Comprehensive Research Methodology Report

Precog Quant Task 2026

Systematic Alpha Discovery & Algorithmic Trading Pipeline

Research Documentation

February 9, 2026

Contents

1 Executive Summary

This report documents the complete research journey through 12 experimental notebooks, chronicling the evolution from basic momentum strategies to advanced state-space (Kalman Filter) and regime detection (Hidden Markov Model) approaches.

1.1 Key Achievements

- Developed a systematic alpha discovery pipeline with 31 curated features
- Implemented walk-forward validation to prevent look-ahead bias
- Achieved **OOS Sharpe Ratio $\approx$ 2.1** with vol-targeted LGBM strategy
- Identified and fixed critical biases (SL/TP look-ahead, turnover destruction)
- Created reproducible pipeline with explicit configuration management

1.2 Research Evolution

1. **Phase 1 (NB01-02)**: ML feature engineering $\rightarrow$ turnover control
2. **Phase 2 (NB03-04)**: Statistical arbitrage $\rightarrow$ OOS evaluation
3. **Phase 3 (NB05-06)**: Regime conditioning $\rightarrow$ Sharpe maximization
4. **Phase 4 (NB07-09)**: Kalman Filter integration $\rightarrow$ hybrid strategies
5. **Phase 5 (NB10-12)**: Pipeline verification $\rightarrow$ HMM regime features

2 Data Overview

2.1 Dataset Characteristics

- **Assets**: 100 anonymized stocks with daily OHLCV data
- **Period**: 2016-01-01 to 2026-12-31 ($\approx$10 years)
- **Observations**: $\approx$2,516 trading days per asset

2.2 Train/Test Split Philosophy

Period	Dates	Purpose	Usage
Training	2016-2021	Feature eng. + model training	Walk-forward
Validation	2022-2023	Hyperparameter tuning	Cross-validation
In-Sample (IS)	2016-2023	Complete development set	Model selection
Out-of-Sample (OOS)	2024-2026	Final holdout test	**Never touched until final eval**

Table 1: Data Split Strategy

3 Feature Engineering

3.1 Feature Categories

We developed 31 curated features across 6 categories:

Category	Count	Description
Momentum	8	Trend-following signals: returns over 5, 21, 63, 126 days
Volatility	6	Risk indicators: rolling vol, ATR, vol ratios
Mean Reversion	4	Counter-trend: z-scores, Bollinger position
Cross-Sectional	4	Relative ranks within universe
Kalman Filter	5	State-space derived features (see §??)
HMM Regime	4	Regime probabilities and transitions (see §??)

Table 2: Feature Categories (31 Total)

3.2 Cross-Sectional Standardization

All features are z-scored cross-sectionally each day:

$$f^z_{i,t} = \frac{f_{i,t} - \mu_t}{\sigma_t} \tag{1}$$

where $\mu_t = \frac{1}{N}\sum_i f_{i,t}$ and $\sigma_t = \sqrt{\frac{1}{N}\sum_i (f_{i,t} - \mu_t)^2}$.

This ensures features are comparable across time while avoiding look-ahead bias.

3.3 Feature Selection via Information Coefficient

We computed the Spearman rank correlation (IC) between features and forward returns:

$$IC = \text{corr}_{\text{Spearman}}(f_{i,t}, r_{i,t+1:t+k}) \tag{2}$$

Features with $|IC| > 0.02$ and low correlation with existing features were retained.

4 Model Development

4.1 Walk-Forward Validation

To prevent look-ahead bias, we employed strict walk-forward validation:

```
# Walk-forward training (pseudocode)
for t in rebalance_dates:
    train_data = data[t - 756 days : t]  # 3-year rolling window
    model.fit(train_data)
    predictions[t] = model.predict(data[t])
```

Key parameters:

- Training window: 756 days (3 years)
- Retrain frequency: 126 days (6 months) or 21 days (monthly)

4.2 Model Architecture

4.2.1 Primary Model: LightGBM

- Objective: Regression on forward returns
- Parameters: 500 trees, max_depth=4, learning_rate=0.02
- Features: 31 z-scored features
- Feature importance: Used for feature selection refinement

4.2.2 Ensemble Strategy

Final production ensemble combines LightGBM with Ridge regression:

$$\hat{y} = 0.7 \cdot \hat{y}_{\text{LGBM}} + 0.3 \cdot \hat{y}_{\text{Ridge}} \tag{3}$$

4.3 Signal Generation

Predictions are converted to trading signals via cross-sectional z-scoring:

$$s_{i,t} = \frac{\hat{y}_{i,t} - \mu_{\hat{y},t}}{\sigma_{\hat{y},t}} \tag{4}$$

5 Turnover Control: From Gross to Net Alpha

5.1 The Turnover Problem

Critical Discovery (Notebook 02): Initial ML models generated 86-117x annual turnover, completely destroying gross alpha after transaction costs.

Strategy	Gross Sharpe	Net Sharpe	Turnover
ML Baseline (daily)	+0.59 to +1.26	-1.28 to -0.95	86-117x
ML + Smoothing (EMA 0.93)	+0.35	-0.24	12x
Simple Momentum (quarterly)	—	+0.20	∼4x
Final Ensemble (60/40)	—	**+0.42**	**6.2x**

Table 3: Turnover Impact on Performance

5.2 Turnover Reduction Techniques

1. **Position Smoothing**: EMA decay on positions

$$w_t = (1 - \alpha) \cdot w_{t-1} + \alpha \cdot w_t^{\text{target}} \tag{5}$$

2. **Rebalance Frequency**: Monthly (21 days) instead of daily
3. **Signal Smoothing**: EMA on raw signals before ranking
4. **Top-N Portfolio**: Fixed number of positions (reduces turnover from rank changes)

5.3 Key Insight

"Gross Sharpe is meaningless without turnover analysis."

Simple factor signals (momentum + mean reversion) proved more robust than complex ML models after transaction costs.

6 Statistical Arbitrage Analysis

6.1 Cointegration Testing

We tested all $\binom{100}{2} = 4,950$ asset pairs for cointegration using the Engle-Granger two-step method:

1. Regress $\log(P_A)$ on $\log(P_B)$ to get spread: $s_t = \log(P_{A,t}) - \beta \cdot \log(P_{B,t})$
2. Test spread for stationarity via ADF test

6.2 Multiple Testing Correction

With 4,950 tests at $\alpha = 0.05$, we expected ≈248 false positives. We applied:

- **Bonferroni**: $\alpha_{\text{adj}} = 0.05/4950 = 1.01 \times 10^{-5}$ (conservative)
- **FDR (Benjamini-Hochberg)**: Controls false discovery rate at 5% (more powerful)

6.3 Tradability Filter

Pairs were filtered by half-life of mean reversion:

$$\text{Half-life} = \frac{-\ln(2)}{\ln(\phi)} \tag{6}$$

where ϕ is the AR(1) coefficient of the spread. Tradable pairs have half-life in 2-60 days.

6.4 Lead-Lag Analysis

We computed cross-correlations at various lags to identify predictive relationships between assets.

7 Kalman Filter Implementation

For detailed mathematical derivation and implementation, see the companion document: ***kalman_filter_techn***

7.1 Overview

The Kalman Filter provides a principled way to:

- Estimate the "true" underlying state from noisy observations
- Quantify estimation uncertainty
- Generate smooth signals for trading

7.2 State-Space Model

We use the Local Level Model:

$$x_t = x_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q) \qquad \text{(State equation)} \tag{7}$$

$$y_t = x_t + v_t, \quad v_t \sim \mathcal{N}(0, R) \qquad \text{(Observation equation)} \tag{8}$$

7.3 Parameter Estimation via EM Algorithm

The process noise Q and observation noise R are estimated using Expectation-Maximization on in-sample data only.

7.4 Critical Separation: IS vs OOS

Component	In-Sample	Out-of-Sample
Parameter Estimation (Q, R)	✓	×
RTS Smoother (oracle labels)	✓	×(Look-ahead!)
Forward Filter (signals)	✓	✓

Table 4: Kalman Filter Usage Rules

7.5 Kalman-Derived Features

- `kf_innovation`: Prediction error $y_t - \hat{y}_{t|t-1}$
- `kf_uncertainty`: State covariance P_t
- `kf_gain`: Kalman gain K_t
- `kf_state_price_gap`: Filtered state vs observed price
- `kf_likelihood_ratio`: Innovation normalized by variance

8 HMM Regime Detection

For detailed mathematical derivation and implementation, see the companion document: ***hmm_ regime_ techn***

8.1 Motivation

Hidden Markov Models identify latent market regimes (e.g., trending vs. mean-reverting, high vs. low volatility) without explicit labeling.

8.2 HMM as Feature Generator

We use HMM **not** for direct return prediction, but as a regime filter:

- Extract $P(\text{regime} = k|\text{data})$ as features
- Condition trading signals on regime state
- Adjust position sizing by regime

8.3 Training Protocol

1. Fit HMM on in-sample data only
2. Freeze parameters for out-of-sample use
3. Use forward algorithm for real-time regime inference

8.4 HMM-Derived Features

- `hmm_state`: Most likely regime (0, 1, 2, ...)
- `hmm_prob_high`: Probability of high-volatility regime
- `hmm_transition`: Recent regime transition indicator
- `hmm_stability`: Regime persistence measure

9 Backtesting Framework

9.1 Backtest Configuration

```
FINAL_STRATEGY = {
    'signal': 'pred_lgbm_zscore',
    'top_pct': 0.10,            # Top 10% of signals
    'rebalance_freq': 21,        # Monthly
    'target_vol': 0.15,          # 15% annualized volatility
    'max_leverage': 2.0,         # Cap at 2x
    'tc_bps': 10                 # 10 basis points per trade
}
```

9.2 Vol-Targeting Implementation

Leverage is adjusted to target constant portfolio volatility:

$$\text{leverage}_t = \min\left(\frac{\sigma_{\text{target}}}{\hat{\sigma}_t}, \text{max\_leverage}\right) \tag{9}$$

where $\hat{\sigma}_t$ is the 63-day rolling volatility.

9.3 Transaction Cost Model

$$\text{TC}_t = \sum_i |w_{i,t} - w_{i,t-1}| \times \text{tc\_bps}/10000 \tag{10}$$

9.4 Stop-Loss/Take-Profit (CRITICAL FIX)

Original Bug: SL/TP was checked *after* adding day's return (look-ahead bias).

Corrected Implementation: Check triggers at market OPEN using yesterday's cumulative:

```
# CORRECT: Check SL/TP at market OPEN (before today's return)
cum_before_today = prev_cum  # Yesterday's cumulative
if cum_before_today <= -stop_loss or cum_before_today >= take_profit:
    position_closed = True
    final_return = cum_before_today  # Exit at open
else:
    # Hold through day
    final_return = cum_before_today + today_return
```

10 Final Strategy Performance

10.1 Strategy: LGBM Top 10% + Vol-Targeting 15%

Metric	In-Sample (2016-2023)	Out-of-Sample (2024-2026)
Sharpe Ratio	~2.5	~2.1
Annual Return	—	—
Annual Volatility	~15% (targeted)	~15%
Max Drawdown	—	—
Trading Days	2,012	513

Table 5: Final Strategy Metrics

10.2 Sharpe Decay Analysis

$$\text{Sharpe Decay} = \left(1 - \frac{\text{Sharpe}_{\text{OOS}}}{\text{Sharpe}_{\text{IS}}}\right) \times 100\% \tag{11}$$

An OOS Sharpe of 2.1 vs IS Sharpe of 2.5 represents reasonable decay, suggesting the model generalizes rather than overfits.

11 Mistakes Made & Lessons Learned

11.1 Critical Mistakes Identified

1. **Turnover Destruction (NB02)**
   - ML models generated 86-117x turnover, destroying all alpha
   - *Lesson*: Always analyze net (after TC) performance
2. **SL/TP Look-Ahead Bias (NB/Stage4)**
   - Original implementation checked SL/TP after adding day's return
   - *Lesson*: Risk management triggers must use EOD-1 state
3. **Multiple Testing Inflation (NB03)**
   - Without correction, 248+ spurious cointegration pairs appeared
   - *Lesson*: Apply Bonferroni or FDR correction for large-scale testing
4. **RTS Smoother OOS Usage (NB07)**
   - RTS Smoother uses future data for backward pass
   - *Lesson*: RTS is IS-only; use forward filter for OOS
5. **HMM State Label Flipping (NB11)**
   - State labels can flip between training runs
   - *Lesson*: Use consistent ordering convention (e.g., by volatility)

11.2 Key Insights

1. **Simplicity Often Wins**: Simple momentum beat complex ML after costs
2. **Regime Shifts Matter**: OOS degradation often due to regime change, not model failure
3. **Kalman Value**: Structures uncertainty, not prediction; features more valuable than raw estimates
4. **HMM Value**: Improves stability, not magnitude; use as filter, not predictor
5. **Documentation**: Explicit CONFIG dict prevents hidden parameter dependencies

12 Reproducibility

12.1 Saved Artifacts

- `outputs/final_strategy/lgbm_predictions.parquet`
- `outputs/final_strategy/backtest_is_results.parquet`
- `outputs/final_strategy/backtest_oos_results.parquet`
- `outputs/final_strategy/strategy_summary.json`

12.2 Configuration Management

All parameters are explicitly documented in the `FINAL_STRATEGY` dictionary.

13 Conclusion

This research journey demonstrates the iterative nature of quantitative research:

- Started with complex ML approaches
- Discovered turnover destroyed alpha
- Pivoted to simple factors with careful cost management
- Integrated advanced methods (Kalman, HMM) as feature generators
- Achieved robust OOS performance with Sharpe ≈ 2.1

Final Strategy: LGBM Top 10% + Vol-Targeting 15% with monthly rebalancing.

"In quantitative research, the journey of discovering what doesn't work is as valuable as finding what does."

14 References

1. Hamilton, J.D. (1994). *Time Series Analysis*. Princeton University Press.
2. Durbin, J. & Koopman, S.J. (2012). *Time Series Analysis by State Space Methods*. Oxford University Press.
3. Frazzini, A., Israel, R., & Moskowitz, T.J. (2018). Trading Costs. *Journal of Finance*.
4. de Prado, M.L. (2018). *Advances in Financial Machine Learning*. Wiley.