

## Peer-Review

Singh, Aayush. 2026. "Beyond Pooling: The Role of Correlation Structures for Contemporaneous and Lagged Cross-Asset Stock Prediction." *Journal of High School Science* 10 (3): 1–25.  
<https://doi.org/10.64336/001c.164554>.

This manuscript will require significant revision. Please address all my concerns below.

1. Your price-level target construction almost certainly suffers from hard leakage. Your feature vector at time  $t$  includes:  $\text{Close}_t$ ,  $\text{EMA}_t$ ,  $\text{SMA}_t$ ,  $\text{MACD}_t$  (all functions of  $\text{Close}_t$ ). your model is hence an exact proxy for the target. This is not learning, it is target persistence. You will need to correct for this and rerun the model.
2. You probably have Feature-target leakage via rolling indicators such as  $\text{SMA}_{20}$ ,  $\text{EMA}_{12}$ ,  $\text{EMA}_{26}$ ,  $\text{Volatility}_{20D}$ ,  $\text{Momentum}_{10D}$ ,  $\text{Mean\_Reversion}_{20D}$ . If rolling windows are computed before train/test splitting (as described), then: Rolling windows at the start of the test set contain information from immediately preceding days, which were part of the training set. This creates boundary leakage. You need to address this and rerun the model.
3. It is very likely that you have market feature leakage (SPY). Unless explicitly lagged, SPY return on day  $t$  encodes market movement that explains most stock movement. If aligned incorrectly, the model sees market response contemporaneous with or after the stock move. The manuscript never states:  $\text{Lag} = 1$ , Or  $\text{shift}(1)$ . Given pandas defaults, this is a probable leakage point.
4. If  $\text{StandardScaler.fit}()$  is applied to the full pooled dataset before leave-one-ticker-out testing, then the test stock's distribution influences the training transformation. This is hard leakage in pooled linear models.
5. You have cross-asset temporal leakage. Example: NVDA data ends in 2024, JNJ data ends in 2022. When pooled: Model trains on 2023–2024 NVDA data, then tests on 2020 JNJ data. This leaks: Future macro regimes, Post-COVID volatility structure, Rate-hike periods. This inflates pooled-asset performance even without direct price leakage.
6. Target leakage through ticker encoding: If ticker encoding is: One-hot, Or ordinal, The model can: Memorize asset-specific behavior, especially in tree models. While leave-one-ticker-out helps, partial leakage remains if encoding was fit globally.
7. You have evaluation bias due to ONE instance of 80:20 split. This could be just one lucky regime split. You have no variance estimation and no robustness checks. You will need to replace the single split with a minimum 3-fold (1, 2 and 3) walk-forward. Also either perform subperiod tests, or bootstrap intervals.
8. Your ablation study is not really an ablation study; rather, it is a comparison between sector specific and pooled. Nothing is ablated, you are changing datasets; not removing components. Your experiment supports correlational inference, not causal attribution. Please perform a true ablation study (or studies).
9. You have severe multicollinearity by any econometric standard. Your features (regressors) are highly correlated and code redundant information. This means, that, even with ridge regression, Coefficients become unstable, Small data changes  $\rightarrow$  large coefficient swings, Signs can flip without changing predictions. Hence, Reported coefficient interpretations (implicit or explicit) are unreliable. The optimal alpha "stock dependence" result may be an artifact. Pooled vs single comparisons may reflect regularization behavior (from numerical stability, not signal discovery) not learning strategy. Predictive  $R^2$  may look stable, but model meaning is broken. Perform a proper VIF analysis, present a heatmap, perform dimensionality reduction (PCA or orthogonalized feature reduction) and re-run the model with un-correlated features.
10. All the above points are magnified because you only have 27 stocks, hence only 27 cross-sectional degrees of freedom (even if you had multiple years of data). This inflates confidence, exacerbates collinearity, pooled versus single difference. You cannot claim superiority of pooled, you cannot claim generalizability. Your study then becomes exploratory and pedagogical and overlaps with copious literature in the field already published. This confirms known results on a small universe using standard models and therefore does not add any significant value.

---

**Comment One:** Your price-level target construction almost certainly suffers from hard leakage. Your feature vector at time  $t$  includes:  $\text{Close}_t$ ,  $\text{EMA}_t$ ,  $\text{SMA}_t$ ,  $\text{MACD}_t$  (all functions of  $\text{Close}_t$ ). your model is hence an exact proxy for the target. This is not learning, it is target persistence. You will need to correct for this and rerun the model.

$\text{Close}_t$ ,  $\text{EMA}_t$ ,  $\text{SMA}_t$ , and  $\text{MACD}_t$  were all removed from the feature set. All features in the final model are computed as ratios, percentage changes, or rolling statistics rather than raw price levels, ensuring no raw close value appears in the feature list.

---

**Comment Two:** You probably have Feature-target leakage via rolling indicators such as  $\text{SMA}_{20}$ ,  $\text{EMA}_{12}$ ,  $\text{EMA}_{26}$ ,  $\text{Volatility}_{20D}$ ,  $\text{Momentum}_{10D}$ ,  $\text{Mean\_Reversion}_{20D}$ . If rolling windows are computed before train/test splitting (as described), then: Rolling windows at the start of the test set contain information from immediately preceding days, which were part of the training set. This creates boundary leakage. You need to address this and rerun the model.

Rolling window features were computed prior to splitting, which introduces minor boundary leakage at fold edges. This is a known and accepted tradeoff in time-series ML. As a sensitivity check, rolling windows were recomputed strictly within each fold using a 60-day warmup window from the training partition. Method A (global feature engineering, as used in the paper) yielded  $R^2$  of -4.13% and Method B (strict per-fold computation) yielded -3.95%, a difference of 0.0018% percentage points, confirming that boundary leakage does not materially inflate the reported results.

---

**Comment Three:** It is very likely that you have market feature leakage (SPY). Unless explicitly lagged, SPY return on day  $t$  encodes market movement that explains most stock movement. If aligned incorrectly, the model sees market response contemporaneous with or after the stock move. The manuscript never states:  $\text{Lag} = 1$ , Or  $\text{shift}(1)$ . Given pandas defaults, this is a probable leakage point.

SPY\_Return and SPY\_Volatility are now explicitly lagged by one trading day using  $\text{shift}(1)$  before joining to stock data, ensuring the model only sees market context available before the trading day being predicted. This is documented in the Methods section and implemented in the code below:

```
spy_returns = spy['Close'].pct_change().rename('SPY_Return').shift(1)
```

```
spy_volatility = spy['Close'].pct_change().rolling(20).std().shift(1).rename('SPY_Volatility')
```

---

**Comment Four:** "If `StandardScaler.fit()` is applied to the full pooled dataset before leave-one-ticker-out testing, then the test stock's distribution influences the training transformation. This is hard leakage in pooled linear models."

`StandardScaler` was already fit exclusively on the training partition of each fold and applied to the test

partition using training derived parameters only. This is now explicitly stated in the Methods section. The test stock's distribution does not influence the scaling transformation.

---

**Comment Five:** "You have cross-asset temporal leakage. Example: NVDA data ends in 2024, JNJ data ends in 2022. When pooled: Model trains on 2023–2024 NVDA data, then tests on 2020 JNJ data. This leaks: Future macro regimes, Post-COVID volatility structure, Rate-hike periods. This inflates pooled-asset performance even without direct price leakage."

All datasets are now trimmed to the earliest common end date across all tickers, ensuring no stock contributes data from time periods unavailable to others in the pooled training set. This is documented in the Methods section and implemented in the code below:

```
last_indices = [df.index[-1] for df in all_dataframes] earliest_last_date = min(last_indices)
all_dataframes = [df[df.index <= earliest_last_date] for df in all_dataframes]
```

---

**Comment Six:** "Target leakage through ticker encoding: If ticker encoding is: One-hot, Or ordinal, The model can: Memorize asset-specific behavior, especially in tree models. While leave-one-ticker-out helps, partial leakage remains if encoding was fit globally." Ticker encoding was not used as a feature in the final model. The leave-one-ticker-out methodology ensures the target stock's data is entirely held out during training, preventing the model from memorizing asset-specific behavior through encoding.

---

**Comment Seven:** "You have evaluation bias due to ONE instance of 80:20 split. This could be just one lucky regime split. You have no variance estimation and no robustness checks. You will need to replace the single split with a minimum 3-fold (1,2 and 3) walk-forward. Also either perform subperiod tests, or bootstrap intervals."

Data was split using a three-fold walk-forward TimeSeriesSplit to maintain temporal order. Models were trained and evaluated on the final fold, preserving the largest possible training window while ensuring the test period reflects the most recent available data. This approach exposes the model to different temporal regimes through the fold structure while avoiding the regime sensitivity of a single arbitrary split.

---

**Comment Eight:** "Your ablation study is not really an ablation study; rather, it is a comparison between sector specific and pooled. Nothing is ablated, you are changing datasets; not removing components. Your experiment supports correlational inference, not causal attribution. Please perform a true ablation study"

(or studies)."

The experiment described as a feature ablation study now tests the contribution of individual feature groups to model performance by removing them and retraining, then comparing results. The other experiment comparing cross-sector versus sector-specific training data is more precisely a data composition comparison rather than a feature ablation. The paper now explicitly acknowledges this distinction in the Methods section.

---

**Comment Nine:** "You have severe multicollinearity by any econometric standard. Your features (regressors) are highly correlated and code redundant information. This means, that, even with ridge regression, Coefficients become unstable, Small data changes → large coefficient swings, Signs can flip without changing predictions. Hence, Reported coefficient interpretations (implicit or explicit) are unreliable. The optimal alpha "stock dependence" result may be an artifact. Pooled vs single comparisons may reflect regularization behavior (from numerical stability, not signal discovery) not learning strategy. Predictive  $R^2$  may look stable, but model meaning is broken. Perform a proper VIF analysis, present a heatmap, perform dimensionality reduction (PCA or orthogonalized feature reduction) and re-run the model with un-correlated features." VIF analysis was conducted on all 18 features prior to model training, documented in Figure 1.1. Three intraday price features showed extreme multicollinearity with VIF scores exceeding 14,000. PCA was applied to reduce the feature space to 12 orthogonal principal components preserving 95% of variance, directly resolving the multicollinearity concern. The PCA loadings matrix is presented in Figure 5.2, showing how the features were broken down into interpretable independent components.

---

**Comment Ten:** All the above points are magnified because you only have 27 stocks, hence only 27 cross-sectional degrees of freedom (even if you had multiple years of data). This inflates confidence, exacerbates collinearity, pooled versus single difference. You cannot claim superiority of pooled, you cannot claim generalizability. Your study then becomes exploratory and pedagogical and overlaps with copious literature in the field already published. This confirms known results on a small universe using standard models and therefore does not add any significant value."

The 26-stock dataset is acknowledged as a limitation, and the paper frames conclusions as exploratory rather than claiming broad generalizability. That said, the contribution here is not a large-scale cross-sectional analysis, but instead a controlled comparison of single-asset versus pooled-asset training strategies across both price and return prediction, with a sector composition test included. Prior pooled ML research has largely focused on cross-sectional return prediction across thousands of stocks. This study asks a different question, whether pooling a small curated set of individual stocks improves time-series prediction for each specific ticker. That comparison, with consistent model architectures and controlled hyperparameters, is not replicated in the cited literature. The small dataset is a real limitation and is documented as such. It does not eliminate the methodological contribution.

---

Thank you for partiall addressing my comments. However, you still fail to demonstrate that pooled > single is discovered empirially and not guaranteed by design. You have not adequately addressed Point 10, instead just accepting my concern as a limitation. This means that your contribution (as stated in the current title) is negligible. Pooled vs single is

almost guaranteed to favor pooled because it has more data and implicit regularization, hence the main result risks being structurally predetermined rather than discovered. This result (that pooled models outperform narrower training sets) is also well-established in the literature and is expected under basic statistical learning principles.

What I find mildly interesting about your paper is that cross sector > sector specific (although effect sizes are small and you do not provide statistical validation). I would imagine that if you did, you would not find this statistically significant UNLESS you define the degree of cross-sector; i.e. highly/weakly/anti-correlated. Compute a correlation matrix of sector returns. Compute highly/weakly/anti-correlated stocks/sectors. Define sector regimes: i.e. same sector training, correlated sector training, anti-correlated sector training, mixed (baseline, pooled) training. Control for sample size across all regimes. Evaluate performance differences, then bootstrap confidence intervals and test significance. If anti-correlated > same sector then Models benefit from diverse / orthogonal signals, which would suggest that pooling works due to information diversity, not just size. If same sector > anti-correlated, then domain similarity > diversity, which contradicts naive pooling assumptions.

Instead of framing the result as:

“pooled models outperform single-asset models,”

the manuscript should pivot toward a more mechanistic and interesting question that remains under-researched in the published literature.

“The effect of cross-asset training depends on the correlation structure of the training universe; training on less correlated or oppositely moving sectors may provide more informative signals than sector-homogeneous data.” Your title would change to : “Beyond Pooling: The Role of Inter-Sector Correlation Structure in Cross-Asset Stock Prediction Models” or something similar. This reframing would allow the study to test whether performance gains arise from data diversity (orthogonal signal structure) rather than simply increased sample size.

Method caution:

However, True negative correlation between sectors is rare. Most sectors are positively correlated just with different betas. Hence, for anti-correlation, you may need to use residual returns (market neutralized) or subtract SPY/factor exposure.

---

### Comment 1:

**"You still fail to demonstrate that pooled > single is discovered empirically and not guaranteed by design. Pooled vs single is almost guaranteed to favor pooled because it has more data and implicit regularization, hence the main result risks being structurally predetermined rather than discovered."**

Response:

The pooled vs. single comparison is completely eliminated as the central finding. The paper now controls for sample size across all regimes, and they are all subsampled to the same size as the smallest regime for each target stock (typically 14 stocks, described in the Methodology section under "Correlation Matrix and Regime Construction"). Under this experiment design, the question is whether the composition of a fixed-size training set affects performance. The title, abstract, and framing have all been updated to reflect this pivot.

### Comment 2:

**"What I find mildly interesting about your paper is that cross sector > sector specific (although effect sizes are small and you do not provide statistical validation). I would**

**imagine that if you did, you would not find this statistically significant UNLESS you define the degree of cross-sector; i.e. highly/weakly/anti-correlated. Compute a correlation matrix of sector returns. Compute highly/weakly/anti-correlated stocks/sectors. Define sector regimes: i.e. same sector training, correlated sector training, anti-correlated sector training, mixed (baseline, pooled) training. Control for sample size across all regimes. Evaluate performance differences, then bootstrap confidence intervals and test significance."**

Response:

All of this was implemented. We compute a full  $75 \times 75$  pairwise Pearson correlation matrix on market-neutralized residual returns (Figure 8.1). Per-target quartile thresholds define four regimes: high-correlation (top 25%), medium-correlation (middle 50%), anti-correlation (bottom 25%), and same-sector, with a size-matched pooled baseline (Methodology, "Correlation Matrix and Regime Construction"). Sample size is held constant across all five regimes by subsampling to the minimum regime size per target stock. BCa bootstrap confidence intervals with 1,000 resamples are reported for both mean  $R^2$  per regime (Figure 6.1) and pairwise  $R^2$  differences between regimes (Figure 7.1). High-correlation training outperforms anti-correlation (95% CI: [0.0002, 0.0018]) and medium-correlation training (95% CI: [0.0015, 0.0043]), while being indistinguishable from same-sector training (95% CI: [-0.001, 0.0001]).

### **Reviewer Comment 3:**

**"Instead of framing the result as 'pooled models outperform single-asset models,' the manuscript should pivot toward a more mechanistic and interesting question... 'The effect of cross-asset training depends on the correlation structure of the training universe.' Your title would change to: 'Beyond Pooling: The Role of Inter-Sector Correlation Structure in Cross-Asset Stock Prediction Models.'"**

Response:

The manuscript has been fully reframed around this question. The title was changed to exactly what the reviewer suggested. The abstract, introduction, and conclusions now center on the said topic.

**Reviewer Comment 4: "True negative correlation between sectors is rare. Most sectors are positively correlated just with different betas. Hence, for anti-correlation, you may need to use residual returns (market neutralized) or subtract SPY/factor exposure."**

Response:

Residual returns are used throughout the entire paper. Each stock's daily returns are regressed against SPY via OLS and the residuals are extracted before any correlation computation, regime assignment, feature construction, or model training (Methodology, "Market Neutralization via OLS Residuals"). The paper explicitly acknowledges the reviewer's caution in the Limitations section, noting that the anti-correlation regime consists of stocks with small-magnitude negative

residual correlations (25th percentile ranging from approximately -0.003 to -0.12 across target stocks), and that true strong economic anti-correlation is rare among large-cap S&P 500 equities.

---

Thank you for addressing my comments. Nicely done. The paper is greatly improved from its original iteration. However, I would like to see one last deficiency addressed. Please address my concern listed below:

The revised manuscript substantially improves upon the original version and now asks a better-controlled question. However, the training regimes are still defined using contemporaneous residual correlations, whereas the prediction task itself is explicitly a lead-lag forecasting problem. As a result, the manuscript tests whether same-day co-movement is associated with future predictive performance, rather than directly testing whether lead-lag relationships determine predictive performance.

Because the residual-return dataset, walk-forward framework, and regime-construction pipeline have already been implemented, I believe an additional analysis is feasible within the current study. Specifically, for each target stock, compute lagged predictive correlations  $\text{Corr}(\epsilon_i(t), \epsilon_{\text{target}}(t+1))$  using only the training portion of the data. Use these values to define high-predictive, medium-predictive, and low- or negatively-predictive training universes, then evaluate model performance on held-out data using the existing pipeline.

This modification would require minimal changes to the experimental framework while providing a more direct test of the paper's central mechanism. If stocks selected by historical predictive correlation substantially outperform stocks selected by contemporaneous correlation, this would strengthen the interpretation that lead-lag information transfer drives the observed effects. Conversely, if the results remain similarly weak, the manuscript would provide stronger evidence that training-universe composition has little influence on cross-asset prediction for large-cap equities. Either outcome would be scientifically informative and would considerably strengthen the paper.

Specifically, for each target stock, compute lagged predictive correlations of the form  $\text{Corr}(\epsilon_i(t), \epsilon_{\text{target}}(t+1))$ . However, it is essential that these predictive correlations be estimated exclusively within the training data and never using observations from the held-out evaluation period.

Otherwise, the analysis becomes circular: stocks would be selected because they already appear predictive, and the subsequent finding that they are predictive would largely be a consequence of the selection procedure itself.

A rigorous implementation would therefore proceed as follows:

Split the data into a training period and a completely held-out test period.

Using only the training period, compute predictive correlations  $\text{Corr}(\epsilon_i(t), \epsilon_{\text{target}}(t+1))$  between each candidate stock and the target.

Rank stocks according to these predictive correlations and construct high-predictive, medium-predictive, and low- or negatively-predictive training universes.

Train models using the existing pipeline.

Evaluate performance exclusively on the held-out test period that played no role in regime assignment.

An even stronger implementation would estimate predictive correlations within each walk-forward training fold, reassign regimes dynamically, and evaluate only on the subsequent fold. This would completely eliminate any possibility of information leakage.

---

#### **Comment 1:**

**"As a result, the manuscript tests whether same-day co-movement is associated with future predictive performance, rather than directly testing whether lead-lag relationships determine predictive performance."**

Response:

We now directly test whether lead-lag relationships determine predictive performance. A second pipeline was added in which training regimes are defined by one-day-ahead predictive correlation,  $\text{Corr}(\varepsilon_{i,t}, \varepsilon_{\text{large}}(t+1))$ , estimated exclusively within each training fold. Results are reported in Figures 8.1 through 14.1 and interpreted in the Discussion under "Lagged Correlation Structure is Significant but Spurious."

#### Comment 2:

**"This modification would require minimal changes to the experimental framework while providing a more direct test of the paper's central mechanism. If stocks selected by historical predictive correlation substantially outperform stocks selected by contemporaneous correlation, this would strengthen the interpretation that lead-lag information transfer drives the observed effects. Conversely, if the results remain similarly weak, the manuscript would provide stronger evidence that training-universe composition has little influence on cross-asset prediction for large-cap equities. Either outcome would be scientifically informative and would considerably strengthen the paper."**

Response:

We found that training-universe composition has little influence on cross-asset prediction for large-cap equities. The lagged pipeline does produce a statistically significant medium-regime advantage, but we interpret this as in-sample data snooping rather than genuine lead-lag structure, since the predictive correlations estimated within training folds fail to generalize out of sample. All regimes across both pipelines produce near-zero or negative  $R^2$ , consistent with the Efficient Market Hypothesis.

#### Reviewer Comment 3:

**"An even stronger implementation would estimate predictive correlations within each walk-forward training fold, reassign regimes dynamically, and evaluate only on the subsequent fold. This would completely eliminate any possibility of information leakage."**

Response:

The stronger implementation was adopted. Predictive correlations are now estimated exclusively within each walk-forward training fold, regimes are reassigned dynamically per fold, and evaluation occurs only on the subsequent held-out fold. This design is described in the Lagged Pipeline section of the Methodology and eliminates any possibility of information leakage between training and test periods.

---

Please include a separate file describing my earlier review comments verbatim and describing where and how in the manuscript each comment was addressed.

---

**Round Three -**

### Comment 1:

**"As a result, the manuscript tests whether same-day co-movement is associated with future predictive performance, rather than directly testing whether lead-lag relationships determine predictive performance."**

Response:

We now directly test whether lead-lag relationships determine predictive performance. A second pipeline was added in which training regimes are defined by one-day-ahead predictive correlation,  $\text{Corr}(\varepsilon_{i,t}, \varepsilon_{\text{arge}}(t+1))$ , estimated exclusively within each training fold. Results are reported in Figures 8.1 through 14.1 and interpreted in the Discussion under "Lagged Correlation Structure is Significant but Spurious."

### Comment 2:

**"This modification would require minimal changes to the experimental framework while providing a more direct test of the paper's central mechanism. If stocks selected by historical predictive correlation substantially outperform stocks selected by contemporaneous correlation, this would strengthen the interpretation that lead-lag information transfer drives the observed effects. Conversely, if the results remain similarly weak, the manuscript would provide stronger evidence that training-universe composition has little influence on cross-asset prediction for large-cap equities. Either outcome would be scientifically informative and would considerably strengthen the paper."**

Response:

We found that training-universe composition has little influence on cross-asset prediction for large-cap equities. The lagged pipeline does produce a statistically significant medium-regime advantage, but we interpret this as in-sample data snooping rather than genuine lead-lag structure, since the predictive correlations estimated within training folds fail to generalize out of sample. All regimes across both pipelines produce near-zero or negative  $R^2$ , consistent with the Efficient Market Hypothesis.

**Comment 3: "An even stronger implementation would estimate predictive correlations within each walk-forward training fold, reassign regimes dynamically, and evaluate only on the subsequent fold. This would completely eliminate any possibility of information leakage."**

Response:

The stronger implementation was adopted. Predictive correlations are now estimated exclusively within each walk-forward training fold, regimes are reassigned dynamically per fold, and evaluation occurs only on the subsequent held-out fold. This design is described in the Lagged Pipeline section of the Methodology and eliminates any possibility of information leakage between training and test periods.

### Round Two - Comment 1:

**"You still fail to demonstrate that pooled > single is discovered empirically and not guaranteed by design. Pooled vs single is almost guaranteed to favor pooled because it has more data and implicit regularization, hence the main result risks being structurally predetermined rather than discovered."**

Response:

The pooled vs. single comparison is completely eliminated as the central finding. The paper now controls for sample size across all regimes, and they are all subsampled to the same size as the smallest regime for each target stock (typically 14 stocks, described in the Methodology section under "Correlation Matrix and Regime Construction"). Under this experiment design, the question is whether the composition of a fixed-size training set affects performance. The title, abstract, and framing have all been updated to reflect this pivot.

## **Comment 2:**

**"What I find mildly interesting about your paper is that cross sector > sector specific (although effect sizes are small and you do not provide statistical validation). I would imagine that if you did, you would not find this statistically significant UNLESS you define the degree of cross-sector; i.e. highly/weakly/anti-correlated. Compute a correlation matrix of sector returns. Compute highly/weakly/anti-correlated stocks/sectors. Define sector regimes: i.e. same sector training, correlated sector training, anti-correlated sector training, mixed (baseline, pooled) training. Control for sample size across all regimes. Evaluate performance differences, then bootstrap confidence intervals and test significance."**

Response:

All of this was implemented. We compute a full  $75 \times 75$  pairwise Pearson correlation matrix on market-neutralized residual returns (Figure 8.1). Per-target quartile thresholds define four regimes: high-correlation (top 25%), medium-correlation (middle 50%), anti-correlation (bottom 25%), and same-sector, with a size-matched pooled baseline (Methodology, "Correlation Matrix and Regime Construction"). Sample size is held constant across all five regimes by subsampling to the minimum regime size per target stock. BCa bootstrap confidence intervals with 1,000 resamples are reported for both mean  $R^2$  per regime (Figure 6.1) and pairwise  $R^2$  differences between regimes (Figure 7.1). High-correlation training outperforms anti-correlation (95% CI: [0.0002, 0.0018]) and medium-correlation training (95% CI: [0.0015, 0.0043]), while being indistinguishable from same-sector training (95% CI: [-0.001, 0.0001]).

## **Comment 3:**

**"Instead of framing the result as 'pooled models outperform single-asset models,' the manuscript should pivot toward a more mechanistic and interesting question... 'The effect of cross-asset training depends on the correlation structure of the training universe.' Your title would change to: 'Beyond Pooling: The Role of Inter-Sector Correlation Structure in Cross-Asset Stock Prediction Models.'"**

Response:

The manuscript has been fully reframed around this question. The title was changed to exactly what the reviewer suggested. The abstract, introduction, and conclusions now center on the said topic.

#### **Comment 4:**

**"True negative correlation between sectors is rare. Most sectors are positively correlated just with different betas. Hence, for anti-correlation, you may need to use residual returns (market neutralized) or subtract SPY/factor exposure."**

Response:

Residual returns are used throughout the entire paper. Each stock's daily returns are regressed against SPY via OLS and the residuals are extracted before any correlation computation, regime assignment, feature construction, or model training (Methodology, "Market Neutralization via OLS Residuals"). The paper explicitly acknowledges the reviewer's caution in the Limitations section, noting that the anti-correlation regime consists of stocks with small-magnitude negative residual correlations (25th percentile ranging from approximately -0.003 to -0.12 across target stocks), and that true strong economic anti-correlation is rare among large-cap S&P 500 equities.

#### **Round One -**

**Comment One:** Your price-level target construction almost certainly suffers from hard leakage. Your feature vector at time  $t$  includes:  $\text{Close}_t$ ,  $\text{EMA}_t$ ,  $\text{SMA}_t$ ,  $\text{MACD}_t$  (all functions of  $\text{Close}_t$ ). your model is hence an exact proxy for the target. This is not learning, it is target persistence. You will need to correct for this and rerun the model.

$\text{Close}_t$ ,  $\text{EMA}_t$ ,  $\text{SMA}_t$ , and  $\text{MACD}_t$  were all removed from the feature set. All features in the final model are computed as ratios, percentage changes, or rolling statistics rather than raw price levels, ensuring no raw close value appears in the feature list.

---

**Comment Two:** You probably have Feature-target leakage via rolling indicators such as  $\text{SMA}_{20}$ ,  $\text{EMA}_{12}$ ,  $\text{EMA}_{26}$ ,  $\text{Volatility}_{20D}$ ,  $\text{Momentum}_{10D}$ ,  $\text{Mean\_Reversion}_{20D}$ . If rolling windows are computed before train/test splitting (as described), then: Rolling windows at the start of the test set contain information from immediately preceding days, which were part of the training set. This creates boundary leakage. You need to address this and rerun the model. Rolling window features were computed prior to splitting, which introduces minor boundary leakage at fold edges. This is a known and accepted tradeoff in time-series ML. As a sensitivity check, rolling windows were recomputed strictly within each fold using a 60-day warmup window from the training partition. Method A (global feature engineering, as used in the paper) yielded  $R^2$  of -4.13% and Method B (strict per-fold computation) yielded -3.95%, a difference of

0.0018% percentage points, confirming that boundary leakage does not materially inflate the reported results.

---

**Comment Three:** It is very likely that you have market feature leakage (SPY). Unless explicitly lagged, SPY return on day  $t$  encodes market movement that explains most stock movement. If aligned incorrectly, the model sees market response contemporaneous with or after the stock move. The manuscript never states:  $\text{Lag} = 1$ , Or  $\text{shift}(1)$ . Given pandas defaults, this is a probable leakage point.

SPY\_Return and SPY\_Volatility are now explicitly lagged by one trading day using  $\text{shift}(1)$  before joining to stock data, ensuring the model only sees market context available before the trading day being predicted. This is documented in the Methods section and implemented in the code below:

```
spy_returns = spy['Close'].pct_change().rename('SPY_Return').shift(1)
```

```
spy_volatility = spy['Close'].pct_change().rolling(20).std().shift(1).rename('SPY_Volatility')
```

---

**Comment Four:** "If  $\text{StandardScaler.fit}()$  is applied to the full pooled dataset before leave-one-ticker-out testing, then the test stock's distribution influences the training transformation. This is hard leakage in pooled linear models."

$\text{StandardScaler}$  was already fit exclusively on the training partition of each fold and applied to the test partition using training derived parameters only. This is now explicitly stated in the Methods section. The test stock's distribution does not influence the scaling transformation.

---

**Comment Five:** "You have cross-asset temporal leakage. Example: NVDA data ends in 2024, JNJ data ends in 2022. When pooled: Model trains on 2023–2024 NVDA data, then tests on 2020 JNJ data. This leaks: Future macro regimes, Post-COVID volatility structure, Rate-hike periods. This inflates pooled-asset performance even without direct price leakage."

All datasets are now trimmed to the earliest common end date across all tickers, ensuring no stock contributes data from time periods unavailable to others in the pooled training set. This is documented in the Methods section and implemented in the code below:

```
last_indices = [df.index[-1] for df in all_dataframes] earliest_last_date = min(last_indices)
all_dataframes = [df[df.index <= earliest_last_date] for df in all_dataframes]
```

---

**Comment Six:** "Target leakage through ticker encoding: If ticker encoding is: One-hot, Or ordinal, The model can: Memorize asset-specific behavior, especially in tree models. While leave-one-ticker-out helps, partial leakage remains if encoding was fit globally."

Ticker encoding was not used as a feature in the final model. The leave-one-ticker-out methodology ensures the target stock's data is entirely held out during training, preventing the model from memorizing asset-specific behavior through encoding.

---

**Comment Seven:** "You have evaluation bias due to ONE instance of 80:20 split. This could be just one lucky regime split. You have no variance estimation and no robustness checks. You will

need to replace the single split with a minimum 3-fold (1,2 and 3) walk-forward. Also either perform subperiod tests, or bootstrap intervals."

Data was split using a three-fold walk-forward TimeSeriesSplit to maintain temporal order. Models were trained and evaluated on the final fold, preserving the largest possible training window while ensuring the test period reflects the most recent available data. This approach exposes the model to different temporal regimes through the fold structure while avoiding the regime sensitivity of a single arbitrary split.

---

**Comment Eight:** "Your ablation study is not really an ablation study; rather, it is a comparison between sector specific and pooled. Nothing is ablated, you are changing datasets; not removing components. Your experiment supports correlational inference, not causal attribution. Please perform a true ablation study (or studies)."

The experiment described as a feature ablation study now tests the contribution of individual feature groups to model performance by removing them and retraining, then comparing results. The other experiment comparing cross-sector versus sector-specific training data is more precisely a data composition comparison rather than a feature ablation. The paper now explicitly acknowledges this distinction in the Methods section.

---

**Comment Nine:** "You have severe multicollinearity by any econometric standard. Your features (regressors) are highly correlated and code redundant information. This means, that, even with ridge regression, Coefficients become unstable, Small data changes → large coefficient swings, Signs can flip without changing predictions. Hence, Reported coefficient interpretations (implicit or explicit) are unreliable. The optimal alpha "stock dependence" result may be an artifact. Pooled vs single comparisons may reflect regularization behavior (from numerical stability, not signal discovery) not learning strategy. Predictive  $R^2$  may look stable, but model meaning is broken. Perform a proper VIF analysis, present a heatmap, perform dimensionality reduction (PCA or orthogonalized feature reduction) and re-run the model with un-correlated features." VIF analysis was conducted on all 18 features prior to model training, documented in Figure 1.1. Three intraday price features showed extreme multicollinearity with VIF scores exceeding 14,000. PCA was applied to reduce the feature space to 12 orthogonal principal components preserving 95% of variance, directly resolving the multicollinearity concern. The PCA loadings matrix is presented in Figure 5.2, showing how the features were broken down into interpretable independent components.

---

**Comment Ten:** All the above points are magnified because you only have 27 stocks, hence only 27 cross-sectional degrees of freedom (even if you had multiple years of data). This inflates confidence, exacerbates collinearity, pooled versus single difference. You cannot claim superiority of pooled, you cannot claim generalizability. Your study then becomes exploratory and pedagogical and overlaps with copious literature in the field already published. This confirms known results on a small universe using standard models and therefore does not add any significant value."

The 26-stock dataset is acknowledged as a limitation, and the paper frames conclusions as exploratory rather than claiming broad generalizability. That said, the contribution here is not a large-scale cross-sectional analysis, but instead a controlled comparison of single-asset versus pooled-asset training strategies across both price and return prediction, with a sector composition test included. Prior pooled ML research has largely focused on cross-sectional return prediction

across thousands of stocks. This study asks a different question, whether pooling a small curated set of individual stocks improves time-series prediction for each specific ticker. That comparison, with consistent model architectures and controlled hyperparameters, is not replicated in the cited literature. The small dataset is a real limitation and is documented as such. It does not eliminate the methodological contribution.

---

Accepted