Spatio-Temporal Momentum: Jointly Learning Time-Series and Cross-Sectional Strategies

Wee Ling Tan
Department of Engineering Science
Oxford-Man Institute of Quantitative Finance
University of Oxford
weeling@robots.ox.ac.uk

Stephen Roberts
Department of Engineering Science
Oxford-Man Institute of Quantitative Finance
University of Oxford
sjrob@robots.ox.ac.uk

Stefan Zohren
Department of Engineering Science
Oxford-Man Institute of Quantitative Finance
University of Oxford
stefan.zohren@eng.ox.ac.uk

Abstract

We introduce Spatio-Temporal Momentum strategies, a class of models that unify both time-series and cross-sectional momentum strategies by trading assets based on their cross-sectional momentum features over time. While both time-series and cross-sectional momentum strategies are designed to systematically capture momentum risk premia, these strategies are regarded as distinct implementations and do not consider the concurrent relationship and predictability between temporal and cross-sectional momentum features of different assets. We model spatio-temporal momentum with neural networks of varying complexities and demonstrate that a simple neural network with only a single fully connected layer learns to simultaneously generate trading signals for all assets in a portfolio by incorporating both their time-series and cross-sectional momentum features. Backtesting on portfolios of 46 actively-traded US equities and 12 equity index futures contracts, we demonstrate that the model is able to retain its performance over benchmarks in the presence of high transaction costs of up to 5-10 basis points. In particular, we find that the model when coupled with least absolute shrinkage and turnover regularization results in the best performance over various transaction cost scenarios.

1 Introduction

Momentum strategies form a class of systematic strategies that rely on the premise of a persistence in the direction of asset returns over time [25, 15]. These strategies are constructed to exploit the continuation of the underlying trend by increasing or assuming long positions during uptrends, and decreasing or assuming short positions during downtrends. Momentum strategies are often accompanied by volatility targeting, allowing momentum strategies to leverage positions taken during periods of low volatility, and reduce exposures during periods of high volatility. Volatility targeting at

both the asset and portfolio level has been shown to boost Sharpe ratios, reducing the probability of extreme tail returns and minimizing maximum drawdowns for risk asset portfolios [12, 16].

Momentum strategies can be distinguished as belonging to a time-series or cross-sectional strategy. In terms of predictability, the former is primarily driven by the persistence in the trend of individual and market-level returns while the latter is usually deployed as a relative value strategy due to its perceived market-neutrality [4]. As such, current work has mainly regarded time-series and cross-sectional momentum as distinct implementations.

In time-series momentum strategies [25, 11, 20, 41], trading signals for individual assets are constructed based on the asset's own historical returns. In addition, the signals for individual assets in the portfolio are typically constructed independently of other assets. In the presence of a defined universe or portfolio of assets, this strategy does not take into consideration any form of mutual interactions and predictability between assets. We believe that momentum features from other assets represent sources of information as these assets collectively represent the market state at any point in time, and should be considered when constructing trading signals for a given asset.

On the other hand, cross-sectional momentum strategies [15, 33, 29, 30, 31], require a momentum score to first be quantified for each individual asset in the portfolio, before computing a relative ranking of these scores in order to formulate positions for a select group of assets. In the first step of calculating a momentum score, cross-sectional momentum strategies ultimately consider only an asset's own historical returns, independent of returns from other assets. In principle, this is still identical to the way a trading signal is constructed in a time-series momentum strategy. Subsequently in the ranking step, a typical cross-sectional momentum strategy would allocate positions by assuming a maximum long position for assets ranked in the top decile, while taking a maximum short position for assets ranked in the bottom decile. We argue that this maximum long-short allocation does not accurately reflect the underlying trading signal for the selected assets. Moreover, this approach leaves a large majority of assets classified in intermediate deciles with no directional positions. As such, a cross-sectional strategy is unable to exploit any underlying trends for these intermediate assets.

In this work, we combine both types of strategies by considering a form of multi-asset momentum that simultaneously constructs trading signals based on momentum features from multiple assets over time. We model the spatio-temporal momentum strategy using a variety of neural networks trained in a data-driven manner, each representing different complexities. In predicting the trend and position size for any given asset, we consider the asset's own momentum features as well as the momentum features from other assets in the portfolio. In contrast to a time-series momentum strategy, our model learns to incorporate information collectively from the universe of assets into signal construction. Our proposed strategy also directly generates trading signals for every asset as a multi-target output of the neural network, effectively bypassing the need to manually rank assets relative to one another as in the case of cross-sectional momentum. We directly optimize the models with the Sharpe ratio [35], allowing the spatio-temporal momentum networks to learn from risk-adjusted performance [20].

Our strategy closely resembles multitask learning [7, 9], a training framework that aims to train a learner on multiple tasks simultaneously. We observe that the prediction of momentum signals for different assets can be treated as multiple different, but closely related prediction tasks. In the context of spatio-temporal momentum, the model leverages on the usefulness of a shared feature representation between different assets, and is trained by optimizing over an aggregate Sharpe ratio computed from the multi-output model.

2 Related Work

Momentum strategies are typically classified as either time-series or cross-sectional. The work of [25] incorporates volatility scaling [12] into a time-series momentum strategy and documents significant excess returns arising from trading based on the sign of an asset's own returns over the past 12 months, backtesting across 58 instruments and more than 25 years of data. Since then, other works have introduced more complex trend estimation techniques, such as using volatility normalised MACD signals [4] and blending slow and fast momentum strategies with various weights to reduce downside exposure [11]. On the other hand, [15] show that a cross-sectional momentum strategy of buying winners and selling losers over a lookback horizon of 3 to 12 months led to significant excess returns for a portfolio of NYSE and AMEX stocks. The cross-sectional momentum effect has also been observed in international equity markets [33] and futures contracts [29].

Machine learning algorithms have been increasingly developed to perform predictions by extracting and modelling features from data. Deep learning has eliminated the need for manual feature engineering, allowing for hierarchical feature representations to be learnt directly from data [18]. With recent advances in deep learning methods and open source libraries [1, 27], deep neural networks have been applied to model financial datasets such as high frequency limit order book data [43] and to construct portfolios [44]. Very often, neural networks that depend on recurrence and backpropagation through time, like recurrent neural networks (RNN) and long short-term memory networks (LSTM) [14], are used to model temporal relationships commonly present in financial data.

Deep learning algorithms have also been applied to momentum strategies. The work of [20] introduces Deep Momentum Networks (DMNs), a series of neural network architectures directly optimised for portfolio Sharpe ratio, in place of standard regression and classification methods for time-series momentum. The models directly output positions for individual contracts, combining trend estimation and position sizing into a single function. However, we note that the single-output DMN model is trained on batches of input features belonging to different asset classes. This requires the model to learn to construct and output trading signals individually for all types of instruments, regardless of the possibility of negative transfer between asset classes. Our approach adopts a multitask learning approach that incorporates a multi-output target, permitting specialization for individual assets while retaining the advantage of having only a single model that leverages on a common shared representation between different assets.

Multitask learning [7, 9, 34] aims to train a model on multiple tasks simultaneously to improve generalization performance across tasks. It has has been successfully adopted in a range of applications, including computer vision [42, 24, 21] and natural language processing [23, 10, 32]. Training a deep learning-based multitask model involves backpropagation with training signals from multiple related tasks. This can be seen as a form of model regularization via shared representations, specifically by coupling a main task with auxiliary tasks to introduce an inductive bias to the model [34]. We take the approach where all tasks (assets in the portfolio) are treated equally, unlike some multitask learning models that distinguish between a main task and one or more auxiliary tasks.

3 Momentum Strategies

Following the definition of [3, 25], the overall returns of a time-series momentum (TSMOM) strategy that equally diversifies over N_t assets at time t is:

$$r_{t,t+1}^{\text{TSMOM}} = \frac{1}{N_t} \sum_{i=1}^{N_t} X_t^{(i)} \frac{\sigma_{\text{tgt}}}{\sigma_t^{(i)}} r_{t,t+1}^{(i)} \tag{1}$$

where $X_t^{(i)} \in [-1, 1]$ denotes the trading signal or position for asset i at time t. Given the differences in volatility across individual assets, we scale the realized returns $r_{t,t+1}^{(i)}$ by their volatility to target equal risk assignments. We set the annualized volatility target σ_{tgt} to be 15% and estimate the ex-ante volatility $\sigma_t^{(i)}$ with a 60-day exponentially weighted moving standard deviation of daily returns.

Most momentum strategies are concerned with designing a proper trading signal $X_t^{(i)}$. We illustrate this with the following examples which we incorporate as benchmarks in our work:

Long Only In the simplest case, a long only strategy takes a maximum long position $X_t^{(i)} = 1$ for each asset and performs a daily rebalancing according to changes in the ex-ante volatility estimate.

TSMOM In the time-series momentum strategy of *Moskowitz et al., 2012* [25], the position taken for an asset is based on the sign of the asset's returns over the past 12 months: $X_t^{(i)} = \text{sgn}(r_{t-252,t}^{(i)})$

MACD In *Baz et al., 2015* [4], volatility normalised moving average convergence divergence (MACD) indicators are used in place of the sign of returns as signals for trend estimation:

$$\begin{aligned}
&\text{MACD}(i,t,S,L) = m(i,t,S) - m(i,t,L) \\
&\text{MACD}_{\text{norm}}(i,t,S,L) = \frac{\text{MACD}(i,t,S,L)}{\text{std}(p_{t-63:t})} \\
&Y_t^{(i)} = \frac{\text{MACD}_{\text{norm}}(i,t,S,L)}{\text{std}(\text{MACD}_{\text{norm}}(i,t-252:t,S,L))} \\
&X_t^{(i)} := \tilde{Y}_t^{(i)} = \frac{1}{3}\sum_{k=1}^{3} \phi(Y_t^{(i)}(S_k, L_k))
\end{aligned} \tag{2}$$

where $\text{MACD}(i,t,S,L)$ is the MACD value of asset i at time t with a short time scale S and long time scale L. Further, $m(i,t,j)$ is defined as the exponentially weighted moving average of asset i prices at time t, with a time scale j that corresponds to a half-life of $HL = \log(0.5)/\log(1-\frac{1}{j})$. The MACD value is normalized by $\text{std}(p^{(i)}_{t-63:t})$, the 63-day rolling standard deviation of asset i prices. Multiple intermediate MACD signals over different short and long time scales $S_k \in \{8, 16, 32\}$ and $L_k \in \{24, 48, 96\}$ are combined in an equally weighted sum to yield a position $X_t^{(i)}$ (or an aggregated MACD signal $\tilde{Y}_t^{(i)}$) where $\phi(y) = \frac{y\exp(\frac{-y^2}{4})}{0.89}$ is a response function as defined in [4].

Deep Momentum Networks (DMN) *Lim et al., 2019* [20] use deep neural networks to directly generate trading signals for individual assets, combining trend estimation and position sizing into a single function as approximated by the learnt model f using time-series momentum features $\mathbf{u}_t^{(i)}$:

$$X_t^{(i)} = f\left(\mathbf{u}_t^{(i)}; \boldsymbol{\theta}\right) \tag{3}$$

The model parameters $\boldsymbol{\theta}$ are learnt by optimizing over loss metrics that include volatility characteristics of returns. As a benchmark, we consider f to be the Sharpe-optimized LSTM.

CSMOM Following *Jegadeesh & Titman, 1993* [15], we consider a cross-sectional momentum (CSMOM) strategy that scores and ranks an asset based on its returns computed over the past 12 months. The strategy utilizes a decile portfolio, taking a maximum long and short position for the top and bottom 10% of ranked assets respectively.

4 Spatio-Temporal Momentum

Multitask Learning Premise We observe that the prediction of individual momentum signals $X_t^{(i)}$ for different assets can be treated as multiple different, but closely related prediction tasks. In the context of an equally weighted portfolio, we take the approach where all assets (tasks) are treated equally in a multi-target setting. In addition, we follow the definition of multitask learning in [7], where all tasks share the same inputs, unlike some definitions of multitask learning where different inputs are utilized for different tasks.

4.1 Multitask Learning in Spatio-Temporal Momentum

We model the simultaneous prediction of multiple trading signals $X_t^{(i)}$ with model f as a multitask learning problem, with each task i corresponding to predicting the trading signal for asset i. Given time t, our goal is to construct a multitask learning model f that is able to perform N^t tasks of predicting an aggregate trading signal $\mathbf{X}_t \in [-1,1]^{N^t}$ over all assets.

In general, we have an input spatio-temporal tensor $\mathbf{u}_t \in \mathbb{R}^{N^t \times \tau \times d}$, where N^t is the number of assets, τ is the temporal history, d is the number of features, and $\mathbf{u}_t(i,j,k)$ represents the k-th feature of the i-th asset at time $t-j$. We want to learn a model f parameterized by $\boldsymbol{\theta}$:

$$\mathbf{X}_t = f(\mathbf{u}_t; \boldsymbol{\theta}) \tag{4}$$

where

$$\mathbf{X}_t = \begin{bmatrix} X_t^{(1)} \\ X_t^{(2)} \\ \vdots \\ X_t^{(N^t)} \end{bmatrix} \tag{5}$$

In this framework, trading signals $X_t^{(i)}$ are directly computed using features from the universe of assets in the form of the spatio-temporal tensor $\mathbf{u}_t$. Specifically, this reduces to a time-series momentum strategy for $N^t = 1$. The realized returns of this framework still follows Equation (1).

4.2 Deep Learning Architectures

Given that the choice of model architecture is crucial in modelling the relationship between the trading signals and the spatio-temporal momentum features, we examine a range of architectures of different complexities that are able to serve as candidate end-to-end functions.

Single Layer Perceptron (SLP) We consider the simplest case of a fully connected neural network with a single hidden layer that computes a linear combination of the input features prior to activation:

$$\mathbf{X}_t = f(\mathbf{u}_t; \boldsymbol{\theta}) = g(\mathbf{W}^\top \mathbf{u}_t + \mathbf{b}) \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{m \times N^t}$, $\mathbf{u}_t \in \mathbb{R}^m$ with $m = N^t \cdot \tau \cdot d$, $\mathbf{b} \in \mathbb{R}^{N^t}$ and $g = \tanh$ is the activation function.

Multilayer Perceptron (MLP) We consider a fully connected neural network with two hidden layers, representing a step up in model complexity from the SLP model:

$$\mathbf{X}_t = f(\mathbf{u}_t; \boldsymbol{\theta}) = g[\mathbf{W}^{[2]\top} \sigma(\mathbf{W}^{[1]\top} \mathbf{u}_t + \mathbf{b}^{[1]}) + \mathbf{b}^{[2]}] \tag{7}$$

where $\mathbf{W}^{[l]}$ and $\mathbf{b}^{[l]}$ are the weights and biases of the hidden layers and $g = \sigma = \tanh$ are the activation functions. Given the non-linear activation function σ, the model incorporates non-linearity with respect to the features of $\mathbf{u}_t$ prior to mapping into trading signals.

Convolutional Neural Networks (CNN) Apart from their applications in computer vision [37, 13], CNNs have been used to model multivariate time-series by incorporating convolutional filters that extract features from temporal data [6]. These models rely on autoregressive causal convolutions and optionally, dilated convolutions [26] that allow the encoding of features over various receptive fields. The use of causal convolutions preserves the autoregressive ordering of temporal features, in that predictions made at time t cannot depend on any future time steps $t+1, \cdots, T$.

Long Short-term Memory (LSTM) Given the issues with exploding and (primarily) vanishing gradients in learning neural networks especially with long sequential data [5], RNNs that incorporate memory cell states and gating mechanisms such as LSTMs [14] have been used to address these limitations [19]. The combination of memory cell states and gating functions helps in backpropagating gradients through time, allowing RNNs to better learn long-range dependencies in sequences.

We consider both the CNN and LSTM architectures for modelling spatio-temporal momentum. For full details of the implementations, we refer the reader to Appendix A.

4.3 Training Details

4.3.1 Loss Function

Incorporating risk-adjusted metrics such as the Sharpe ratio during optimization have allowed models to better incorporate risk into learning and generation of trading signals [20]. Given a set $\mathcal{D} = \{(\mathbf{u}_t, \mathbf{X}_t = f(\mathbf{u}_t; \boldsymbol{\theta})) \mid \mathbf{X}_t \in [-1, 1]^{N^t}\}_{t=1}^T$ of spatio-temporal tensors and their corresponding

signals, we define the loss function $\mathcal{L}_{\text{sharpe}}(\boldsymbol{\theta})$ over $\mathcal{D}$ as the annualized Sharpe ratio:

$$\mathcal{L}_{\text{sharpe}}(\boldsymbol{\theta}) = \sum_{i=1}^{N^t} \lambda_i \cdot \mathcal{L}^{(i)}_{\text{sharpe}}(\boldsymbol{\theta}) \tag{8}$$

$$\mathcal{L}^{(i)}_{\text{sharpe}}(\boldsymbol{\theta}) = -\frac{\sum_{t=1}^{T} R_i(t) \times \sqrt{252}}{\sum_{t=1}^{T} R_i(t)^2 - \left[\sum_{t=1}^{T} R_i(t)\right]^2} \tag{9}$$

$$R_i(t) = X_t^{(i)} \, \frac{\sigma_{\text{tgt}}}{\sigma_t^{(i)}} \, r_{t,t+1}^{(i)} \tag{10}$$

where $R_i(t)$ represents the volatility-scaled captured returns for asset i from time t to $t+1$. The multitask loss weights λ_i balance task importance for the individual task-specific loss functions $\mathcal{L}^{(i)}_{\text{sharpe}}(\boldsymbol{\theta})$. In the case where all tasks (assets) are treated equally, we have $\lambda_i = 1/N^t$.

Shrinkage Penalty Taking into account the high dimensionality of the weight matrix $\mathbf{W}$ as per Equation (6), we incorporate L_1 regularization as an additional penalty term to the loss function of the SLP. This encourages feature selection and model sparsity as the weight coefficients of features that are of less relevance to prediction are shrunk towards zero. We incorporate L_1 regularization in the form of a penalty term on the sum of absolute weights of matrix $\mathbf{W}$ in the overall loss function:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{sharpe}}(\boldsymbol{\theta}) + \alpha \sum_{w_{ij} \in \mathbf{W}} |w_{ij}| \tag{11}$$

where α is a hyperparameter controlling the shrinkage penalty term.

4.3.2 Optimization

In processing our datasets, we conduct a train-validation split with the earlier 90% of data for training and the most recent 10% reserved for validation. To minimize the overall empirical loss, we perform backpropagation using minibatch stochastic gradient descent with the Adam optimizer [17], and initiate early stopping based on the validation set loss. For all machine learning methods, we conduct hyperparameter optimization with 100 iterations of random search for selecting optimal candidate models. We refer the reader to Appendix A for a detailed description of training parameters and specific details for the models.

5 Performance Evaluation

5.1 Overview of Datasets

US Equities Our first dataset consists of 46 actively-traded US equities from the Financials sector with data obtained from the Center for Research in Security Prices (CRSP) [8]. We work with daily returns data, rebalancing the portfolio daily and performing strategy backtesting from 1990 to 2022.

Equity Index Futures Our second dataset comprises 12 ratio-adjusted continuous equity index futures contracts with data obtained from the Pinnacle Data Corp CLC Database [28]. We work with daily returns data, performing portfolio rebalancing daily and backtesting from 2003 to 2020.

The full list of instruments is detailed in Appendix B.

5.2 Backtest Details

We utilize an expanding window approach, where all models are trained with every iteration of 5 additional years as it becomes available. Taking US Equities as an example, the first iteration would involve training and validating on the period from 1990 to 1995, then fixing the model weights and evaluating the model on out-of-sample data from 1995 to 2000. The second iteration would involve training and validating from 1990 to 2000, and evaluating from 2000 to 2005, and so on. We conduct each experiment over multiple random seeded runs and report the aggregate performance.

5.3 Momentum Features

In constructing a general input spatio-temporal tensor $\mathbf{u}_t \in \mathbb{R}^{N^t \times \tau \times d}$ as per Equation (4), we include the below d momentum features:

- **F.1 Volatility Normalized Returns** – we use $r^{(i)}_{t-k,t}/(\sigma^{(i)}_t \sqrt{k})$, representing asset returns normalized by daily volatility estimates scaled to a time scale $k \in \{1, 20, 63, 126, 252\}$, corresponding to daily, monthly, quarterly, semiannual and annual returns.
- **F.2 MACD** – we take volatility normalised MACD signals $Y^{(i)}_t(S_k, L_k)$ as per Equation (2) as input features, using short and long time scales $S_k \in \{8, 16, 32\}$ and $L_k \in \{24, 48, 96\}$.

5.4 Results and Discussion

In evaluating the performance of all strategies, we use the following annualized metrics:

- **M.1 Profitability** – Expected Returns ($\mathbb{E}$[Returns]), Hit Rate
- **M.2 Risk** – Volatility (Vol.), Downside Deviation, Maximum Drawdown (MDD)
- **M.3 Performance Ratios** – Sharpe, Sortino and Calmar Ratios, Average Profit over Loss $\left(\frac{\text{Ave. P}}{\text{Ave. L}}\right)$

For US Equities (and respectively for Equity Index Futures), we report the aggregated out-of-sample performance of all strategies from 1995 to 2022 (2008 to 2020) for overall returns computed in accordance with Equation (1). In this section, we compute the performance of the strategies in the absence of transaction costs to understand the raw predictive ability of the strategies. We provide an analysis of the impact of transaction costs in Section 5.6. We first present the performance of all strategies from their raw signal outputs in Table 1 (Table 3). In order to facilitate the comparison between different strategies, we apply to all strategies an additional layer of volatility scaling at the portfolio level to target an annualized volatility of 15% and report the performance in Table 2 (Table 4) and cumulative returns in Figure 1 (Figure 2).

5.4.1 US Equities

From Table 1, we first observe that the cross-sectional decile portfolio was an unprofitable strategy as seen from the CSMOM strategy delivering negative returns over the backtest period. In addition, classical time-series momentum portfolios like TSMOM and MACD generally underperformed compared to a simple long only approach. Two machine learning methods, the DMN and SLP were able to outperform the long only approach as seen from their performance ratios.

With the introduction of volatility scaling at the portfolio level, we observe improvements in performance across all strategies as shown in Table 2. In particular, we see that spatio-temporal momentum (STMOM) methods, with the exception of CNN, experienced the largest increase in performance ratios among machine learning methods, with the Sharpe ratio of the SLP increasing by about 134% (MLP - 251%, LSTM - 217%) as compared to an increase of 43% for the DMN. This increase led to the MLP and LSTM outperforming the long only strategy. Both the DMN and SLP demonstrated the best performance and exhibited a large gap above all other methods as seen from Table 2 and the cumulative returns plot in Figure 1. This can be attributed to the DMN and SLP both capturing higher expected returns while being subject to lower downside risks.

Within STMOM models, we observe a deterioration in performance of the STMOM strategy with an increased level of model complexity, as seen from the underperformance of the MLP, LSTM and CNN relative to a simple SLP model trained with a shrinkage penalty. This is contrary to the expectation that a model of greater complexity would be better suited to model the dynamics of spatio-temporal momentum. It is also possible that the underperformance in complex architectures is associated with difficulties in training models with multiple model configurations, especially the CNN as seen from its poor performance, echoing a similar conclusion from [20].

The underperformance of some STMOM models as compared to the DMN may be attributed to the relatively low signal-to-noise ratio of returns data, coupled with the limited amount of data available to STMOM strategies. During training, STMOM deep learners are exposed to only t samples, while deep TSMOM strategies like the DMN have access to a much larger pool of $t \times N^t$ samples. As a result, overly complex STMOM models also run the risks of overfitting to in-sample data.

Table 1: Performance Metrics for Strategies – Raw Signal Outputs **(US Equities)**

	E[Return]	Vol.	Downside Deviation	MDD	Sharpe	Sortino	Calmar	Hit Rate	$\frac{\text{Ave. P}}{\text{Ave. L}}$
Benchmarks									
Long Only	**0.068**	0.102	0.072	0.235	0.667	0.944	0.290	0.541	0.951
TSMOM	0.012	0.067	0.050	0.287	0.177	0.240	0.042	0.526	0.932
MACD	0.001	0.045	0.033	0.237	0.021	0.029	0.004	0.519	0.931
CSMOM	-0.033	0.048	0.036	0.631	-0.702	-0.937	-0.053	0.494	0.911
Reference									
DMN	0.056	**0.028**	**0.018**	**0.067**	**2.043**	**3.145**	**0.924**	**0.598**	**1.055**
	(0.008)	(0.008)	(0.005)	(0.029)	(0.263)	(0.452)	(0.258)	(0.005)	(0.054)
STMOM									
SLP	0.032	0.029	0.020	0.075	1.114	1.609	0.435	0.581	0.955
	(0.006)	(0.005)	(0.004)	(0.015)	(0.182)	(0.288)	(0.092)	(0.007)	(0.041)
MLP	0.013	0.044	0.032	0.176	0.296	0.410	0.082	0.544	0.902
	(0.005)	(0.008)	(0.006)	(0.044)	(0.106)	(0.148)	(0.040)	(0.007)	(0.038)
CNN	0.010	0.051	0.037	0.167	0.195	0.273	0.066	0.516	0.980
	(0.006)	(0.006)	(0.005)	(0.037)	(0.106)	(0.150)	(0.043)	(0.007)	(0.029)
LSTM	0.014	0.044	0.031	0.159	0.320	0.458	0.103	0.546	0.903
	(0.004)	(0.008)	(0.006)	(0.053)	(0.108)	(0.158)	(0.061)	(0.009)	(0.036)

(Standard deviation shown in parentheses)

Table 2: Performance Metrics for Strategies – Rescaled to Target Volatility **(US Equities)**

	E[Return]	Vol.	Downside Deviation	MDD	Sharpe	Sortino	Calmar	Hit Rate	$\frac{\text{Ave. P}}{\text{Ave. L}}$
Benchmarks									
Long Only	0.131	0.155	0.109	0.344	0.841	1.197	0.380	0.541	0.976
TSMOM	0.056	0.157	0.112	0.470	0.358	0.501	0.119	0.526	0.960
MACD	0.038	0.157	0.112	0.524	0.245	0.343	0.073	0.519	0.968
CSMOM	-0.101	**0.154**	0.115	0.964	-0.655	-0.880	-0.105	0.494	0.919
Reference									
DMN	**0.487**	0.167	0.105	**0.260**	**2.920**	**4.647**	**1.887**	**0.598**	**1.179**
	(0.019)	(0.001)	(0.002)	(0.029)	(0.119)	(0.229)	(0.181)	(0.005)	(0.020)
STMOM									
SLP	0.423	0.162	**0.102**	0.301	2.609	4.161	1.428	0.581	1.170
	(0.048)	(0.001)	(0.002)	(0.033)	(0.282)	(0.524)	(0.252)	(0.007)	(0.034)
MLP	0.175	0.169	0.110	0.410	1.040	1.590	0.439	0.544	1.032
	(0.030)	(0.013)	(0.002)	(0.070)	(0.178)	(0.270)	(0.109)	(0.007)	(0.020)
CNN	0.035	0.200	0.139	0.665	0.192	0.314	0.079	0.516	0.977
	(0.041)	(0.059)	(0.056)	(0.318)	(0.223)	(0.340)	(0.082)	(0.007)	(0.032)
LSTM	0.178	0.182	0.136	0.544	1.015	1.422	0.405	0.546	1.025
	(0.040)	(0.028)	(0.035)	(0.264)	(0.305)	(0.511)	(0.202)	(0.009)	(0.028)

5.4.2 Equity Index Futures

Similar to US Equities, it is clear from Table 3 that the cross-sectional decile portfolio was again unprofitable with the CSMOM strategy delivering negative returns. Both classical time-series momentum portfolios like TSMOM and MACD and the reference DMN were profitable but underperformed a long only approach. With the exception of CNN, all STMOM methods demonstrated the best performance, with the SLP and MLP models outperforming the DMN by more than four times as seen from their performance ratios.

Incorporating volatility scaling at the portfolio level, we notice improvements in the performance for all strategies as shown in Table 4. The increase in performance ratios for the benchmarks strategies and DMN were minimal, whereas STMOM methods experienced larger increases in their performance ratios with volatility scaling. This increase drove STMOM strategies to further outperform all other strategies, with the SLP model outperforming the DMN by more than six times in risk-adjusted returns. Echoing the observation for US Equities, the simplest SLP model again demonstrated the best performance above all other STMOM methods as seen from Table 4 and the cumulative returns

plot in Figure 2, further lending support to the notion that a model of lower complexity is better suited to model spatio-temporal momentum.

Table 3: Performance Metrics for Strategies – Raw Signal Outputs **(Equity Index Futures)**

	E[Return]	Vol.	Downside Deviation	MDD	Sharpe	Sortino	Calmar	Hit Rate	$\frac{\text{Ave. P}}{\text{Ave. L}}$
Benchmarks									
Long Only	**0.054**	0.125	0.093	0.275	0.427	0.574	0.195	0.549	0.883
TSMOM	0.020	0.107	0.078	0.251	0.191	0.262	0.081	0.523	0.944
MACD	0.006	0.068	0.050	0.175	0.081	0.111	0.032	0.529	0.905
CSMOM	-0.048	0.081	0.060	0.554	-0.584	-0.786	-0.086	0.489	0.946
Reference									
DMN	0.012	0.034	0.026	0.102	0.301	0.410	0.110	0.525	0.971
	(0.017)	(0.024)	(0.018)	(0.066)	(0.220)	(0.307)	(0.088)	(0.009)	(0.042)
STMOM									
SLP	0.047	0.049	0.035	0.152	1.242	1.856	0.657	**0.574**	0.992
	(0.017)	(0.019)	(0.015)	(0.081)	(0.834)	(1.433)	(0.781)	(0.018)	(0.142)
MLP	0.037	**0.032**	**0.022**	**0.088**	**1.319**	**2.050**	**0.683**	0.559	**1.075**
	(0.014)	(0.013)	(0.010)	(0.063)	(0.593)	(1.065)	(0.478)	(0.014)	(0.127)
CNN	-0.016	0.065	0.049	0.291	-0.215	-0.283	-0.045	0.518	0.886
	(0.014)	(0.020)	(0.015)	(0.111)	(0.170)	(0.228)	(0.033)	(0.014)	(0.048)
LSTM	0.039	0.057	0.040	0.172	0.746	1.079	0.289	0.550	0.973
	(0.013)	(0.014)	(0.010)	(0.070)	(0.329)	(0.529)	(0.198)	(0.016)	(0.098)

(Standard deviation shown in parentheses)

Table 4: Performance Metrics for Strategies – Rescaled to Target Volatility **(Equity Index Futures)**

	E[Return]	Vol.	Downside Deviation	MDD	Sharpe	Sortino	Calmar	Hit Rate	$\frac{\text{Ave. P}}{\text{Ave. L}}$
Benchmarks									
Long Only	0.070	**0.154**	0.114	0.319	0.456	0.616	0.221	0.549	0.887
TSMOM	0.033	**0.154**	0.112	0.355	0.212	0.291	0.092	0.523	0.946
MACD	0.023	0.156	0.114	0.390	0.145	0.198	0.058	0.529	0.915
CSMOM	-0.107	**0.154**	0.115	0.838	-0.697	-0.929	-0.128	0.489	0.929
Reference									
DMN	0.055	0.162	0.115	0.391	0.340	0.478	0.156	0.525	0.966
	(0.026)	(0.006)	(0.002)	(0.086)	(0.165)	(0.233)	(0.093)	(0.009)	(0.028)
STMOM									
SLP	**0.333**	0.161	0.104	0.244	**2.066**	**3.228**	**1.619**	**0.574**	1.121
	(0.084)	(0.003)	(0.004)	(0.076)	(0.498)	(0.910)	(0.957)	(0.018)	(0.071)
MLP	0.288	0.162	0.107	**0.238**	1.776	2.699	1.302	0.559	**1.123**
	(0.058)	(0.004)	(0.005)	(0.059)	(0.333)	(0.560)	(0.452)	(0.014)	(0.048)
CNN	0.030	0.164	0.112	0.443	0.174	0.270	0.089	0.518	0.963
	(0.046)	(0.015)	(0.002)	(0.121)	(0.279)	(0.412)	(0.101)	(0.014)	(0.023)
LSTM	0.251	0.195	**0.103**	0.298	1.389	2.451	0.890	0.550	1.118
	(0.085)	(0.115)	(0.004)	(0.063)	(0.384)	(0.892)	(0.415)	(0.016)	(0.087)

For subsequent sections, we focus our analysis on the US Equities dataset.

5.5 Signal Diversification

From Figure 3, the SLP exhibits mostly zero to moderately positive correlation with other momentum strategies, with the correlation between SLP and DMN at about 46%. The correlations of the SLP with other strategies are also fairly unstable, with the SLP periodically displaying a negative correlation with TSMOM, CSMOM and DMN, translating into possible benefits from signal diversification.

We evaluate the combination of time-series momentum (TSMOM and DMN) with a cross-sectional momentum strategy (CSMOM) and compare it to a spatio-temporal momentum strategy. Focusing on the Sharpe ratio, we see from Table 5 that the combinations of both TSMOM+CSMOM (0.177) and DMN+CSMOM (2.115) portfolios do not outperform a single SLP portfolio (2.609), demonstrating that the benefit of using a spatio-temporal momentum strategy outweighs a portfolio that combines

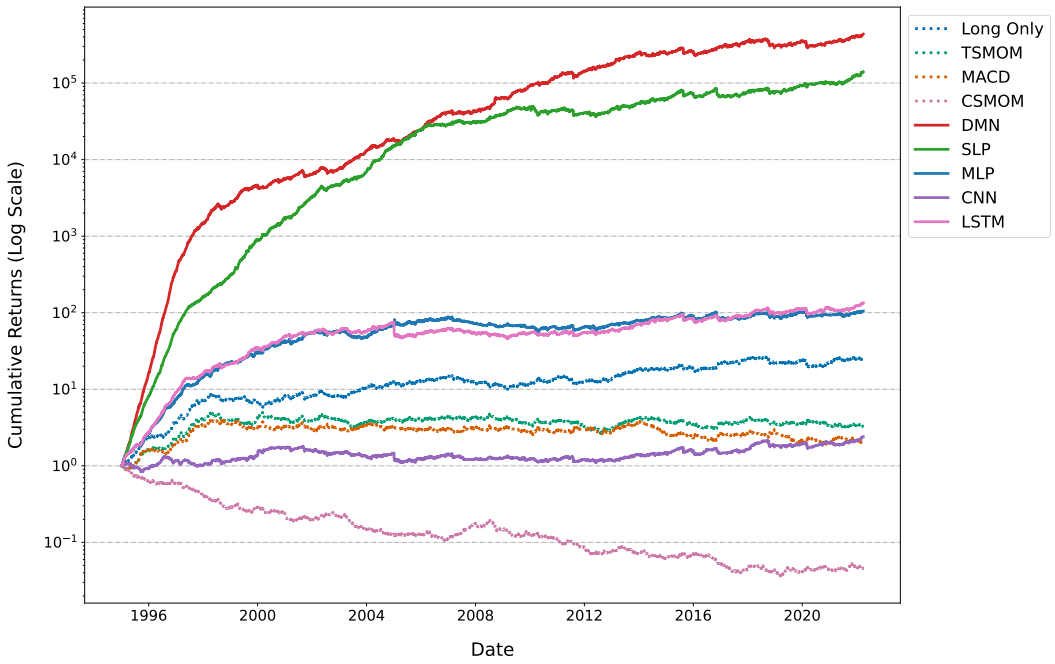

Figure 1: Cumulative Returns - Rescaled to Target Volatility **(US Equities)**

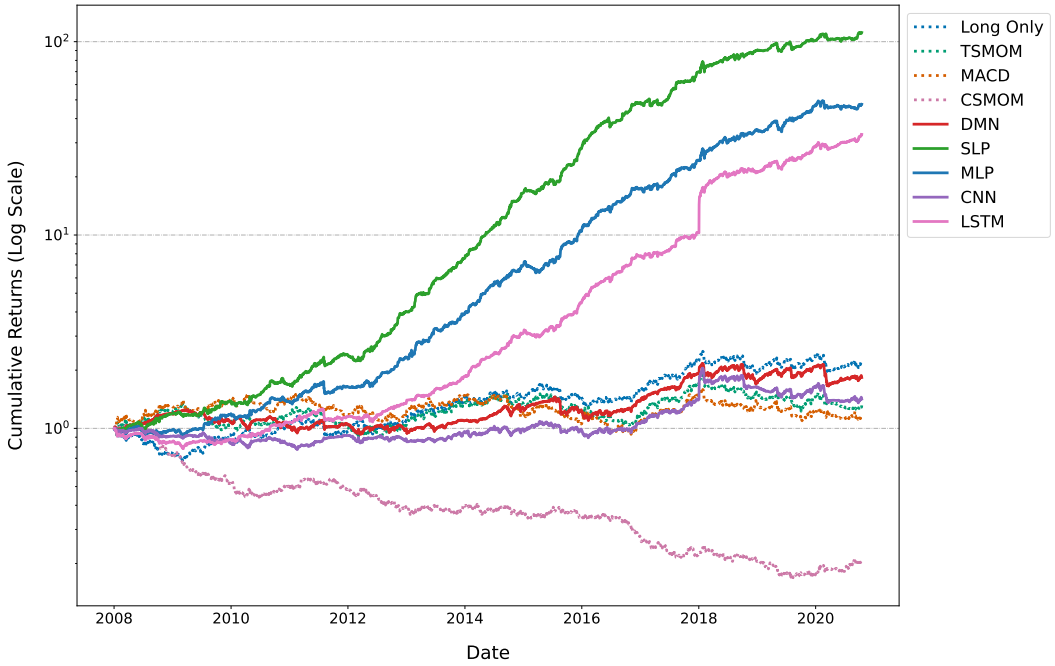

Figure 2: Cumulative Returns - Rescaled to Target Volatility **(Equity Index Futures)**

both time-series and cross-sectional momentum, thus indicating that the model is learning meaningful interactions between the time-series and cross-sectional domain which are not captured by separate models. In addition, we analyze the combination of a time-series momentum (DMN) with a spatio-temporal momentum strategy (SLP). The combination of DMN (2.920) and SLP (2.609) yielded an overall higher Sharpe ratio (3.304), highlighting substantial scope for strategy diversification by incorporating a spatio-temporal momentum strategy.

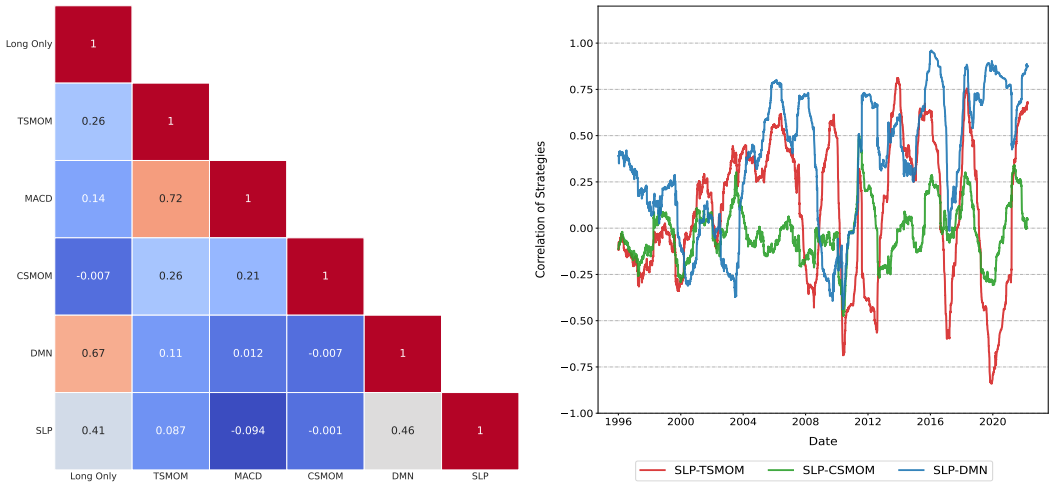

Figure 3: **(Left)** Correlation of Strategies **(Right)** 1Y Rolling Correlations of SLP **(US Equities)**

Table 5: Performance Metrics for Strategy Combinations – Rescaled to Target Volatility **(US Equities)**

	E[Return]	Vol.	Downside Deviation	MDD	Sharpe	Sortino	Calmar	Hit Rate	Ave. P / Ave. L
Combination									
TSMOM+CSMOM	0.028	**0.156**	0.113	0.577	0.177	0.245	0.048	0.521	0.946
DMN+CSMOM	0.340	0.161	0.106	0.307	2.115	3.222	1.132	0.572	1.099
	(0.025)	(0.001)	(0.001)	(0.045)	(0.153)	(0.261)	(0.189)	(0.005)	(0.017)
DMN+SLP	**0.551**	0.167	**0.101**	**0.230**	**3.304**	**5.441**	**2.402**	**0.603**	**1.229**
	(0.027)	(0.001)	(0.001)	(0.013)	(0.151)	(0.308)	(0.147)	(0.005)	(0.026)

5.6 Transaction Costs and Turnover Regularization

In this section, we analyze the impact of transaction costs on the performance of all strategies. Following [3], we define turnover $\text{TO}_t^{(i)}$ as the absolute daily change in the trading signal of an asset:

$$\text{TO}_t^{(i)} = \sigma_{\text{tgt}} \left| \frac{X_t^{(i)}}{\sigma_t^{(i)}} - \frac{X_{t-1}^{(i)}}{\sigma_{t-1}^{(i)}} \right| \tag{12}$$

which encompasses the amount of rebalancing as determined by shifts in both volatility estimates and underlying positions. In Figure 4, we plot the distributions of turnover averaged across all assets for individual strategies. We observe that machine learning models trade considerably more than the benchmarks, with the SLP having a lower turnover than the DMN. To study the impact of transactions costs on performance, we compute ex-cost Sharpe ratios using captured returns net of transaction costs $\tilde{r}_{t,t+1}^{\text{TSMOM}}$:

$$\tilde{r}_{t,t+1}^{\text{TSMOM}} = \frac{1}{N_t} \sum_{i=1}^{N_t} \left(X_t^{(i)} \frac{\sigma_{\text{tgt}}}{\sigma_t^{(i)}} r_{t,t+1}^{(i)} - c \cdot \text{TO}_t^{(i)} \right) \tag{13}$$

where c is a constant approximating the specific transaction cost scenario. Based on the ex-cost Sharpe Ratios in Table 6, the non-regularized DMN and SLP models are able to retain their performance over benchmarks for transactions costs of up to $c = 5$ to 10 basis points, which are considered to be realistic transaction costs for equity markets. Following the approach of [20], we investigate the

effect of incorporating turnover regularization during training in the form of optimizing for ex-cost Sharpe ratios computed using $\tilde{r}^{\text{TSMOM}}_{t,t+1}$. For non-recurrent models that do not generate sequential signals, such as the SLP, we introduce a localized minibatch turnover regularization in the form:

$$\widetilde{\text{TO}}^{(i)}_t = \sigma_{\text{tgt}} \left| \frac{X^{(i)}_t}{\sigma^{(i)}_t} - \frac{X^{(i)}_{t^*}}{\sigma^{(i)}_{t^*}} \right| \tag{14}$$

where $t \neq t^*$ with t and t^* representing consecutive samples within a given training minibatch. Compared to its non-regularized counterpart, incorporating turnover regularization for the DMN resulted in a lower turnover but better performance is only observed at a transaction cost of $c = 10$ basis points, while performance for all other transaction cost scenarios deteriorated. On the other hand, turnover regularization consistently improved the SLP's performance across all transaction cost scenarios. Referring to the distribution of average turnover for the regularized SLP, we observe a lower mean turnover but an increase in the spread between the minimum, maximum and interquartile range. This is likely a direct consequence of performing stochastic gradient descent with the localized minibatch turnover regularization as per Equation (14), requiring batching over a shuffled training set.

Table 6: Impact of Transactions Costs on Sharpe Ratio – Rescaled to Target Volatility **(US Equities)**

Transaction Cost (Basis Points)	**0.0**	**0.5**	**1.0**	**2.0**	**3.0**	**4.0**	**5.0**	**10.0**
Benchmarks								
Long Only	0.841	0.839	0.838	0.835	0.832	0.829	0.826	0.812
TSMOM	0.358	0.347	0.336	0.315	0.293	0.271	0.249	0.140
MACD	0.245	0.238	0.232	0.219	0.207	0.194	0.182	0.119
CSMOM	-0.655	-0.683	-0.710	-0.765	-0.820	-0.875	-0.930	-1.204
Reference								
DMN	**2.920**	**2.844**	**2.768**	**2.615**	**2.462**	**2.308**	**2.153**	1.375
DMN+Reg	2.073	2.044	2.015	1.957	1.899	1.840	1.782	**1.486**
STMOM								
SLP	2.609	2.518	2.427	2.243	2.060	1.876	1.691	0.762
SLP+Reg	**2.672**	**2.603**	**2.534**	**2.395**	**2.256**	**2.116**	**1.976**	**1.271**

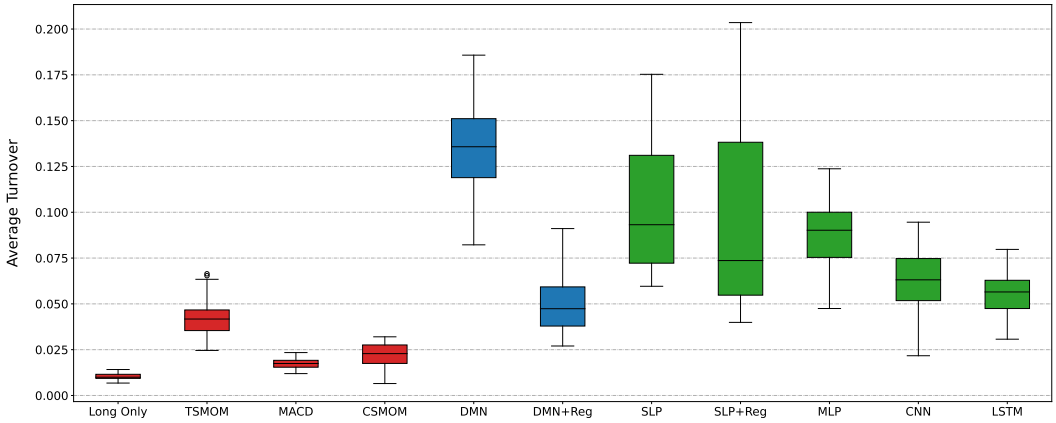

Figure 4: Average Turnover of Strategies **(US Equities)**

5.7 Interpretability

Apart from demonstrating the best performance, the SLP is the simplest model out of all STMOM architectures. We perform an analysis on the interpretability of the trained SLP using SHAP (SHapley Additive exPlanations) [22], a model-agnostic interpretation method for computing feature importance and effects for machine learning models. SHAP values associated to each individual feature measures the change in the expected value of the model's prediction when conditioning on that feature. We approximate SHAP values with Deep SHAP [22], a high-speed approximation algorithm which builds on DeepLIFT [36].

Prediction for a Single Asset We analyze the importance and effects of the spatio-temporal features on the predicted signal of a single asset – BAC, by plotting SHAP values for the top 20 features ordered according to their importance in Figure 5. It is evident that the top features are dominated by volatility normalized MACD, indicating that the MACD features of assets are, on average, contributing more to the predicted signal of BAC as compared to other types of features like volatility normalized returns. This could mean that the cross over of exponentially weighted moving averages are considered significant momentum features for the model in generating its trading signals. Interestingly, the top features do not contain much of the BAC's own features, and instead other assets' features in the cross-section are considered important in its own predicted signal.

Moving from low to high feature values and vice versa, there are clear patterns in the resultant impact of spatio-temporal features on the predicted signal output as seen from the gradual transitions in colour. Momentum and mean-reversion effects of individual features are also relatively distinct and can be inferred from the SHAP plot. Taking the first two features as examples, the first feature "AJG_t-4_MACD_32_96" exhibits a mean-reversion effect as higher feature values lead to a lower predicted signal for BAC, while the second feature "JEF_t-3_MACD_32_96" shows a momentum effect as higher feature values lead to a higher predicted signal for BAC.

Prediction for All Assets To examine global feature importance, we compute the mean absolute SHAP values for each spatio-temporal feature on the predicted signal for all assets in US Equities and plot their cumulative impact in Figure 6. Ordered according to importance, the top 20 features are again dominated by volatility normalized MACD features. We note that the cumulative absolute impact of a feature does not distinguish between its momentum or mean-reversion effects, and serves only as a proxy of its feature importance over all assets.

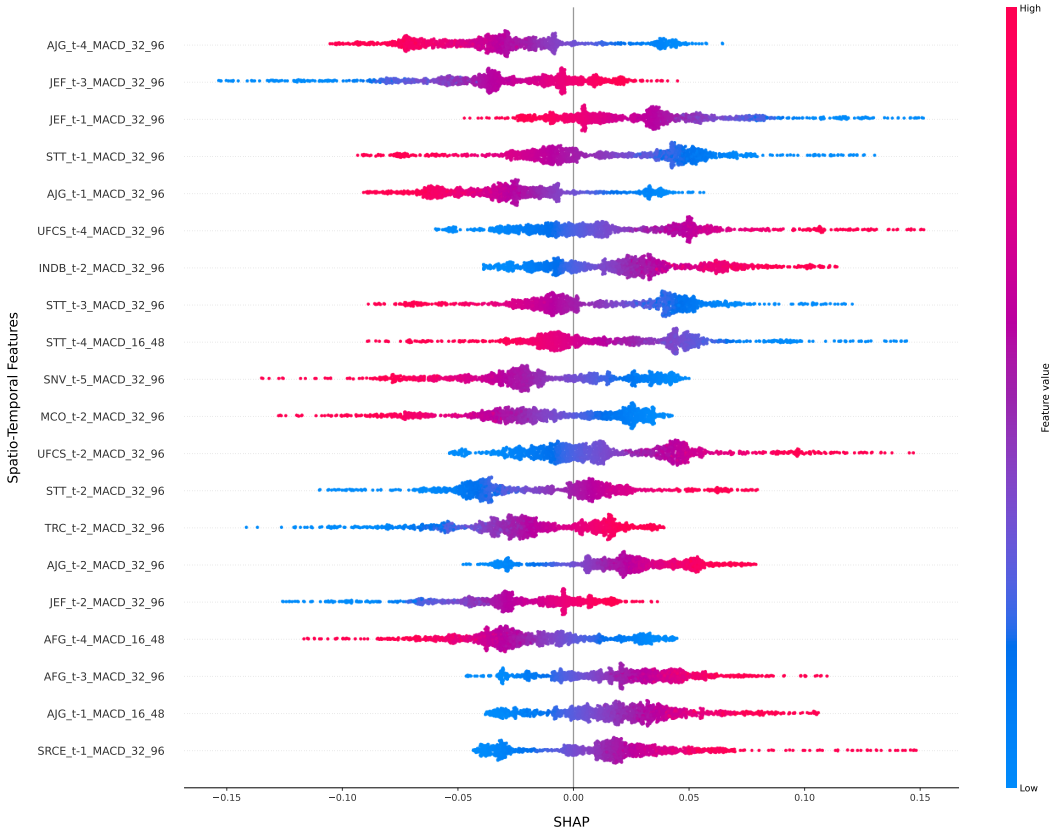

Figure 5: Impact on Predicted Signal Output for a Single Asset – BAC **(US Equities)**

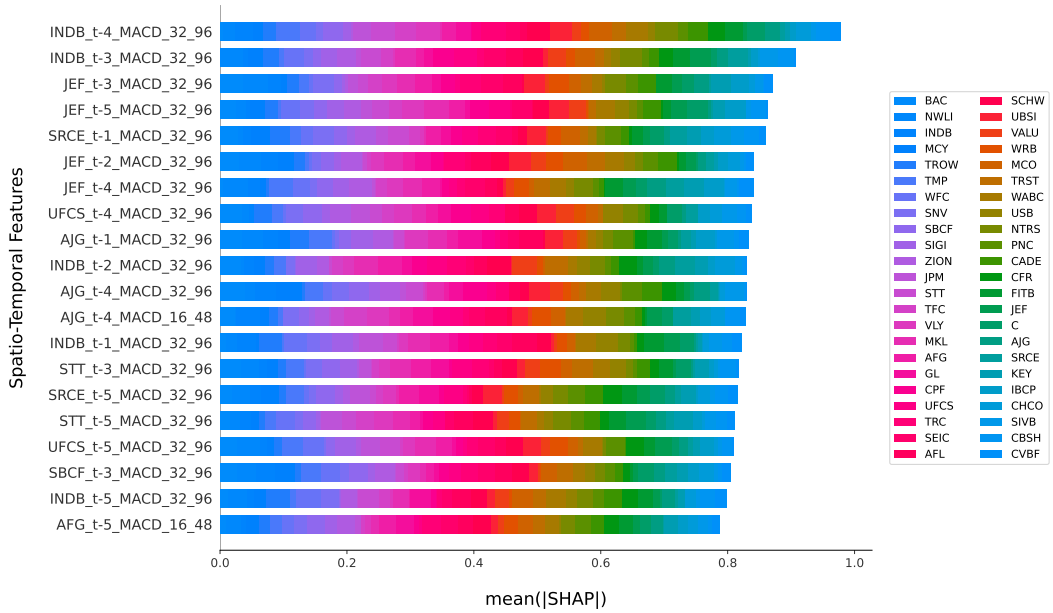

Figure 6: Cumulative Mean Absolute Impact on Predicted Signal Output for All Assets **(US Equities)**

6 Conclusions

We introduce Spatio-Temporal Momentum, a class of machine learning models that combine time-series and cross-sectional momentum strategies. This class of momentum strategies simultaneously generates trading signals for a portfolio of assets in a multitask setting, using both time-series and cross-sectional features from all assets in the portfolio. We demonstrate that the SLP, a neural network with a single hidden layer trained with a shrinkage penalty, is able to outperform more complex models in modelling spatio-temporal momentum, demonstrating its effectiveness over benchmarks on two different datasets.

We show that the performance of a spatio-temporal momentum strategy can be superior to a simple blend of time-series and cross-sectional momentum strategies. Exhibiting unstable correlation with other momentum strategies, the spatio-temporal momentum strategy when combined with a time-series momentum strategy like the DMN yields higher risk-adjusted returns, making spatio-temporal momentum a valuable strategy for diversification.

Given various cost scenarios, we examine the impact of transaction costs on the performance of all strategies and show that both the DMN and SLP were able to retain their performance over benchmarks for costs up to 5 to 10 basis points when tested on the US equities data. We incorporate turnover regularization for the DMN and a different localized minibatch turnover regularization for non-recurrent models like the SLP, resulting in the SLP achieving better performance ratios across all transaction cost scenarios.

With the simplicity of the SLP model, it is possible to directly visualize and interpret its weights corresponding to each spatio-temporal momentum feature. We analyze the importance of individual spatio-temporal momentum features using the model-agnostic SHAP method, revealing clear patterns in momentum and mean-reversion effects of individual features for predicting the trading signal of an individual asset, as well as showing global feature importance for all assets.

In future works, we would like to study the performance of spatio-temporal momentum using alternative feature representations. Another extension of this work includes adapting attention-based deep learning architectures [40, 2, 39] and investigating their effectiveness in modelling spatio-temporal momentum.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473*, 2014.

[3] Nick Baltas and Robert Kosowski. Demystifying Time-Series Momentum Strategies: Volatility Estimators, Trading Rules and Pairwise Correlations. *Market Momentum: Theory and Practice, Wiley*, 2020.

[4] Jamil Baz, Nicolas Granger, Campbell R Harvey, Nicolas Le Roux, and Sandy Rattray. Dissecting Investment Strategies in the Cross Section and Time Series. *SSRN 2695101*, 2015.

[5] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[6] Mikolaj Binkowski, Gautier Marti, and Philippe Donnat. Autoregressive Convolutional Neural Networks for Asynchronous Time Series. In *International Conference on Machine Learning*, pages 580–589. PMLR, 2018.

[7] Rich Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997.

[8] The University of Chicago Booth School of Business Center for Research in Security Prices (CRSP). Calculated (or Derived) based on data from CRSP Daily Stock 2022.

[9] Michael Crawshaw. Multi-Task Learning with Deep Neural Networks: A Survey. *arXiv:2009.09796*, 2020.

[10] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-Task Learning for Multiple Language Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, 2015.

[11] Ashish Garg, Christian L Goulding, Campbell R Harvey, and Michele Mazzoleni. Momentum Turning Points. *SSRN 3489539*, 2021.

[12] Campbell R Harvey, Edward Hoyle, Russell Korgaonkar, Sandy Rattray, Matthew Sargaison, and Otto Van Hemert. The Impact of Volatility Targeting. *The Journal of Portfolio Management*, 45(1):14–33, 2018.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

[15] Narasimhan Jegadeesh and Sheridan Titman. Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. *The Journal of Finance*, 48(1):65–91, 1993.

[16] Abby Y Kim, Yiuman Tse, and John K Wald. Time Series Momentum and Volatility Scaling. *Journal of Financial Markets*, 30:103–124, 2016.

[17] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep Learning. *Nature*, 521(7553):436–444, 2015.

[19] Bryan Lim and Stefan Zohren. Time Series Forecasting with Deep Learning: A Survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[20] Bryan Lim, Stefan Zohren, and Stephen Roberts. Enhancing Time-Series Momentum Strategies Using Deep Neural Networks. *The Journal of Financial Data Science*, 1(4):19–38, 2019.

[21] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning Multiple Tasks with Multilinear Relationship Networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[22] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[23] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-Task Sequence to Sequence Learning. *arXiv:1511.06114*, 2015.

[24] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-Stitch Networks for Multi-task Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[25] Tobias J Moskowitz, Yao Hua Ooi, and Lasse Heje Pedersen. Time Series Momentum. *Journal of Financial Economics*, 104(2):228–250, 2012.

[26] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A Generative Model for Raw Audio. *arXiv:1609.03499*, 2016.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Pinnacle Data Corp. CLC Database. `https://pinnacledata2.com/clc.html`.

[29] Craig Pirrong. Momentum in Futures Markets. *SSRN 671841*, 2005.

[30] Daniel Poh, Bryan Lim, Stefan Zohren, and Stephen Roberts. Building Cross-Sectional Systematic Strategies By Learning to Rank. *The Journal of Financial Data Science*, 3(2):70–86, 2021.

[31] Daniel Poh, Bryan Lim, Stefan Zohren, and Stephen Roberts. Enhancing Cross-Sectional Currency Strategies by Context-Aware Learning to Rank with Self-Attention. *The Journal of Financial Data Science*, 4(3):89–107, 2022.

[32] Marek Rei. Semi-supervised Multitask Learning for Sequence Labeling. *arXiv:1704.07156*, 2017.

[33] K Geert Rouwenhorst. International Momentum Strategies. *The Journal of Finance*, 53(1):267–284, 1998.

[34] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv:1706.05098*, 2017.

[35] William F Sharpe. The Sharpe Ratio. *The Journal of Portfolio Management*, 21(1):49–58, 1994.

[36] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[37] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, 2014.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.

[40] Kieran Wood, Sven Giegerich, Stephen Roberts, and Stefan Zohren. Trading with the Momentum Transformer: An Intelligent and Interpretable Architecture. *arXiv:2112.08534, Risk*, 2023.

[41] Kieran Wood, Stephen Roberts, and Stefan Zohren. Slow Momentum with Fast Reversion: A Trading Strategy Using Deep Learning and Changepoint Detection. *The Journal of Financial Data Science*, 4(1):111–129, 2022.

[42] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial Landmark Detection by Deep Multi-task Learning. In *European Conference on Computer Vision*, pages 94–108. Springer, 2014.

[43] Zihao Zhang, Stefan Zohren, and Stephen Roberts. DeepLOB: Deep Convolutional Neural Networks for Limit Order Books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.

[44] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep Learning for Portfolio Optimization. *The Journal of Financial Data Science*, 2(4):8–20, 2020.

A Deep Learning Architectures & Optimization

A.1 Deep Learning Architectures

For the CNN and LSTM models, we consider the spatio-temporal tensor $\mathbf{u}_t \in \mathbb{R}^{\tau \times m'}$ with $m' = N^t \cdot d$, representing an input sequence of momentum features of all assets over a temporal history τ.

Convolutional Neural Networks (CNN) We consider a 1-D autoregressive CNN of the following:

$$\mathbf{h}_t = \mathcal{P}\sigma[\mathbf{W}_c^{[2]} * \sigma(\mathbf{W}_c^{[1]} * \mathbf{u}_t + \mathbf{b}_c^{[1]}) + \mathbf{b}_c^{[2]}] \tag{15}$$

$$\mathbf{X}_t = f(\mathbf{u}_t; \boldsymbol{\theta}) = g[\mathbf{W}^{[2]\top}\sigma(\mathbf{W}^{[1]\top}\mathbf{h}_t + \mathbf{b}^{[1]}) + \mathbf{b}^{[2]}] \tag{16}$$

where $\mathbf{W}_c^{[l]}$ represent convolutional kernels with associated bias terms $\mathbf{b}_c^{[l]}$ and activations $\sigma = \tanh$, and $\mathbf{w} * \mathbf{u}$ representing the causal convolution operation between the input sequence $\mathbf{u}$ and kernel $\mathbf{w}$. Additionally, we interface an average pooling layer $\mathcal{P}$ that performs a downsampling step prior to propagating the activations $\mathbf{h}_t$ into an MLP as per Equation (7).

Long Short-term Memory (LSTM) We consider a single layer LSTM model:

$$\boldsymbol{\Gamma}_t^i = \sigma_S(\mathbf{W}_i\mathbf{u}_t + \mathbf{V}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \tag{17}$$

$$\boldsymbol{\Gamma}_t^f = \sigma_S(\mathbf{W}_f\mathbf{u}_t + \mathbf{V}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \tag{18}$$

$$\boldsymbol{\Gamma}_t^o = \sigma_S(\mathbf{W}_o\mathbf{u}_t + \mathbf{V}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \tag{19}$$

$$\tilde{\mathbf{c}}_t = g(\mathbf{W}_c\mathbf{u}_t + \mathbf{V}_c\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{20}$$

$$\mathbf{c}_t = \boldsymbol{\Gamma}_t^i \odot \tilde{\mathbf{c}}_t + \boldsymbol{\Gamma}_t^f \odot \mathbf{c}_{t-1} \tag{21}$$

$$\mathbf{h}_t = \boldsymbol{\Gamma}_t^o \odot g(\mathbf{c}_t) \tag{22}$$

$$\mathbf{X}_t = g(\mathbf{W}_{\text{dist}}\mathbf{h}_t + \mathbf{b}) \tag{23}$$

where $\mathbf{W}_*$, $\mathbf{V}_*$ represent the weights and $\mathbf{b}_*$ the biases of the input, forget and output gates $\boldsymbol{\Gamma}_t^i$, $\boldsymbol{\Gamma}_t^f$, $\boldsymbol{\Gamma}_t^o$ respectively, with activation functions $\sigma_S =$ sigmoid and $g = \tanh$. Subsequently, the LSTM computes the memory cell state $\mathbf{c}_t$, hidden state activation $\mathbf{h}_t$ with $\odot$ representing the Hadamard product. Given the spatio-temporal input of the form $\mathbf{u}_t \in \mathbb{R}^{\tau \times m'}$, the network then maps the hidden state activation $\mathbf{h}_t$ to a sequence of trading signals $\mathbf{X}_t \in [-1, 1]^{\tau \times N^t}$ by a fully connected layer with time-distributed weights $\mathbf{W}_{\text{dist}}$.

A.2 Fixed Parameters & Hyperparameter Optimization

We perform hyperparameter optimization with 100 iterations of random search for all machine learning models, using the search range as shown in Table 8. We initiate early stopping with a patience of 25 epochs for the validation loss. We include regularization via dropout [38] for the reference DMN, as well as the MLP, CNN and LSTM STMOM models.

Table 7: Fixed Parameters for Machine Learning Models

Parameters	**DMN**	**SLP**	**MLP**	**CNN**	**LSTM**
Epochs	100	500	500	500	500
Patience	25	25	25	25	25
Random Search Iterations	100	100	100	100	100
Temporal History	63	5	5	63	63

Table 8: Hyperparameter Search Range for Machine Learning Models

Hyperparameters	**Random Search Grid**
Minibatch Size	32, 64, 128, 256
Dropout Rate	0.1, 0.2, 0.3, 0.4, 0.5
Hidden Layer Size	5, 10, 20, 40, 80, 160
Learning Rate	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}
Max Gradient Norm	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}
L1 Regularisation Weight (α)	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}

B Dataset Details

All individual instruments in our datasets have less than 10% missing data. To reduce the effect of extreme outliers, we winsorize all data to limit values to be within 5 times its 252-day exponentially weighted moving standard deviation from its exponentially weighted moving average.

Equity Index Futures consists of 12 ratio-adjusted continuous equity index futures contracts obtained from the Pinnacle Data Corp CLC Database. We perform backtesting from 2003 to 2020.

Ticker Symbol	**Ticker Description**
SP	S & P 500, day session
YM	Mini Dow Jones ($5.00)
EN	NASDAQ, MINI
ER	RUSSELL 2000, MINI
MD	S&P 400 (Mini electronic)
XU	DOW JONES EUROSTOXX50
XX	DOW JONES STOXX 50
CA	CAC40 INDEX
LX	FTSE 100 INDEX
AX	GERMAN DAX INDEX
HS	HANG SENG
NK	NIKKEI INDEX

US Equities consists of actively-traded US equities with data obtained from the Center for Research in Security Prices (CRSP). We screen for common stocks of domestic US companies listed on the NYSE, AMEX and NASDAQ with market capitalization from Small (300M-2B) to Mega (>200B) using the Stock Screener provided by Nasdaq. We perform a random sample of 46 stocks from the Financials sector. We perform backtesting from 1990 to 2022.

Ticker Symbol	Ticker Description
AFG	AMERICAN FINANCIAL GROUP INC NEW
AFL	AFLAC INC
AJG	GALLAGHER ARTHUR J & CO
BAC	BANK OF AMERICA CORP
C	CITIGROUP INC
CADE	CADENCE BANK
CBSH	COMMERCE BANCSHARES INC
CFR	CULLEN FROST BANKERS INC
CHCO	CITY HOLDING CO
CPF	CENTRAL PACIFIC FINANCIAL CORP
CVBF	C V B FINANCIAL CORP
FITB	FIFTH THIRD BANCORP
GL	GLOBE LIFE INC
IBCP	INDEPENDENT BANK CORP MICH
INDB	INDEPENDENT BANK CORP MA
JEF	JEFFERIES FINANCIAL GROUP INC
JPM	JPMORGAN CHASE & CO
KEY	KEYCORP NEW
MCO	MOODYS CORP
MCY	MERCURY GENERAL CORP NEW
MKL	MARKEL CORP
NTRS	NORTHERN TRUST CORP
NWLI	NATIONAL WESTERN LIFE GROUP INC
PNC	P N C FINANCIAL SERVICES GRP INC
SBCF	SEACOAST BANKING CORP FLA
SCHW	SCHWAB CHARLES CORP NEW
SEIC	S E I INVESTMENTS COMPANY
SIGI	SELECTIVE INSURANCE GROUP INC
SIVB	S V B FINANCIAL GROUP
SNV	SYNOVUS FINANCIAL CORP
SRCE	1ST SOURCE CORP
STT	STATE STREET CORP
TFC	TRUIST FINANCIAL CORP
TMP	TOMPKINS FINANCIAL CORP
TRC	TEJON RANCH CO
TROW	T ROWE PRICE GROUP INC
TRST	TRUSTCO BANK CORP NY
UBSI	UNITED BANKSHARES INC
UFCS	UNITED FIRE GROUP INC
USB	U S BANCORP DEL
VALU	VALUE LINE INC
VLY	VALLEY NATIONAL BANCORP
WABC	WESTAMERICA BANCORPORATION
WFC	WELLS FARGO & CO NEW
WRB	BERKLEY W R CORP
ZION	ZIONS BANCORPORATION N A