# Using Factor Momentum To Predict Monthly Stock Returns

**Viet Nguyen**
Department of Computer Science
Stanford University
vin041@stanford.edu

## 1 Introduction

In Fall 2020, we worked on a project for CS229 that used a 4 layer neural network to predict monthly stock returns. Given features for month $t$, the model predicts a stock's return over month $t + 1$. Our baseline model improves upon [1] by reducing the feature count from 920 to 7 features, increasing the signal-to-noise ratio.

For this project, we experiment with two different approaches for improving this baseline model. The first approach uses factor momentum features which indicate how well a particular factor has been performing. The second approach adds data from more time periods, leveraging the time-series component of stock returns. Note that throughout this project, we keep the neural architecture fixed; [1] and Figures 2-4 suggest that more complicate architectures would lead to overfitting.

## 2 Related Work

### 2.1 The Baseline Model

We first want to briefly describe the baseline model. In [1], the authors tackle monthly stock return prediction, achieving an $R^2$ of .40% and a Sharpe Ratio of 2.45 with a 4 layer neural network with 920 features. Their test set is 2000-2020, but their model starts to underperform in the latter decade.

Our baseline uses the same setup, but narrows down this feature set to just 7 important features, improving the signal-to-noise ratio and performance. Another major change we make for the baseline is predicting the return Z-scores rather than the return values. In Section 3, we will describe the choice of features in additional detail.

### 2.2 Factor Momentum

We aim to improve upon the baseline model using the recently observed phenomenon "factor momentum". [2] trades 43 "factors", or long-short portfolios of stocks filtered on specific criteria, concluding that last month's factor performance is predictive of next month's. This suggests that the under-performance of the baseline model in 2003 and 2009 can be remedied with time-series data — for example factor momentum may allow the model to take advantage of the phenomenon where value stocks over-perform in the recovery period after a recession [3].

[4] expands on [2], suggesting that time-series factor momentum is more powerful than the originally proposed cross-sectional factor momentum. For each of 20 factors, this paper constructs a portfolio (FMOM) that is long a factor if and only if the factor has had positive returns over the past 12 months.

[5] responds to the prior two papers, further analyzing the FMOM portfolio. This paper confirms the profitability of FMOM, as well as a variant that uses the actual return values rather than an indicator. However, the paper complicates [4] by arguing that at the individual factor level, a few key factors

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

dominate the returns of the FMOM portfolio. These factors are beta, global beta, global value, size, long term reversal, and quality.

# 3 Dataset

## 3.1 Data Sources

We have obtained monthly price and quarterly fundamental data for individual stocks via CRSP/Compustat [6]. We obtain 3 month constant maturity yields from FRED [7] and Fama-French 5-factor data from Ken French's website [8]. The processed dataset includes roughly 1.4 Million stock/month pairs from 01/1982-12/2020, with an average of 3000 stocks per month.

## 3.2 Train/Dev/Test Split

It is nontrivial to split this dataset into a simple train, dev, and test set because it is imperative to avoid look-ahead bias. Furthermore, financial data is non-stationary, meaning that there is no guarantee that data from 2000 will fall into a similar distribution as data from 2020. [1] uses a rolling scheme to divide the data. They take the data from 2000-2020 as their test set. For each year of their test set, they train a new model. To make test predictions on the year 2000, the authors train a neural network on data from 1982 - 1998, use data from 1999 for early stopping and tuning hyperparameters, and make predictions on the year 2000. To make predictions for the year 2001, they increase the training data to include data from 1999, and use data from 2000 for hyperparameter tuning.

We adopt the same approach, but hold out data from 2012 - 2020 as a true test set. As we iterate on our choice of features and architecture, we will only train models to predict stock returns from 2011 and earlier. We will evaluate our final models on data from 2012-2020.

## 3.3 Features

| | datadate | tic | year | month | datacqtr2m | prccm | trt1m | mom1m | mom12m | size1m | b/p2m | OP2m | xrdq2m | xsgaq2m |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **335431** | 01/31/2020 | AMZN | 2020.0 | 1.0 | 2019Q3 | 2008.72 | 8.7064 | 2.6122 | 19.895937 | 920224.32 | 0.061407 | 0.148298 | 9200.0 | 25467.0 |
| **335432** | 02/29/2020 | AMZN | 2020.0 | 2.0 | 2019Q4 | 1883.75 | -6.2214 | 8.7064 | 7.512044 | 1000342.56 | 0.062039 | 0.155640 | 9740.0 | 29515.0 |
| **335433** | 03/31/2020 | AMZN | 2020.0 | 3.0 | 2019Q4 | 1949.72 | 3.5021 | -6.2214 | 22.495763 | 938107.50 | 0.066154 | 0.155640 | 9740.0 | 29515.0 |

Figure 1: AMZN Features before Z-Score transform

For the baseline, we use 7 stock-specific features for each stock month pair. Three of these features are used in the Fama-French 5 Factor model [9]. These are market cap / size, book to price (bp) / value, and profitability (OP). According to the Fama-French 5 Factor model, companies that are small, have high book value relative to price, and are profitable tend to perform well. Profitability is defined in [9] by the following equation:

$$\text{Operating Profitability} = \frac{\text{Revenue} - \text{COGS} - \text{R\&D} - \text{SG\&A} - \text{Interest}}{\text{Book Value}}$$

Following [10], we also use R&D and SG&A which enhance the value feature. Finally, we use two different momentum features. mom1m measures a stock's return over the last month - there tends to be a short-term reversal effect where poor performing stocks in the last month tend to do well, whereas high performing stocks in the last month tend to perform poorly. mom12m measures a stock's return from month t - 12 to month t - 1. Stocks with strong performance in this time window tend to continue to perform well, and stocks with weak performance tend to perform poorly. [1]

Inspired by Lewellen's linear models in [11] and experimentation with regression models from our previous project, we decide to use log transformations of size, bp, R&D, and SG&A. We found that these log transforms improve performance.

[1] To remove outliers from the dataset, we remove stocks with price < \$5 following [9]. we also remove stocks with mom1m > 250 and mom12m > 100.

Note that to avoid lookahead bias, quarterly fundamental features such as profitability are delayed by 2 months. For example, Amazon's 2020 Q4 earnings report was released on 2/2/2021. In this specific case, we make the assumption that we can use these fundamental features to make predictions for AMZN's monthly return from 2/1/2021 - 2/28/2021. In general, most earnings reports are released in mid January, April, July, and October.

Finally, before these features can be fed into a model, they are transformed using a cross-sectional Z-score. For example, if AMZN's mom1m feature has a Z-score of 1.5 in 10/2020, that means it's mom1m feature is 1.5 standard deviations above the average during this month. Similarly, model will attempt to predict the Z-score of the stock's monthly return - we care about the relative ranking of returns rather than the actual value. [2]

# 4 Baseline Experiments

## 4.1 Baseline Hyperparameters

For our baseline model, we use a learning rate of 0.01 with the ADAM optimizer. We train the network for 20 epochs using MSELoss. Following [1] we use a standard 4-layer neural network with hidden layer sizes of 32, 16, and 8 and ReLU activations.

## 4.2 Evaluation Metrics

The model outputs a predicted return Z-score for each (stock, time) pair. We report the model's $R^2$, and a corresponding portfolio's annualized Sharpe Ratio, following [1]. The $R^2$ metric compare the model's residuals to those of a model that always predicts a return Z-score of 0 (the mean return). The portfolio will be formed by taking the long-short portfolio of the top and bottom decile of returns. Of these two metrics, the Sharpe Ratio is the one we aim to maximize.

Note that the $R^2$ metric for this baseline is more challenging than that used in $[1]$. [1] compares their residuals to a model that predicts $0\%$ returns, whereas our models compare their residuals to a model that predicts the ex post average return.

## 4.3 Batch Normalization Test

We ran a quick test to compare the effect of adding batch normalization before every ReLU activation of the baseline model. We generate predictions on the first half of the "test set" for years 2000 - 2011. The baseline model without batch normalization achieves a total $R^2$ of .21% and a Sharpe of 2.80. With batch normalization, the model achieves a total $R^2$ of .19% and a Sharpe of 2.71.

## 4.4 Validation / Test Loss Analysis

From Figure 2, we observe that the validation and test loss can vary greatly depending on the evaluation year. For the validation year 2001 and test year 2002 the correlation between the validation and test loss is 0.31. For the validation year 2002 and test year 2003 there is a correlation of -0.33. Empirically, we find that early stopping on either of $R^2$ or Sharpe decreases performance compared to using a fixed epoch count of 20. In section 7.1, we further analyze why early stopping fails.

# 5 Factor Momentum Features

We test two variants of the two FMOM features described in section 2.2. The first feature is an indicator that is on if and only if a factor has had positive returns over the past 12 months. The second feature uses the past 12 month return of the factor directly. We use FMOM features of subsets the 5 Fama-French Factors, and report the results in Table 1.

All models are trained for 20 epochs, and the features are simply concatenated to the input layer of the baseline model with no batch normalization. Since a separate model is trained for each year from 2000 - 2011, the $R^2$ and Sharpe metrics are averaged over 12 models / years.

[2] When training the models, N/A's (which only apply to the R&D and SG&A features) are replaced with 0, the cross-sectional mean.

Table 1: Adding FMOM features (2000 - 2011)

| FMOM Features | indicator | $R^2(\%)$ | Sharpe |
|---|---|---|---|
| all 5 factors | Yes | -0.24 | 2.41 |
| size / OP / value | Yes | 0.00 | 1.73 |
| size / OP | Yes | 0.00 | 1.13 |
| size / OP | No | -0.44 | 2.20 |

Note that adding FMOM features decreases performance relative to the baseline, which achieves an $R^2$ of .21% and a Sharpe of 2.80.

In one experiment, adding 5 FMOM features using absolute returns causes the model to overfit relative to the baseline, as shown in Figures 3 and 4. The best FMOM model uses all 5 indicator features rather than absolute returns to avoid overfitting.

## 6 Time Series Features

Another approach we took was using the original features, but using features from months prior to month t. Out of the original 7 features, we tried using further delayed versions of 6 to the model — all but mom12m. If a model used features from x months of data, the total input feature count would be $6x + 1$. We report the results in the Table 2.

Table 2: Adding Time Series Features (2000-2011)

| # Time Periods | Epoch Count | $R^2$ (%) | Sharpe |
|---|---|---|---|
| 1 (baseline) * | 20 | 0.20 | 2.81 |
| 2 (run 1) | 20 | 0.30 | 3.47 |
| 2 (run 2) * | 30 | 0.20 | 3.37 |
| 3 (run 1) | 20 | 0.17 | 3.38 |
| 3 (run 2) * | 30 | 0.14 | 2.84 |
| 11 (run 1) | 20 | 0.00 | 2.67 |
| 11 (run 2) * | 20 | -0.01 | 2.88 |

In runs marked with a (*), we choose the number of epochs each model is trained on ex post facto from multiples of 10 up to 50, introducing a small amount of bias into these results. For example, the 2 month model (run 2) was trained for 100 epochs, and test metrics are recorded for each epoch of the training process. We chose to represent model performance using metrics from epoch 30.

We believe that this is a reasonable decision given Figures 5 and 6. Note that as training time increases from 30 to 100, the test Sharpe oscillates with a slight positive trend, suggesting that choosing any epoch count between 20 and 100 will produce a similar test Sharpe. Also note that although the Sharpe ratio increases over 100 epochs, the $R^2$ decreases. This phenomenon is even more extreme for the 11 month model.

The results suggest that adding from a single time period creates the most effective model. Adding more months may overfit the test set.

## 7 Results

### 7.1 How Early Stopping Fails

Empirically, we find that early stopping using validation $R^2$ and Sharpe do not improve performance compared to using a fixed epoch count. To demonstrate this effect, we report the following correlations for the model using 2 months of data. The correlation between validation loss and test Sharpe is 0.003. The correlation between validation Sharpe and test Sharpe is 0.083. The correlation between epoch count (over 100 epochs) and test Sharpe is 0.185. Figures 5 and 6 suggest that the models achieve a peak Sharpe value after some amount of training, at which point $R^2$ drops and the Sharpe ratio oscillates.

### 7.2 Test Results

Considering the analysis from Section 7.1, we run the following experiment for final test results. On the held out test years 2012-2020, we train the baseline model for 30 epochs, reporting the mean and std. $R^2$ and Sharpe for epochs 21 - 30. We also report the max drawdown (MDD), which uses averaged returns of the model at different stages of training to yield a single result. The same process is repeated for the 2 month model for epochs 31 to 40 and the best FMOM model, using 5 indicator features, reporting statistics for epochs 21 - 30. The results are reported below.

Table 3: Test Results: 2012 - 2020

| Model | Epochs Used | $R^2$ mean (%) | $R^2$ std (%) | Sharpe mean | Sharpe std | MDD |
|---|---|---|---|---|---|---|
| Baseline | 21 - 30 | -0.12 | 0.02 | 1.79 | 0.13 | -17.6% |
| FMOM | 21 - 30 | -0.10 | 0.03 | 2.61 | 0.30 | -8.36% |
| 2 Month | 31 - 40 | -0.15 | 0.02 | 1.73 | 0.14 | -13.8% |

### 7.3 Analysis

From the ablation results in Figures 12-14, we notice that each model prioritizes different features. The baseline model prioritizes size, price momentum, and profitability. The 2 month model prioritizes value from the last two time periods, size, and profitability. The FMOM model prioritizes price momentum, R&D, profitability, and profitability momentum. The FMOM model, unlike the other models, seems to take advantage of a larger subset of its features, additionally taking advantage of all other FMOM features.

Surprisingly, the FMOM model significantly outperforms the baseline and 2 month models, which are comparable. In Figures 9 - 11, we observe that all models maintain an average of 0 exposure to the Fama French factors. From these graphs, we observe that the FMOM model manages to avoid drawdowns in 2017 and 2019 that the other models face. This may be a result of the models' more holistic use of FMOM features leading to more stable performance — drawdowns may come all at once, and while the baseline and 2 month models outperformed from 2000-2011, they faced drawdowns on key years in the test set.

I also measure Sharpe ratios achieved after various filters on the data and portfolios, summarized in the following table:

Table 4: Sharpe: Data / Portfolio Filters

| Model | Large Stocks (70%+) | 2m Lag | 3m Lag | 4m Lag | Long | Short |
|---|---|---|---|---|---|---|
| Baseline | 0.73 | 1.35 | 0.34 | 2.45 | 3.09 | -1.68 |
| FMOM | -0.01 | 0.87 | 1.20 | 3.08 | 3.20 | -0.92 |
| 2 Month | -0.08 | 2.37 | 0.11 | 2.12 | 3.08 | -1.46 |

The models' long-only portfolios all perform similarly, with the difference in performance coming down to short-only Sharpe. All models prefer filtering large stocks out of the dataset. This result is opposite of [1], likely because the heavy focus on fundamental features is more valuable for predicting smaller stock performance — large stocks do not move as predictably on earnings.

Most interestingly, the number of months fundamental features are lagged drastically effects model performance. One might expect that the shortest amount of lag would lead to the best performance, since the fundamental features would be the most up to date, but the opposite is true. Since the majority of earnings reports come out in mid January, April, July, and October, it makes sense that the market is more likely to move on these months. The 2 month model manages to perform well on the months before and after earnings, likely as a result of using b/p features on these two time periods. All models except FMOM underperform on a 3 month lag, most likely due to a reliance on fundamental features, which have less signal when earnings are not moving prices.

## 8 Conclusion

In conclusion, we experimented with adding FMOM and time-series data to a baseline 4 layer neural network. Adding indicator FMOM features causes the model to focus on momentum and profitability,

achieving a Sharpe ratio of **2.61** compared to a baseline of **1.79**. Adding additional time-series data improves baseline performance on the month after earnings, perhaps allowing the model to build features comparing how fundamental features changed. This model achieves a Sharpe ratio of **2.37** compared to a baseline of **1.35** on the month after earnings.

An analysis of the results suggests possible next steps on how to improve these models further. The importance of fundamental lag time suggests that engineering features which take into account lag time or training separate models for each lag time period could improve performance. The under performance of the model on large stocks suggest that non-fundamental features may be necessary to predict returns on specific months - perhaps beta, which was championed in [5], or volatility, could improve model performance.

# 9 Appendix

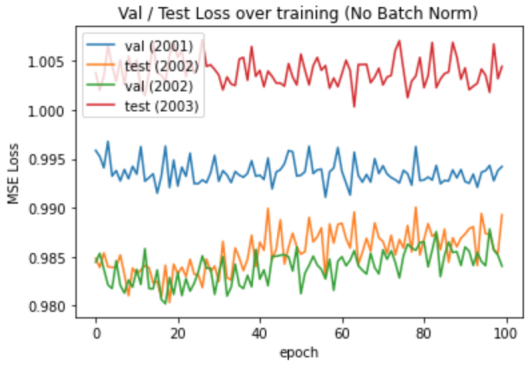


Figure 2: Validation and Test Loss For Consecutive Rolling Models

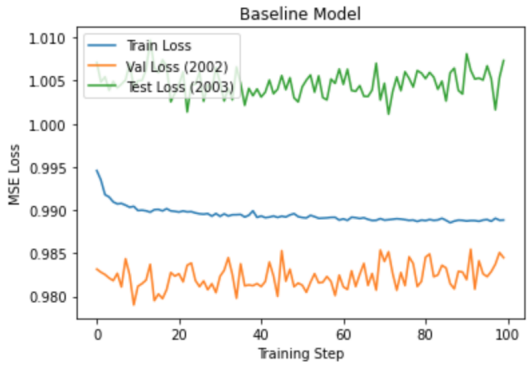


Figure 3: Baseline Loss 2003


Figure 4: FMOM 5 Factor Loss 2003

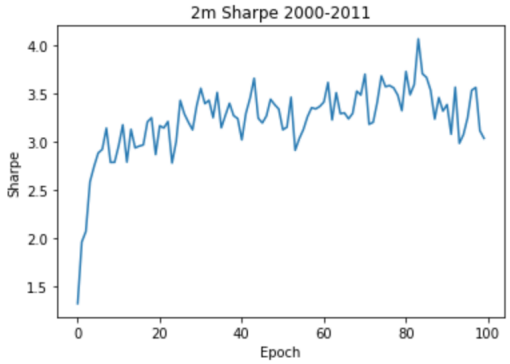


Figure 5: 2m Sharpe over Training

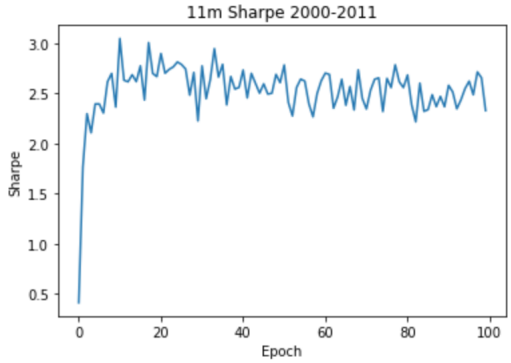


Figure 6: 11m Sharpe over Training

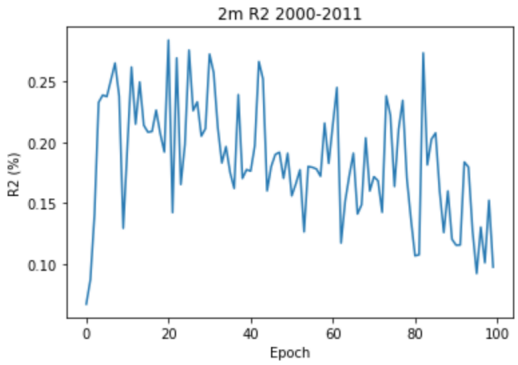


Figure 7: 2m $R^2$ over Training

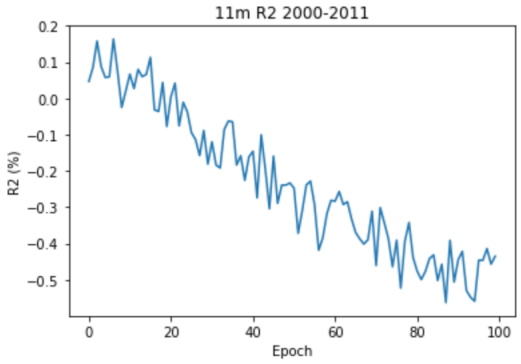


Figure 8: 11m $R^2$ over Training

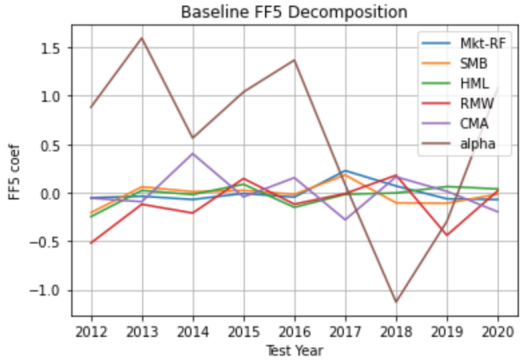


Figure 9: Baseline FF5 Decomposition

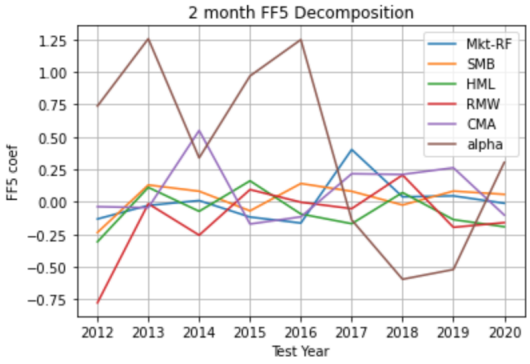


Figure 10: 2 Month FF5 Decomposition

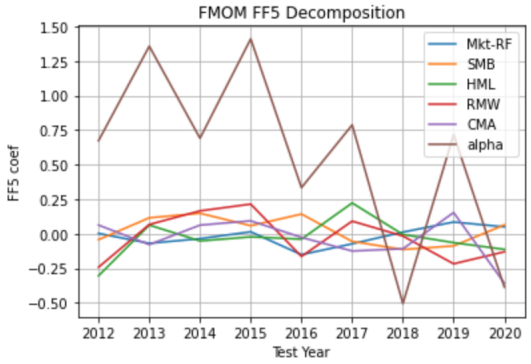


Figure 11: FMOM FF5 Decomposition

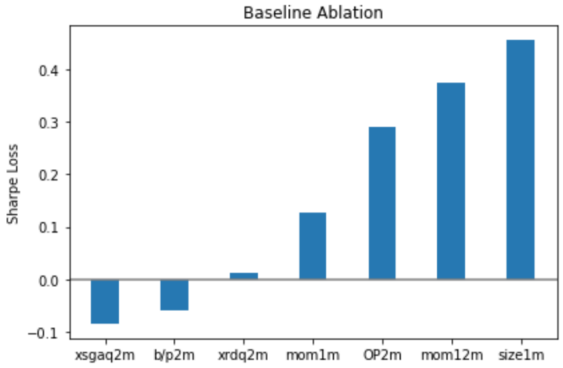


Figure 12: Baseline Ablation

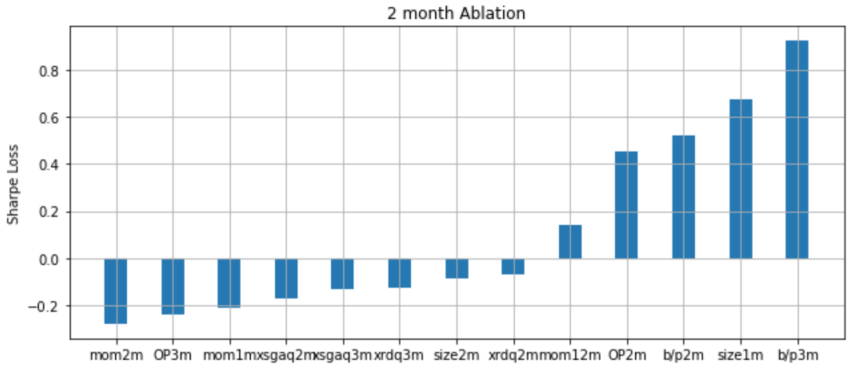


Figure 13: 2 Month Ablation

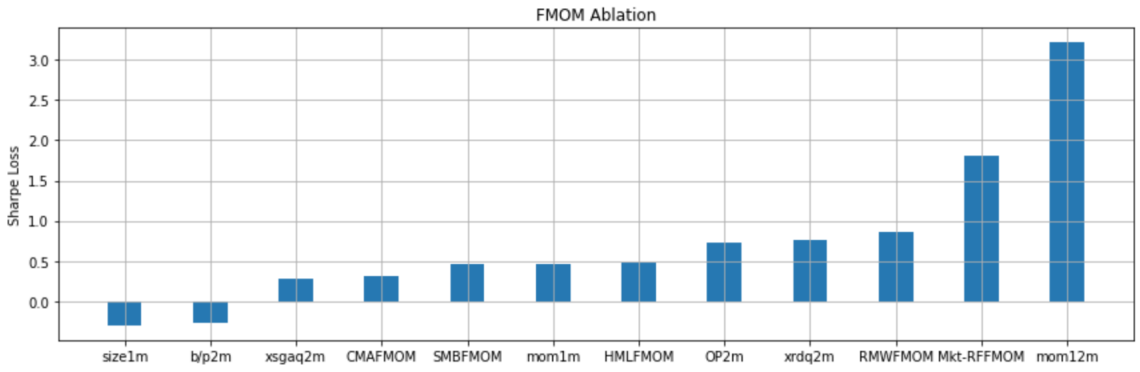


Figure 14: FMOM Ablation

## References


[1] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. Working Paper 25398, National Bureau of Economic Research, December 2018.

[2] Robert D. Arnott, Mark Clements, Vitali Kalesnik, and Juhani T. Linnainmaa. Factor momentum. *SSRN*, March 2021.

[3] Arnav Sheth and Tee Lim. Fama-french factors and business cycles. *SSRN*, December 2017.

[4] Sina Ehsani and Juhani T. Linnainmaa. Factor momentum and the momentum factor. *Journal of Finance*, December 2020.

[5] Minyou Fan, Youwei Li, Ming Liao, and Jiadong Liu. A reexamination of factor momentum: How strong is it? *Journal of Finance*, May 2021.

[6] Crsp/compustat merged database.

[7] 3-month treasury constant maturity rate.

[8] Kenneth R. French. Kenneth r. french - data library.

[9] Eugene F. Fama and Kenneth R. French. A five-factor asset pricing model. *SSRN*, September 2014.

[10] Robert D. Arnott, Campbell R. Harvey, Vitali Kalesnik, and Juhani T. Linnainmaa. Reports of value's death may be greatly exaggerated. *SSRN*, October 2020.

[11] Jonathan W. Lewellen. The cross section of expected stock returns. *SSRN*, August 2014.