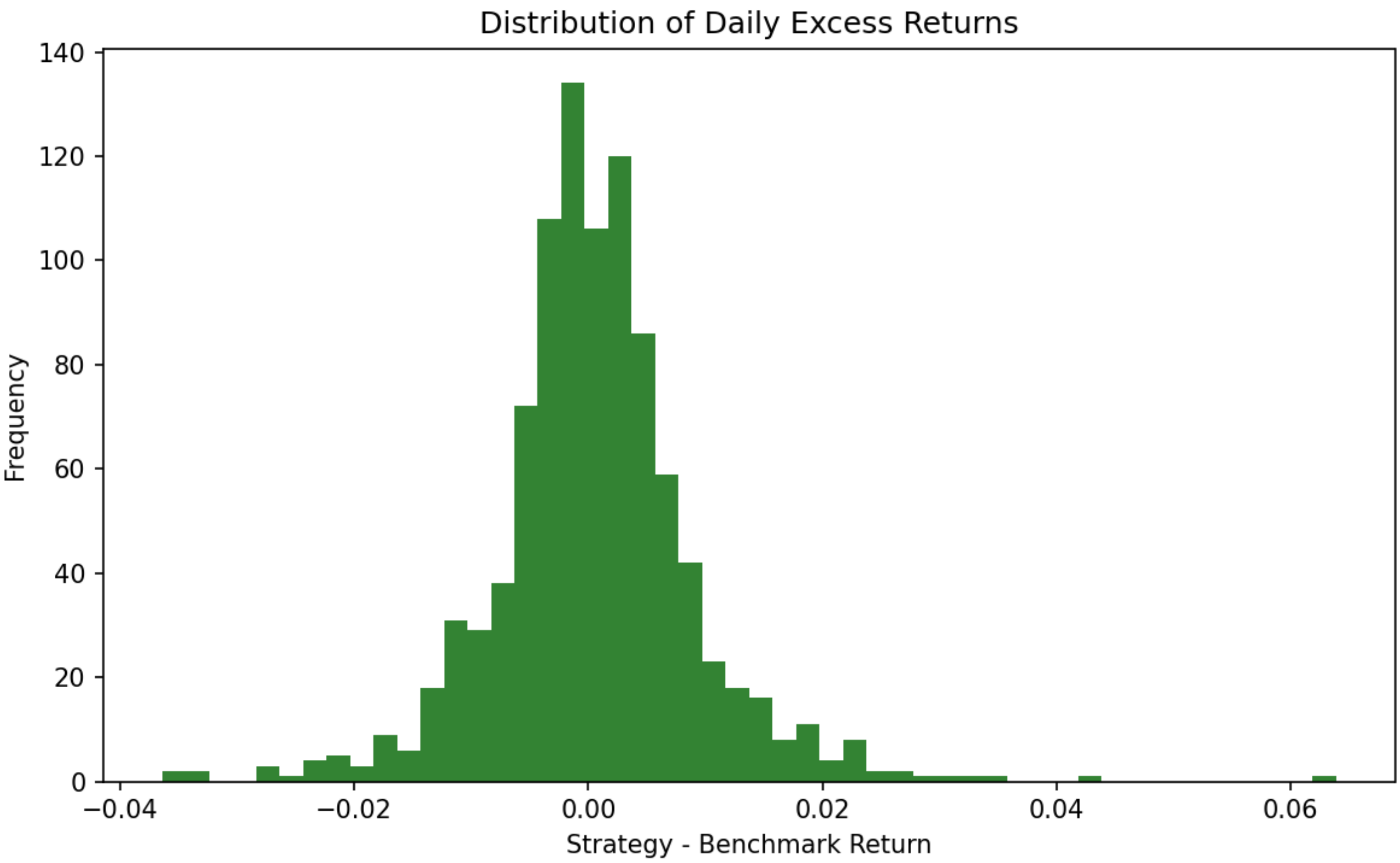
Cumulative Return vs Equal-Weight Benchmark



Cumulative Excess Return



Rolling Alpha/Beta (60d)



Excess Return Distribution

8. Factor Contribution (Top SHAP)

Alpha32: mean IC=0.009 (std=0.119, coverage=3991)
Alpha28: mean IC=-0.009 (std=0.117, coverage=3997)
Alpha19: mean IC=0.010 (std=0.158, coverage=3747)
BOP: mean IC=0.005 (std=0.186, coverage=3997)
NATR_14: mean IC=-0.009 (std=0.174, coverage=3983)

9. Monthly Performance Snapshot

2024-12-31: Strategy=-4.82%, Benchmark=-6.21%, Excess=1.53%
2025-01-31: Strategy=8.18%, Benchmark=3.73%, Excess=4.29%
2025-02-28: Strategy=-2.30%, Benchmark=-0.64%, Excess=-1.63%
2025-03-31: Strategy=-5.81%, Benchmark=-3.46%, Excess=-2.31%
2025-04-30: Strategy=2.87%, Benchmark=-1.92%, Excess=5.75%
2025-05-31: Strategy=12.59%, Benchmark=4.50%, Excess=7.84%
2025-06-30: Strategy=6.43%, Benchmark=3.42%, Excess=2.96%
2025-07-31: Strategy=1.61%, Benchmark=0.91%, Excess=0.71%
2025-08-31: Strategy=3.81%, Benchmark=2.99%, Excess=0.83%
2025-09-30: Strategy=1.17%, Benchmark=0.83%, Excess=0.36%
2025-10-31: Strategy=-0.35%, Benchmark=-1.13%, Excess=0.82%
2025-11-30: Strategy=-4.63%, Benchmark=-0.85%, Excess=-3.75%

10. Conclusion & Next Steps

Strategy assessment: after a severe 2022 drawdown, the system delivered 21% annualized return and Sharpe 0.76 in 2023-2025,
making it attractive yet risk-heavy. Excess returns are concentrated in the last two years, so alpha is regime-dependent.
Max drawdown (~31%) and beta (~1) show the portfolio still rides market swings, requiring neutralization or volatility control
before larger capital deployment.

Research rigor: the workflow follows a disciplined loop—data → factors → single-factor diagnosis → ranking model → weights →
shift-based backtest → multi-angle evaluation—with strict guards against look-ahead (feature alignment, shift(-1), caching,
IC cross-check). SHAP plus daily IC attribution provides consistent explanations, so the study is logically sound and reproducible.

Next steps:

- Risk: add beta/sector-neutral or name-cap constraints, plus volatility targeting to reduce the -31% drawdown.
- Factors: flip/remove Alpha28 and NATR_14; keep pruning inputs via ongoing single-factor runs.
- Strategy: test smaller topk or smoother n_drop, consider risk-parity / vol-scaling overlays.
- Monitoring: maintain rolling IC/alpha dashboards to detect regime shifts and trigger updates.