MEANT: Multimodal Encoder for Antecedent Information

Anonymous ACL submission

Abstract

The stock market provides a rich well of information that can be split across modalities, making it an ideal candidate for multimodal evaluation. Multimodal data plays an increasingly important role in the development of machine learning and has shown to positively impact performance. But information can do more than exist across modes— it can exist across time. How should we attend to temporal data that consists of multiple information types? This work introduces (i) the MEANT model, a Multimodal Encoder for Antecedent information and (ii) a new dataset called *TempStock*, which consists of price, Tweets, and graphical data with over a million Tweets from all of the companies in the S&P 500 Index. We find that MEANT improves performance on existing baselines by over 15%, and that the textual information affects performance far more than the visual information on our time-dependent task from our ablation study. [1]

1 Introduction

Recently, multimodal models have garnered serious momentum, with the release of large pretrained architectures such as Microsoft's Kosmos-1 (Huang et al., 2023) and OpenAI's GPT-4 (OpenAI et al., 2023). Their general use has exploded in many domains, such as language and image processing (Lu et al., 2019; Kim et al., 2021; Huang et al., 2023). Particularly interesting to this study is the deployment of multimodal models on time-dependent environments, where recent successes have shown that event driven models processing multiple modalities are far more performant on stock market tasks than previously state of the art (SOTA) algorithms focusing purely on price information (Li et al., 2021; Zhang et al., 2022). Language data from news and social media sources have shown to greatly increase the performance of models for price prediction (Li et al., 2021; Zhang et al., 2022; Bybee et al., 2023; Mittermayer and Knolmayer, 2006; Xu and Cohen, 2018). However, these approaches typically lack attention components specifically designed to process inputs with sequential, time-dependent information (Li et al., 2021; Sun et al., 2017; Zhang et al., 2022; Xu and Cohen, 2018). This type of data is particularly important when making predictions about stock prices or market movements, as price prediction is a time series task (Zhang et al., 2022; Xu and Cohen, 2018).

In this work, we introduce MEANT, a multimodal model architecture with a novel, temporally focused self-attention mechanism. We extract image features using the TimeSFormer architecture (Bertasius et al., 2021) to find relationships in longer range information (i.e a graph of stock prices over a month), while extracting language features from social media information to pick up more immediate trends (e.g.: Tweets pertaining to stock prices over a five day period). Furthermore, we release *TempStock*, a multimodal stock-market dataset that is designed to be sequentially processed in chunks of varying lag periods.

2 Related Work

Multimodal Models for Financial Twitter Data Several studies have employed natural language processing (NLP) techniques to financial markets, giving birth to the field of natural language-based financial forecasting (NLFF). Many of these studies have focused on public news (Ashtiani and Raahemi, 2023; Bybee et al., 2023). However, social media presents more time-sensitive information from active investors. Thus, for short term analysis, many researchers have begun to focus on Tweets for feature extraction (Araci, 2019; Wu et al., 2018), through which some have combined NLP techniques with traditional analysis on price data (Huang et al., 2022) . Since Tweets often cor-

[1] The code and dataset will be made available upon publication.

respond to events as they happen in real time, such data is better suited for smaller windows (Xu and Cohen, 2018; Zhang et al., 2022). When working with stock market data, combining the features extracted through Natural Language Processing (NLP) methods with price data has shown promising results (Li et al., 2021; Zhang et al., 2022; Xu and Cohen, 2018). However, it is ineffective to feed the concatenated information to the model without encoding temporal dependencies (Li et al., 2021).

Modeling media-aware stock movements is essentially a binary classification problem. Many traditional machine learning methods have been deployed to solve it, including SVMs and Bayesian classifiers (Huang et al., 2012; Wang, 2003; Zuo et al., 2012). More recently, researchers have applied deep learning to the problem. Huang et al. (2016) used a convolutional neural network to explore the impact of Tweets on the stock market. Sun et al. (2017) and Selvin et al. (2017) then employed a recurrent architecture, specifically an LSTM, to extract relevant sentiments from Twitter data for stock market analysis, making their model multimodal, as it could handel Tweets and price information. Li et al. (2021) built atop this architecture, employing different tensor representations for their LSTM input to create more meaningful relationships between the price and Tweets data.

Xu and Cohen (2018) introduced StockNet, a large generative architecture built atop generative architectures, particularly the Variational Auto Encoder (VAE). StockNet represented the first deep generative model for stock market prediction (Xu and Cohen, 2018). TEANet, the most relevant work to our own, similarly used an LSTM to process their final output, but used a BERT-style transformer to extract relevant features from the Tweets (Zhang et al., 2022). TEANet is a language model equipped to handle lag periods similarly to MEANT. They concatenate their language features to price data as an input for an LSTM and a subsequent softmax temporal encoding. We abandon recurrence altogether, developing a novel temporal mechanism, entirely based upon traditional self-attention methods (Vaswani et al., 2017). The temporal processing in TEANet consists of concatenation methods similar to our own, but they do not employ attention over time. Furthermore, their model was built to handle Tweets and price inputs alone. MEANT can handle images as well, employing a dual encoder architecture similar to that of Su et al. (2023).

Financial Twitter Datasets Previous financial datasets have shown the power of Twitter data for financial analysis (Pei et al., 2022; Araci, 2019; Li et al., 2021). Twitter is powerful in its ability to generate real time information about the market before traditional newswires (Pei et al., 2022). Souza et al. (2015) focused on Twitter as a resource for examine financial dynamics in the retail sector. Pei et al. (2022) introduced TweetsFinSent, a large corpus specifically for sentiment analysis. Sun et al. (2017) introduced a dataset consisting of Tweets and prices, where the Tweets information served as a sentiment analysis accompaniment for the price data. Xu and Cohen (2018) introduced the StockNet-dataset, consisting of Tweets and price information for a selection of 88 companies over a two year period from 01/01/2014 to 01/01/2016. Mao et al. (2012) matched Tweets with price information from companies in the S&P 500 dataset, which is the most similar to the TempStock dataset that we introduce below.

3 TempStock Dataset

We collected a new dataset containing 1,755,998 Tweets and price information from all of the companies in the S&P 500 from 4/10/2022 to 4/10/2023.

From the price information, we calculated the Moving Average Convergence-Divergence (MACD) (Appel, 2005) for each company over a year. The MACD is built on the back of Exponential Moving Average (EMA) (Brown, 1964). The EMA is defined as follows:

$$EMA_t = (1 - \alpha) \cdot EMA_{t-1} + \alpha \cdot y_t$$

where t represents the day of EMA and y_t represents the closing price on that day, or in the case of the signal line, the MACD value on that day. α represents the degree of decrease, where $\alpha = \frac{2}{t+1}$. The MACD consists of (i) an MACD line, which is the difference between the fast EMA and the slow EMA (commonly set to 12 days and 26 days respectively), (ii) a signal line, which is the EMA of the MACD line itself (usually over a 9 day period), and (iii) a histogram, which is the difference between the MACD and the signal line. The MACD indicator was chosen [2] because it has been shown to perform well against other indicators in terms

[2] For more on this, see 6.3

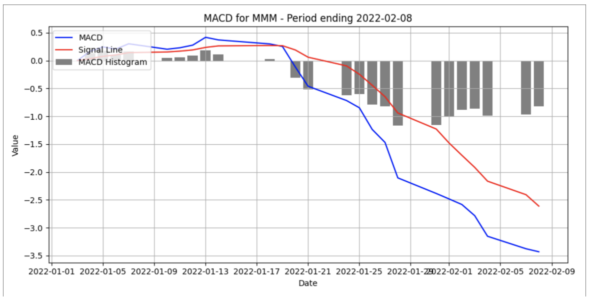

Figure 1: An example of a graph from our MACD data, which displays the MACD (in blue) and the signal line (in red) for MMM (3M) over a 26 day period. Along the x-axis, we see 11 of the dates listed, and the the y-axis shows the value of the aforementioned indicators. In each bar lies the value of the MACD histogram, which is the difference between the MACD (blue) and the Signal line (red).

Description	**Count**
Total Tweets	1,755,998
Total MACD Values	122,959

Table 1: TempStock-large Raw Numbers

of making accurate assertions about price directions (Appel, 2005; Chio, 2022). From our MACD data, we created graphs of the MACD indicator and the corresponding signal line over 26 day periods, which served as our image inputs to the MEANT model. A example of the graph inputs can be seen in Figure 1.

The MACD of each ticker in the subset was taken over a year period, along with the Tweets mentioning that company for each day in that period. The MACD information was gathered using the Yahoo-Finance API (Finance, 2024), and the Tweets were scraped using the snscraper (JustAnotherArchivist, 2021) in April 2023.

TempStock contains Tweets, graphs, and MACD. Each input is arranged into five day lag periods leading up to target day t, consisting of five MACD vectors,

$$M = [M_{t-5}, M_{t-4}, M_{t-3}, M_{t-2}, M_{t-1}]$$

five days of Tweets,

$$X = [X_{t-5}, X_{t-4}, X_{t-3}, X_{t-2}, X_{t-1}]$$

and five images containing graphs of the MACD indicator over 26 days.

$$G = [G_{t-5}, G_{t-4}, G_{t-3}, G_{t-2}, G_{t-1}]$$

For the Tweets stored daily, there were a variable amount for each ticker. We concatenated all available Tweets with [SEP] tokens in between each Tweet. These concatenations were then stored for each day in the lag period, which produced great informational variation across tickers and across days. Each MACD vector M_i contains the EMA_{12}, EMA_{26}, Signal line s_i, MACD histogram h_i, and MACD value m_i for that day.

$$M_{t-i=5,...,1} = [EMA^i_{12}, EMA^i_{26}, s_i, h_i, m_i]$$

In order to separate the dataset into positive and negative signals, we chose to use the MACD signal cross strategy (Appel, 2005). Data points were classified as *positive* if the MACD value on our preceding day to target day t, m_{t-1}, was below the Signal s_{t-1}, and if the MACD on our target day m_t was above our Signal s_t.

$$m_{t-1} < m_{t-1} \wedge m_t > s_t$$

Adversely, data points were classified as *negative* if m_{t-1} was above the Signal s_{t-1}, and if m_t was below s_t

$$m_{t-1} > s_{t-1} \wedge m_t < s_t$$

The lag periods which did not fall in either of these cases were removed, along with the lag periods in which there was insufficient tweet information. This resulted in 92.57% of the lag periods being thrown out, with the exclusion of 41 tickers from the S&P500 all together. For more specifics on the tickers that were included, and to what extent they contributed to TempStock in its final form, please see A.6. The resulting dataset was surprisingly balanced, with no augmentation or oversampling required. These stocks experience similar degrees of up-trends and downtrends in the time period according to the MACD rule employed above, illustrating their stability in a good market climate (Goetzmann and Massa, 2003).

Category	**Count**	**Proportion**
Positive	4,221	51.36%
Negative	3,997	48.64%
Total	8,218	

Table 2: TempStock splits

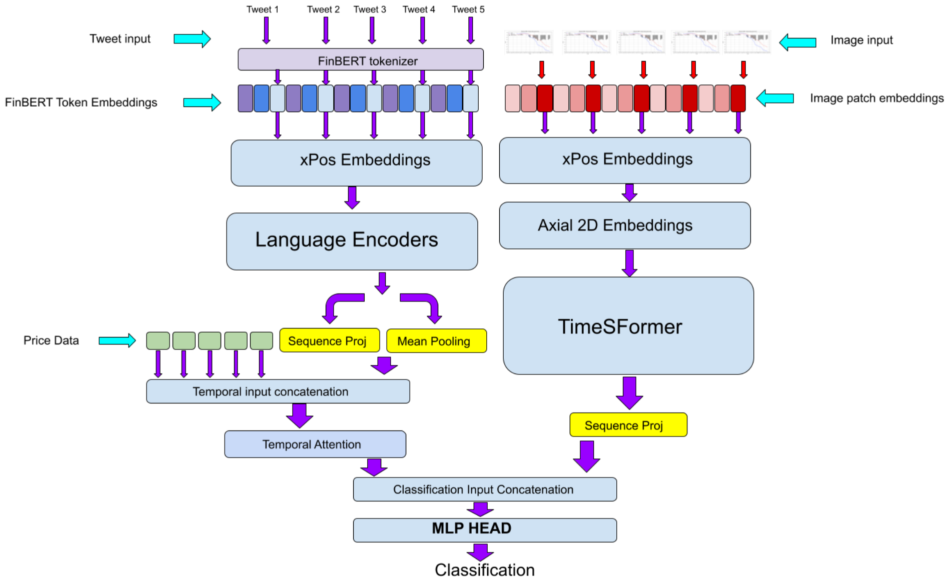

Figure 2: A schematic overview of the MEANT architecture. As seen in the diagram, the output of the language encoder is processed in two different variants: sequence projection, and mean pooling.

4 MEANT

MEANT combines the advantages of image and language processing with temporal attention, in order to extract dependencies from multimodal, sequential information, where 2 displays the full architecture. MEANT, similarly to most SOTA multimodal models (Liang et al., 2021; Kim et al., 2021; Su et al., 2019; Huang et al., 2023; OpenAI et al., 2023), is built atop the Transformer architecture (Vaswani et al., 2017).

4.1 Encoder Only

MEANT is an encoder-only model, similar to BERT (Devlin et al., 2018). Our model contains two pipelines, an image and a language pipeline. The language encoder stacks the attention mechanism with linear layers to extract relevant features from the input. Between the 2 parts of the encoder, and before the output, there is a standard residual connection, meaning that the input to that portion of the architecture is fed through added with the original input. This is done to alleviate the vanishing gradient problem (Pascanu et al., 2013). The interleaved encoder structure is employed by the language pipeline is inspired by the Magneto model (Wang et al., 2022). It makes use of sublayer normalization, meaning that a layer norm is interleaved between the attention and linear layer components of the encoder. This architecture was chosen because it has been shown to be successful on a wide variety of uni-modal and multimodal problems (Huang et al., 2023; Wang et al., 2022).

For the backbone of our image pipeline, we chose to use a variant of the TimeSFormer model (Bertasius et al., 2021), which is an encoder model designed to handle video inputs. We chose this model because of its ability to extract dependencies in the temporal dimension. Our lag graph inputs change in place in a similar manner to a video. We altered the implementation to make use of the interleaved layernorm strategy from Magneto, and used different positional embeddings. In earlier iterations of the model, we used ViT encoders, and fed the outputs of our image pipeline to our temporal attention mechanism along with our Tweets. We found this to be less performant (see 6).

4.2 Token and Patch Embeddings

Before being fed to the attention mechanism, the two input types have to be prepared for processing using two different embedding strategies. The Tweets in MEANT are tokenized using the FinBERT tokenizer (Araci, 2019) and we use the FinBERT pretrained word embedding layer.

The images are first transformed into tensors of rgb values and reshaped to a manageable size. MEANT handles input image sizes of 3 x 224 x 224, where 3 represents the number of channels and the subsequent dimensions are the height and width respectively. TimeSFormer breaks down the vectors using the patch embedding strategy from the original vision transformer (Dosovitskiy et al., 2020) (Bertasius et al., 2021).

4.3 Positional Encoding

In MEANT, the language and vision encoders use different variants of the rotary embedding (Su et al., 2021). The language encoder uses the $xPos$ embeddings (Sun et al., 2022), while the TimeSFormer uses both rotary and axial 2-D embeddings (Su et al., 2021). In axial 2-D embeddings, the angle θ of rotation is altered according to the following equation:

$$\theta_i = i * floor(d/2) * pi$$

4.3.1 Temporal Encoder

We developed two different variants of our temporal encoding pipeline, which work better in different cases: temporal attention with mean pooling, and temporal attention with sequence projection.

In both cases, the outputs of our language encoders L_{out} are tensors of the shape $b \times l \times s \times d_l$, where b denotes the batch size, l denotes the lag period, s is the sequence length, and d_l is the dimension of each encoded language token. For temporal attention with mean pooling, we use mean pooling along the s dimension:

$$L_{seq} = mp(L_{out}) = \frac{1}{s}\sum_{i=1}^{s} L_{out}[:,:,i,:] \tag{1}$$

For temporal attention with sequence projection, we use a parameterized projection matrix to reduce L_{out} along the s dimension:

$$L_{seq} = sp(L_{out}) = \text{GELU}(\text{layNm}(W_{sl}(L_{out}^T)+b_{sl})) \tag{2}$$

$W_{sl} \in \mathbb{R}^{s\times 1}$ represents our reduction weights for the language encoding. Essentially, we are extracting a latent representation for each lag day using a single layer coupled with a non-linear layer. The benefit of this is that each lag day comes to represent a token in the sequence for the attention mechanism to process.

Both of these strategies have different trade-offs, which we discuss in section 6.3 and illustrate in our ablation study (see section 7). Figure 2 indicates where the two variations are employed to the language encoding output.

In both cases, L_{seq} has the shape $b \times l \times d_l$. To emphasize, these are the alternate formulations for the same step:

$$L_{seq} = \begin{cases} mp(L_{out}) & \text{(mean pooling)} \\ sp(L_{out}) & \text{(sequence projection)} \end{cases} \tag{3}$$

We then concatenate our L_{seq} outputs to our MACD information M from that five day lag period:

$$T = [L_{seq}, M] \in \mathbb{R}^{l\times d_t} \tag{4}$$

Where $T = [T_{t-5}, T_{t-4}, T_{t-3}, T_{t-2}, T_{t-1}]$. T has the shape $b \times l \times d_T$, where d_T is the temporal dimension, which is the sum of d_l, and MACD price length, which is 5. T signifies our inputs for the temporal encoder. In the vanilla implementation of the MEANT model, the temporal dimension is 773.

We then pass our outputs T to the temporal attention mechanism. At this point in the pipeline, relevant text features have been extracted for each trading day in relation to themselves, not to one another. The temporal attention mechanism focuses on the day before our target, or the last day in our lag period input, and its relationship to the preceding days. We put extra emphasis on this final day because its when the stock has its final movement. The inspiration for this increased focus on the final day in the lag period comes from previous work (Zhang et al., 2022)(Xu and Cohen, 2018), which rely on simpler, softmax-focused informational dependencies between auxiliary trading days. At inference time, we would want our model to extract a pattern from the preceding days, and act on current day with a sense of what will happen the next day, choosing to produce a buy or sell recommendation. MEANT does this by using a strategy we call *Query-Targeting*, in which the query matrix of the attention mechanism is produced from the target day alone. To produce our query-targeted matrix Q_t, we first extract the pre-target day vectors from our T input, T_{t-1}, which are of the shape $b \times 1 \times d_T$. We then multiply our learned q matrix by this value to produce Q_t.

$$Q_t = dot(T_{t-1}, q) \tag{5}$$

The key and value matrices, K and V, are calculated normally over all of T. The attention computation then proceeds normally with our Q_t, K, and V matrices.

$$T_{lang} = tempAtten(Q_t, K, V) = softmax\left(\frac{Q_t K^T}{\sqrt{d}}\right)V \tag{6}$$

$tempAtten(Q_t, K, V)$ results in out temporal language output T_{lang}, which has found the temporal dependencies between our Tweets and prices in tandem.

TimeSFormer uses a separate strategy to extract the temporal dependencies in our image inputs, called divided space-time attention (T + S) (Bertasius et al., 2021). The following two equations are pulled directly from Bertasius et al. (2021). (T + S) uses the patch embeddings as input, similarly to the ViT (Dosovitskiy et al., 2020). (T + S) first executes its temporal mechanism, where each patch attends to the patch at the same location across all of the frames.

$$\alpha^{(\ell,a)\text{time}}_{(p,t)} = \text{SM}\left(\frac{\mathbf{q}^{(\ell,a)}_{(p,t)}}{\sqrt{D_h}} \cdot \left[\mathbf{k}^{(\ell,a)}_{(0,0)} \left\{\mathbf{k}^{(\ell,a)}_{(p,t')}\right\}_{t'=1,\ldots,F}\right]\right) \tag{7}$$

In the original paper, ℓ denotes the encoder block, a refers to the attention head, p is the patch, and t is the current frame. $\alpha^{(\ell,a)\text{time}}_{(p,t)}$ is then fed back into the spatial attention mechanisms, which executes the attention computation for each patch in relation to the other patches in its same frame, similarly to Dosovitskiy et al. (2020).

$$\alpha^{(\ell,a)\text{space}}_{(p,t)} = \text{SM}\left(\frac{\mathbf{q}^{(\ell,a)}_{(p,t)}}{\sqrt{D_h}} \cdot \left[\mathbf{k}^{(\ell,a)}_{(0,0)} \left\{\mathbf{k}^{(\ell,a)}_{(p',t)}\right\}_{p'=1,\ldots,N}\right]\right) \tag{8}$$

$\alpha^{(\ell,a)\text{space}}_{(p,t)}$ is then fed through a feed-forward network FF and added to a residual to produce our encoded image output I_{out}.

$$I_{out} = FF(\alpha^{(\ell,a)\text{space}}_{(p,t)}) + \alpha^{(\ell,a)\text{space}}_{(p,t)} \tag{9}$$

Our output I_{out} will have the shape $b \times p \times d_p$, where p is the number of patches, and d_p is the dimension of each patch. Similarly to how we *preprocess* the outputs of our language encoder L_{out} before temporal encoding, we now *postprocess* our image output I_{out} to extract the our temporal representation akin to the class token, using a sequence projection.

$$T_{img} = GELU(layerNorm(W_{sp}(I^T_{out}) + b_{sp})) \tag{10}$$

W_{sp} represents our reduction weights for the pixel encoding. We do not use mean pooling for image outputs in any variant of MEANT-base. However, we did train a ViT variant of MEANT in which we experimented with mean pooling and sequence projection for the image output. See sections 6 and 7.

To produce our final classification output, we concatenate our temporal representations into one vector T_{final}.

$$T_{final} = [T_{lang}, T_{img}] \tag{11}$$

WE then pass T_{final} to our MLP head to produce a classification y.

$$y = MLP(T_{final}) \tag{12}$$

5 Experiments

We ran the model at three different sizes, coined MEANT-small, MEANT-large and MEANT-XL. MEANT-small contained one encoder for language and vision, along with one temporal encoder. MEANT-large consisted of twelve language and vision encoders, and one encoder for temporal attention. twelve was selected as the number of encoders used in the original BERT model (Devlin et al., 2018). MEANT-XL had 24 encoders in our language pipeline and our TimeSFormer backbone, along with one temporal encoder. Implementation details can be seen in A.4.

Model	Parameter Count
MEANT-base	48,304,272
MEANT-large	152,367,264
MEANT-XL	265,890,528

Table 3: MEANT Parameter Count

5.1 Fine-tuning on downstream tasks

We tested the viability of the MEANT architecture on two tasks.

5.1.1 TempStock

TempStock is a binary classification task, identifying lag periods which resulted in momentum shifts and those that did not. To further measure MEANT's performance, we ran some similar SOTA encoder-based multimodal models on TempStock. TEANet, a key inspiration for this work, was the most similar model in original purpose, so proved the most interesting benchmark. For more details on the baselines, experiment setup, input differences and model sizes please see A.3 and 10.

5.1.2 Stocknet

The most similar dataset to TempStock was the Stocknet dataset (Xu and Cohen, 2018), which consists of Tweets and price values from a selected batch of stock tickers. Stocknet is different from TempStock as it is a unimodal dataset, containing no graphical component, and is furthermore focused on binary price change rather than momentum shift (as measured by MACD crossing in TempStock). Nonetheless, Stocknet represents one of the only datasets to our knowledge organized in lag periods and is therefore relevant as a benchmark for the MEANT model.

Since the StockNet dataset does not have a visual input, we implemented a MEANT model without the visual capabilities called MEANT-Tweet-price. We ran MEANT-Tweet-price against TEANet (Zhang et al., 2022) which was originally evaluated by the authors on the StockNet dataset, as well as the StockNet model itself (Xu and Cohen, 2018). Details about the StockNet task, baselines used, training settings, input differences can be found in A.3.2, 9 and 10.

6 Results

Tables 4 and 5 in sections 6.1 and 6.2 show the results for our experiments respectively.

6.1 TempStockLarge Experiment results

Observing 4, we can see that MEANT-XL outperformed all other models. MEANT-large performed comparably, coming in second for all three of those categories. The MEANT results in 4 use sequence projection, which performed better in this task (see 7).

Interestingly, TEANet outperformed MEANT-base. TEANet was followed closely by the LSTM baseline, which due to TEANet being built atop an LSTM backbone (Zhang et al., 2022), and that the LSTM takes advantage of temporal information (the MACD values m_{t-i} over all of the lag days). The MLP baseline outperforms all other BERT-based models. This illustrates the importance of the price information (further confirmed in 6) and attention without *Query-Targeting* does not perform well.

ViLT outperforms VL-BERT with and without the price modification. ViLT has a more similar encoding structure to MEANT, taking advantage of the patch embedding strategy, which is likely one reason for its performance advantage over VL-BERT. Since both of VL-BERT and ViLT are not designed to process lag periods, the models were at a severe disadvantage in terms of extracting temporal dependencies in the information they were given.

Model	**F1**	**P**	**R**
FinBERT	0.5047	0.5047	0.5047
BERT	0.5321	0.5300	0.5318
VL-BERT	0.3415	0.2593	0.5000
VL-BERT-price	0.3249	0.2407	0.5000
ViLT	0.5483	0.5554	0.5524
ViLT-price	0.6813	0.6814	0.6816
TimeSFormer	0.3415	0.2593	0.5000
MLP	0.7124	0.7145	0.7122
LSTM	0.7623	0.7622	0.7623
TEANet	0.7898	0.8198	0.7979
MEANT-base	0.7815	0.7917	0.7812
MEANT-large	0.8351	0.8399	0.8343
MEANT-XL	0.**8440**	**0.8497**	**0.8430**

Table 4: TempStock-Large Experiment Results, using Precision (P), Recall (R), and F-1 scores.

For a more in depth examination of how each modality affected performance, see A.1.

6.2 Stocknet results

Model	**Acc%**	**F1**	**P**	**R**
MLP	50.17	0.49	0.50	0.50
LSTM	54.76	0.47	0.59	0.54
FinBERT	46.17	0.29	0.21	0.50
BERTweet	49.20	0.32	0.24	0.50
StockNet	57.53	0.57	0.58	0.57
TEANet	67.75	0.68	0.67	0.68
M-Tweet-price-base	79.92	0.79	0.80	0.79
M-Tweet-price-large	81.35	0.81	0.81	0.81
M-Tweet-price-XL	**82.15**	**0.82**	**0.82**	**0.8211**

Table 5: StockNet-dataset experiment results using Precision (P), Recall (R), F-1 scores and testing accuracy (Acc).

Looking at 5, MEANT-Tweet base and MEANT-Tweet-large, both using mean pooling, outperform all other models by a significant amount on the StockNet task. MEANT-tweet-XL outperformed TEANet, the previous SOTA on the StockNet dataset, by 15%. We ran our own implementation of the TEANet model on the task following their descriptions from the paper, as we could not find publicly available code (see A.4). The original accuracy score reported in the paper was 65.16% (Zhang et al., 2022).

The importance of a temporal component for the StockNet task is clear. BERTweet, a typical encoder architecture without temporal support, performed abysmally. StockNet performed marginally better, but it is with the auxiliary temporal softmax mechanism in TEANet that the first true performance gain can be seen. In these runs, our mean pooling mechanism was more effective than the sequence projection strategy for our temporal encoding (see section 8).

Clearly, the attention-based temporal mechanism in MEANT is the most performant for this problem. *Query-Targeting* is able to extract meaningful relationships between the target day and the auxiliary trading days more effectively than previous mechanisms. There are likely a few reasons for this. Models that depend on multi-head selt-attention (MSA) can be thought of as a low pass filters, meaning that they generally tend to flatten loss landscapes (Park and Kim, 2022). There are Tweets in the StockNet dataset that don't correlate to the buy signal, but because of the nature of the data collection, these are in the vast minority (Xu and Cohen, 2018). However, since we are also extracting trends that are dependent on the order of these Tweets in time, a succession of even a few outlier or irrelevant Tweets could be very damaging to the loss landscape of a more sensitive model. Our temporal attention mechanism is better able to handle the noise in the data. Furthermore, attention scales far better with parameter size, and our MEANT-XL model in particular dwarfs previous TEANet and StockNet in parameter size (Zhang et al., 2022; Xu and Cohen, 2018). Larger parameter spaces tend to lead to a more nuanced loss landscape (Fort and Jastrzebski, 2019; Fort and Scherlis, 2019; Park and Kim, 2022).

6.3 Limitations

Here, we outline considerations, trade-offs and design decisions we have made:

- **Dataset** To explore temporal information processing, we chose momentum buy signals in stock market data. We went with the MACD indicator because of its robustness, and correlation to strong positive returns against other indicators (Joshi, 2022; Chio, 2022). The serious drawback in this choice is in the infrequency of buy and sell signals that occur, which leads to a less robust dataset.

  We gathered our stock price information from companies in the S&P 500. We chose this index because of its stability. However, as a result, we were unable to train our model on more extreme price patterns that are more common on obscure indexes (Goetzmann and Massa, 2003). Thus, in the case of extreme market events that result in periods of steep decline or rise would likely confuse the model.

- **MEANT** The MEANT encoder is built atop the Kosmos-1 encoder architecture, that uses interleaved LayerNorms (Vu et al., 2022). The authors thought this to lead to increased numeric stability (Huang et al., 2023), which in turn helps prevent the exploding gradient problem. However, the inclusion of so many layerNorms in each encoder in our models can lead to an increase in bias, which eventually can lead to a serious overfitting problem (Xu et al., 2019). We chose to go ahead with this risk, as previous architectures have shown the stability gains from the interleaved normalizations to allow for better scaling (Wang et al., 2022; Huang et al., 2023).

  MEANT was trained to identify buy signals and sell signals, instead of trying to classify price periods on a more nuanced scale. We chose this path for simplicity's sake. For practical use on financial data, we would likely need more levels of categorization.

7 Conclusion and Future Work

We introduced a multimodal encoder with a novel temporal component comprised entirely of self-attention. MEANT outperforms previous models on the StockNet benchmark by 15%, and proves to be the most performant model on our own TempStock benchmark. To our knowledge, MEANT-XL is the largest model to be applied to StockNet, and is the first multimodal model to contain an attention mechanism to deal with data over a lag period of days. MEANT combines the realms of language, vision, and time to produce SOTA results. We would like to explore different early fusion methods in order to make MEANT more robust against other common multi modal benchmarks, and expand upon our *Query-Targeting* strategy to emphasize relevant queries automatically, rather manually emphasizing any specific component such as the final lag day. We believe that the MEANT architecture has the potential to succeed on a wide variety of tasks. Furthermore, the image space that we trained MEANT on was limited. We would like to introduce more variation into our image inputs, to fully utilize the capabilities of that modality in our model.

8 Ethics Statement

Bias and Data Privacy: We acknowledge that there are biases in our study, including limiting our work to a specific time period, a small sample of securities and the general public, where we cannot verify they financial expertise in assessing markets. The data collected in this work will only be made available via Tweet IDs collected to protect X's users rights to remove, withdraw or delete their content. All datasets and Language Models are publicly available and were used under the license category that allows use for academic research.

Reproducibility: We make all of our code publicly available upon publication on Github, where we provide instructions to reproduce our results.

Use case: We strongly advise against the use of our proposed model and dataset for financial decision making, including automated or high frequency trading.

References

G. Appel. 2005. *Technical Analysis: Power Tools for Active Investors*. Financial Times Prentice Hall books. Financial Times/Prentice Hall.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models.

Mehdi Arjmand, Mohammad Javad Dousti, and Hadi Moradi. 2021. TEASEL: A transformer-based speech-prefixed language model. *CoRR*, abs/2109.05522.

Matin N. Ashtiani and Bijan Raahemi. 2023. News-based intelligent prediction of financial markets using text mining and machine learning: A systematic literature review. *Expert Systems with Applications*, 217:119509.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding? *CoRR*, abs/2102.05095.

Robert Goodell Brown. 1964. Smoothing, forecasting and prediction of discrete time series.

Leland Bybee, Bryan Kelly, and Yinan Su. 2023. Narrative Asset Pricing: Interpretable Systematic Risk Factors from News Text. *The Review of Financial Studies*, 36(12):4759–4787.

Pat Tong Chio. 2022. A comparative study of the MACD-base trading strategies: evidence from the US stock market. Papers 2206.12282, arXiv.org.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

Yahoo Finance. 2024. Stock market, finance & business news. https://finance.yahoo.com. Accessed: 2024-05-29.

Stanislav Fort and Stanislaw Jastrzebski. 2019. Large scale structure of neural network loss landscapes. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Stanislav Fort and Adam Scherlis. 2019. The goldilocks zone: towards better understanding of neural network loss landscapes. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Ross B. Girshick. 2015. Fast R-CNN. *CoRR*, abs/1504.08083.

William N. Goetzmann and Massimo Massa. 2003. Index funds and stock market growth. *The Journal of Business*, 76(1):1–28.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Guimin Hu, Ting-En Lin, Yi Zhao, Guangming Lu, Yuchuan Wu, and Yongbin Li. 2022. Unimse: Towards unified multimodal sentiment analysis and emotion recognition.

Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. 2012. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. 2023. Language is not all you need: Aligning perception with language models.

Yifu Huang, Kai Huang, Yang Wang, H. Zhang, Jihong Guan, and Shuigeng Zhou. 2016. Exploiting twitter moods to boost financial trend prediction based on deep network models. In *International Conference on Intelligent Computing*.

Yuxuan Huang, Luiz Capretz, and Danny Ho. 2022. Machine learning for stock prediction based on fundamental analysis.

Dushyant Joshi. 2022. Use of moving average convergence divergence for predicting price movements. *International Research Journal of MMC*, 3:21–25.

JustAnotherArchivist. 2021. snscrape: A social networking service scraper in python. Version 0.34.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*.

Qing Li, Jinghua Tan, Jun Wang, and Hsinchun Chen. 2021. A multimodal event-driven lstm model for stock prediction using online news. *IEEE Transactions on Knowledge and Data Engineering*, 33(10):3323–3337.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Chen, Peter Wu, Michelle A. Lee, Yuke Zhu, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2021. Multibench: Multiscale benchmarks for multimodal representation learning. *CoRR*, abs/2107.07502.

Ilya Loshchilov and Frank Hutter. 2016. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *CoRR*, abs/1908.02265.

Yuexin Mao, Wei Wei, Bing Wang, and Benyuan Liu. 2012. Correlating sp 500 stocks with twitter data. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*, HotSocial '12, page 69–72, New York, NY, USA. Association for Computing Machinery.

Marc-andre Mittermayer and Gerhard F. Knolmayer. 2006. Newscats: A news categorization and trading system. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 1002–1007.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Namuk Park and Songkuk Kim. 2022. How do vision transformers work?

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Yulong Pei, Amarachi Mbakwe, Akshat Gupta, Salwa Alamir, Hanxuan Lin, Xiaomo Liu, and Sameena Shah. 2022. TweetFinSent: A dataset of stock sentiments on Twitter. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, pages 37–47, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Alex Rogozhnikov. 2022. Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*.

Guy D. Rosin and Kira Radinsky. 2022. Temporal attention for language models. *CoRR*, abs/2202.02093.

Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. 2017. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE.

Thársis Tuani Pinto Souza, Olga Kolchyna, Philip C. Treleaven, and Tomaso Aste. 2015. Twitter sentiment analysis applied to finance: A case study in the retail industry. *CoRR*, abs/1507.00784.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. VL-BERT: pre-training of generic visual-linguistic representations. *CoRR*, abs/1908.08530.

Weijie Su, Xizhou Zhu, Chenxin Tao, Lewei Lu, Bin Li, Gao Huang, Yu Qiao, Xiaogang Wang, Jie Zhou, and Jifeng Dai. 2023. Towards all-in-one pre-training via maximizing multi-modal mutual information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15888–15899.

Tong Sun, Jia Wang, Pengfei Zhang, Yu Cao, Benyuan Liu, and Degang Wang. 2017. Predicting stock price returns using microblog sentiment for chinese stock market. In *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, pages 87–96.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. 2022. A length-extrapolatable transformer.

Z Tao, C XiaoYu, L HuiLing, Y XinYu, L YunCan, and Z XiaoMin. 2022. Pooling operations in deep learning: From "invariable" to "variable". *Biomed Res Int*, 2022:4067581.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Doan Nam Long Vu, Nafise Sadat Moosavi, and Steffen Eger. 2022. Layer or representation space: What makes BERT-based evaluation metrics robust? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3401–3411, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Hongyu Wang, Shuming Ma, Shaohan Huang, Li Dong, Wenhui Wang, Zhiliang Peng, Yu Wu, Payal Bajaj, Saksham Singhal, Alon Benhaim, Barun Patra, Zhun Liu, Vishrav Chaudhary, Xia Song, and Furu Wei. 2022. Foundation transformers.

Phil Wang. 2021. Timesformer-pytorch. Last accessed: June 12, 2024.

Yi-Fan Wang. 2003. On-demand forecasting of stock prices using a real-time predictor. *Knowledge and Data Engineering, IEEE Transactions on*, 15:1033–1037.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. 2018. Hybrid deep sequential modeling for social text-driven stock prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1627–1630, New York, NY, USA. Association for Computing Machinery.

Zehui Wu, Ziwei Gong, Jaywon Koo, and Julia Hirschberg. 2024. Multimodal multi-loss fusion network for sentiment analysis.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. 2019. Understanding and improving layer normalization. *CoRR*, abs/1911.07013.

Yumo Xu and Shay B. Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, Melbourne, Australia. Association for Computational Linguistics.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. MOSI: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *CoRR*, abs/1606.06259.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *CoRR*, abs/1910.07467.

Qiuyue Zhang, Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang, and Peide Liu. 2022. Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, 202:117239.

Yi Zuo, Masaaki Harada, Takao Mizuno, and Eisuke Kita. 2012. Bayesian network based prediction algorithm of stock price return. In *Intelligent Decision Technologies*, pages 397–406, Berlin, Heidelberg. Springer Berlin Heidelberg.

A Appendix

A.1 Albation Study

To examine the importance of the image and language modalities respectively, we also created many variations of the MEANT model, to target each modality and different combinations of them. Thus, we had a model for each modality individually, and each combination of the three modalities. MEANT-vision-price and MEANT-Tweet-price, for instance, take in the inputs $x = [G, M]$ and $x = [X, M]$ respectively. All variants were similarly fine-tuned and evaluated on the TempStock task (5.1.1) over 15 epochs, with a training batch size of 16, a starting learning rate of 5e-5, the AdamW optimizer, and a cosine-annealing learning rate scheduler with warm restarts.

MEANT Ablation	F1	P	R
MEANT-base	0.7815	0.7917	0.7812
MEANT-large	0.8351	0.8399	0.8343
MEANT-XL	**0.8440**	**0.8497**	**0.8430**
MEANT-base-pt	0.7712	0.8039	0.7654
MEANT-large-pt	0.8249	0.8272	0.8258
MEANT-XL-pt	0.8312	0.8322	0.8288
MEANT-base-10	0.5731	0.5031	0.5631
MEANT-large-10	0.6294	0.6227	0.6285
MEANT-XL-10	0.6315	0.6321	0.6277
M-Tweet-price	0.7375	0.8168	0.7565
M-Tweet-price-large	0.8305	0.8346	0.8327
M-Tweet-price-XL	0.8337	0.8359	0.8348
MEANT-base-pt	0.7738	0.7542	0.7621
MEANT-large-pt	0.8352	0.8364	0.8279
MEANT-XL-pt	0.8340	0.8297	0.8130
M-Tweet	0.3415	0.2593	0.5000
M-Tweet-Large	0.4213	0.4176	0.5328
M-Tweet-XL	0.5013	0.4776	0.5593
M-vision-price	0.3249	0.2407	0.5000
M-vision-price-large	0.5237	0.3815	0.5769
M-vision-price-XL	0.7104	0.7103	0.7104
M-vision-no-price	0.3415	0.2593	0.5000
M-vision-no-price-l	0.3415	0.2593	0.5000
M-vision-no-price-XL	0.3725	0.3293	0.5784
M-price-large	0.7376	0.7285	0.7479
MEANT-ViT-Large	0.7477	0.7844	0.7639
MEANT-no-lag	0.5942	0.5145	0.5523

Table 6: TempStock MEANT-variant Results, using Precision (P), Recall (R), and F-1 scores.

Examining 6, we see that MEANT-XL exhibited the best performance in F1, precision, and recall. What is perhaps more interesting about these results is examining the performance of MEANT-Tweet-price vs MEANT. The performance drop-off from MEANT-large to MEANT-Tweet-price-large is only about 0.046 in F1 score. Yet MEANT-vision-price-large exhibits a performance drop off of 0.31 from MEANT-large. These results indicate that the Twitter inputs contain features which are more indicative of momentum changes in the MACD indicator than the long-range graph inputs. There are likely many reasons for this phenomena, the primary of which being that stock prices seem to flucuate on short time periods (Zhang et al., 2022) (Xu and Cohen, 2018). As such, the long range information encoded in our graphs likely just introduces noise which degrades model performance.

We find that the five day lag period seems to be ideal for price prediction problems. Testing on a lag period of 10, MEANT performance drops considerably. Tweet information in particular is known to be short range, users tending to contribute information predicated upon immediate trends (Araci, 2019)(Nguyen et al., 2020)(Xu and

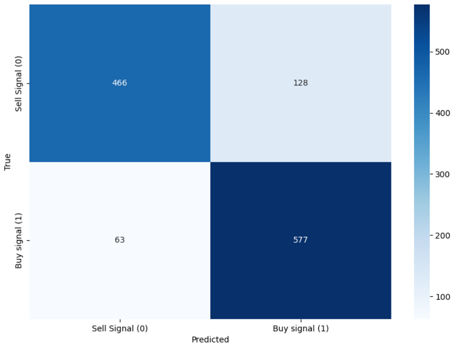

Figure 3: Confusion matrix for MEANT-XL on TempStock

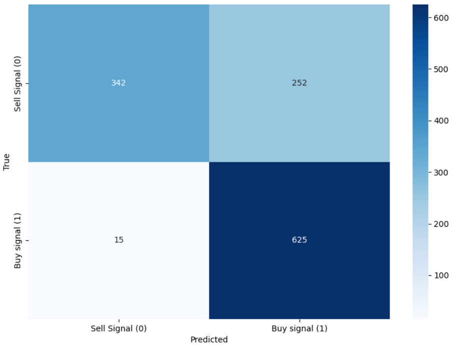

Figure 4: Confusion matrix for TEANet on TempStock

Cohen, 2018)(Zhang et al., 2022). As such, introducing information over a longer time period only serves to weaken the relevant signals our model is looking for. We also tested MEANT without a lag period. Single-day data proves insufficient. As such, 5 appears to be in our Goldilocks zone.

The price modality clearly important to the performance of the MEANT model. MEANT-price, which only takes in M as an input, performs admirably, vastly outperforming MEANT-Tweet-large and MEANT-vision-large, which take in X and G as inputs respectively.

Removing the price modality from MEANT-Tweet model reduces performance by 0.40 in F1 for the large models. For the vision models, the reduction in performance for the large models is 0.36. In fact, performance seems to collapse completely, with the base, large, and XL models of MEANT-vision achieving the same abysmal performance. The price information is what determines to the labels, so it makes sense that our model performance would be negatively affected by the removal of the M inputs.

We did try pretraining, using the TempStock raw Tweets and graphs in masked-language-modeling and masked-image-modling regimes. We found that the pretrained models performed no better on our task. The settings used for our pretraining scheme can be seen in A.2.

In a previous iteration of the model, we used ViTs as the image backbone, and actually fed our concatenated image-tweet encoder outputs into the same temporal attention mechanism. We found that performance with this architecture was worse, likely due to the confusion in the temporal attention pass introduced by the early fusion strategy.

A.1.1 Sequence Projection Vs. Mean Pooling

Using sequence projection vs mean pooling in our temporal attention mechanism had an affect on our model performance across both of our tasks.

Looking at 7, sequence projection outperformed mean pooling for our language encoder outputs on the TempStock task by a reasonable margin, the disparity especially noticeable between MEANT-Large-MP and MEANT-Large-SP.

TempStock is built upon the MACD indicator, which relies on information over a longer time period than simple price prediction (Joshi, 2022), with the MACD calculation involving price averages over 12 and 26 days. Much of that information is not captured in our semantic inputs (Joshi, 2022) which tend to correlate to short term trends of a few days or so (see 6). Furthermore, Tweets tend to vary widely in terms of quality (Araci, 2019)(Xu and Cohen, 2018). What semantic information is pertinent to our final output must be captured with some degree of delicacy, similar to how Xu and Cohen (2018) discerns what Tweets to throw away. A lot of the semantic input is likely just noise which confuses our model, and the parameterized extraction of important Tweets for each lag day alleviates this problem to some extent.

TempStock Seq. proj results	**F1**	**P**	**R**
MEANT-Large-MP	0.6143	0.6241	0.6173
MEANT-XL-MP	0.7983	0.8265	0.8058
MEANT-Large-SP	0.8351	0.8399	0.8343
MEANT-XL-SP	**0.8440**	**0.8497**	**0.8430**

Table 7: TempStock Seq proj results, using Precision (P), Recall (R), and F-1 scores.

Interestingly, mean pooling actually performs better than sequence projection on the StockNet task (see 8). The disparity in this case is glaring. With sequence projection, MEANT performs abysmally, essentially making random guesses with each input. There are likely a few reasons for this.

For one, the StockNet task is a binary price prediction problem, which exists on a far smaller timescale than TempStock it terms of its information. Thus, the semantic Tweet inputs are likely to contain far more robust correlations to the labels then in the TempStock problem. In other words, the Tweets have a far larger sway over StockNet performance then in TempStock (which is a phenomena observed in previous work that measures on the StockNet dataset (Xu and Cohen, 2018) (Zhang et al., 2022)).

Mean pooling manages to preserve spatial information, summarizing local neighborhoods (in this case, Tweets that have been encoded into different part of each sequence in X). A projection, on the other hand, can destroy spatial correlations in the new basis (Tao et al., 2022). What seems to be happening here is our learned projection is throwing away crucial Tweet information, in a problem where the Tweets have a larger importance. While the parameterization serves to intelligently extract the 'relevant' information, in the case where there is little noise in our semantic information, this parameterized projection only serves to damage performance.

Stocknet Seq. proj results	F1	P	R
M-Tweet-large-MP	0.8134	0.8135	0.8133
M-Tweet-XL-MP	**0.8212**	**0.8225**	**0.8211**
MEANT-Large-SP	0.4401	0.5704	0.5259
MEANT-XL-SP	0.4520	0.5725	0.5303

Table 8: Stocknet Seq proj results, using Precision (P), Recall (R), and F-1 scores.

A.2 Pretraining

For experimental purposes, we tried pretraining the MEANT language encoders on the TempStock Tweets.

We follow typical pretraining methods. For our language encoder, we used masked language modeling on our raw TempStock data. We trained our MEANT-small and MEANT-large language encoders on 4 NVIDIA p100 GPUs for 3 and 10 hours respectively. For MEANT-XL, we trained on an A100 GPU for 10 hours. A training batch size of 32 was used.

For the TimeSFormer backbone, we used masked image modeling with block and channel masking. The image encoders were trained on 4 NVIDIA p100 GPUs as well, for 20 hours. We used graphs G from the raw MACD data in TempStock. For these encoders, we also used a training batch size of 32.

A.3 Training Details

All training was done with an AdamW optimizer (Loshchilov and Hutter, 2017) using betas of 0.9 and 0.999, a cosine annealing learning rate scheduler with warm restarts with 7 iterations for the first restart (Loshchilov and Hutter, 2016), and an initial learning rate of $5e^{-5}$. The experiments were all run on a single NVIDIA A100 GPU. More specific settings can be seen in 9.

A.3.1 TempStock Experiment Setup

TEANet makes use of a BERT-style encoder for the Tweet inputs, but uses an LSTM on the concatenated price-Tweet data rather then relying on a pure self-attention based mechanism. Furthermore, TEANet's temporal attention is a softmax-based mechanism which uses some simple concatenation to draw relationships between the last input day and the auxiliary days. TEANet can process lag periods, but cannot process the image inputs and is thus only fed the tweet and price information X and M.

We also fine-tuned VL-BERT (Su et al., 2019) and ViLT (Kim et al., 2021) on TempStock. VL-BERT is an early-fusion multimodal model, that uses a Faster RCNN (Girshick, 2015) to extract the image features, which are concatenated to the textual features before being fed to a BERT-style encoder. VL-BERT cannot process the price data, or data over the lag period, so we fed the model the graphs and Tweets from the final auxiliary day, those being G_{t-1} and X_{t-1} respectively.

ViLT is a single stream encoder that uses a ViT style patch embedding on the images, concatenating these to the text embeddings before feeding the concatenated input to a BERT-style encoder (Kim et al., 2021). ViLT, similarly to VL-BERT, cannot process price data, or data over a lag period. So we fed the model the same inputs as VL-BERT.

We recognized that the lack of price data could give tremendous advantages to TEANet and MEANT over ViLT and VL-BERT, as the labels of TempStock are determined directly from the price component. Thus, we added some extra functionality to our own variants of ViLT and VL-BERT models, called ViLT-price and VL-BERT-price respectively, to handle prices for better comparison of their multimodal strategies. We simply concatenated the price to our encodings of the images and Tweets before feeding the vectors into the attention mechanism. These models recieved the price, graphs, and text data from the last auxiliary day, M_{t-1}, G_{t-1}, and X_{t-1} respectively.

FinBERT and BERT were simply given the Tweets X_{t-1} from the final auxiliary day. For parameter comparisons, see 3

For the TempStock experiment, we used 15 epochs for all MEANT models and a train batch size of 16. We decided to run TimeSFormer on the dataset as well, giving it the images over the lag period as a vision-only baseline. For more simple baselines, we ran a simple MLP on TempStock without a lag functionality, only taking in the prices M_{t-1} from the day before the target period. We also ran an LSTM (Sun et al., 2017), but with a different input of the MACD values m_{t-i} for $i = 1, ...5$, to see if the recurrent properties could extract a pattern.

We used the lag periods from 4/10/2022-12/10/2023 for our training set, the periods from 11/10/2023-2/25/2023 as our validation set, and the periods from 2/25/2023-4/10/2023 as our test set.

A.3.2 StockNet Experiment Setup

The StockNet model was the predecessor to TEANet. StockNet took advantage of a similar Temporal attention mechanism, but used gated recurrent units rather then a BERT-style encoder to process the Tweets, and employed a the use of a latent representation with a variational lower bound for optimization (Xu and Cohen, 2018).

We ran BERTweet on the StockNet-dataset for comparison (Nguyen et al., 2020). For the inputs in this experiment, BERTweet can only process the immediate Tweets before the target day, X_{t-1}. The StockNet model can process the textual information and price information over the lag periods, those being X and M. TEANet X and M in their entirety as well, putting TEANet, StockNet, and MEANT-Tweet on relatively equal footing in terms of their processing capabilities. Experimental settings for each model can be seen in 9.

StockNet is a binary classification problem, like TempStock. StockNet is built upon price movement. Built over a five day lag period, the classification of labels focused on the price change between the adjusted closing price of the last auxiliary day $d-1$ and the target day d, denoted p_d^c and p_{d-1}^c respectively in the original paper (Xu and Cohen, 2018). The labels are determined as follows:

$$y = \mathbb{1}\left(p_d^c > p_{d-1}^c\right) \tag{13}$$

Lag periods that had a movement ratio r where $-0.5\% < r \leq 0.55\%$ were thrown out. The movement ratio is calculated as follows:

$$r = (p_d^c - p_{d-1}^c)/p_{d-1}^c \tag{14}$$

Model	Task	epochs	Batch	Patience
MEANT-B	TempStock	15	16	3
M-Tweet-P-base	StockNet	10	32	3
M-Large	TempStock	15	16	3
	MOSI	15	16	3
M-Tweet-P-Large	StockNet	10	32	3
M-XL	TempStock	15	16	3
M-Tweet-P-XL	StockNet	10	32	3
FinBERT	TempStock	11	16	3
	StockNet	7	32	3
BERT	TempStock	15	16	3
	StockNet	10	32	3
BERTweet	StockNet	15	16	3
VL-BERT	TempStock	15	16	3
ViLT	TempStock	15	16	3
TimeSFormer	TempStock	15	16	3
MLP	TempStock	15	16	3
	StockNet	7	32	3
LSTM	TempStock	15	16	3
	StockNet	4	32	3
TEANet	TempStock	15	32	5
	StockNet	10	16	3

Table 9: Training Settings. M refers to MEANT, and P to Price.

A.4 Model Implementation Details

All models were implemented in Pytorch (Paszke et al., 2019). MEANT was implemented using a typical transformer formula, employing the use of RMSNorm (Zhang and Sennrich, 2019), Flash-attention (Dao et al., 2022), and GELU activation units (Hendrycks and Gimpel, 2016). For our TimeSFormer implementation, we decided to use Phil Wangs (Wang, 2021), for its simplicity, readability, and its use of the Einops library (Rogozhnikov, 2022), which we used in our native MEANT implementations.

There is no public implementation available for TEANet (Zhang et al., 2022), so we implemented the model from the details given in the paper. We used the built in torch LSTM implementation, and the FinBERT embedding layers (Araci, 2019) in order to balance against our implementation of MEANT, and to take advantage of the FinBERT tokenizer.

For all of our BERT-based encoder models, we used the implementations from the transformer models (Wolf et al., 2019).

Model	Parameter Count
MLP	3,400,642
LSTM	16,400,642
VL-BERT	111,450,624
ViLT	111,595,008
BERT	134,899,968
MEANT-base	48,304,272
MEANT-large	152,367,264
MEANT-XL	265,890,528

Table 10: Parameter Counts

A.5 CMU-MOSI

We also decided to test our model on the CMU Multimodal Opinion-level Sentiment Intensity (MOSI) dataset (Zadeh et al., 2016).

This dataset includes audio, text, and video modalities compiled in 299 annotated video segments collected from YouTube monologue movie reviews. The data forms a binary sentiment analysis classification task.

For our purposes, we focus on the text and video modalities. We run MEANT on these inputs.

CMU-MOSI is of interest because it examines videos with aligned text over time. Our vision backbone, the TimeSFormer model, is built for video inputs (Bertasius et al., 2021). We measured MEANT against previous SOTA baselines. TEASEL is a multimodal model that uses a pre-trained RoBerta as a backbone (Arjmand et al., 2021), using a CNN to break down the audio signals before coupling those with the text. UniMSE is an encoder-decoder model which breaks down the audio, visual, and textual modalities in fusion layers (Hu et al., 2022). UniMSE also uses a CNN to process the visual features. MMML is the current SOTA for the CMU-MOSI benchmark. MMML uses cross-modal attention, which is integrated into a fusion network (Wu et al., 2024). Interestingly, MMML does not take in visual inputs. The MEANT-large runs below were collected after 15 epochs of training, using the same optimizer and lr scheduler settings listed above. The other results were taken from previous work (Wu et al., 2024).

CMU-MOSI Results	$F1_{non0}$	$F1_{has0}$	ACC_{2has0}
TEASEL	85	84.72	84.79
UniMSE	86.42	85.83	85.85
MMML	89.67	87.45	87.51
MEANT-large	71.43	70.30	70.32

Table 11: Mosi-dataset experiment results using Precision (P), Recall (R), F-1 scores and testing accuracy (Acc).

Looking at the results above, MEANT-large performs considerably worse then previous SOTA benchmarks on the MOSI task. The disparity is expected. *Query-Targeting* in MEANT is designed to put great emphasis on the final component in the information period. In the CMU-MOSI task, this refers to the final frame in the video clip, along with the final text token, which have been aligned. The clips in the dataset are short movie reviews. The final frame in these clips does not contain significant information as to the entire clip (Zadeh et al., 2016), in the manner that the final price day in a lag period does to a stock price (Zhang et al., 2022) (Xu and Cohen, 2018).

Furthermore, the previous state of the art benchmarks are designed to handle the audio component, which is better aligned to the textual inputs then the video embeddings (Zadeh et al., 2016). MEANT was working off of the visual and textual inputs alone. Thus, the performance we do achieve speaks to the soundness of our current architecture.

We did run TEASEL and UniMSE on TempStock, replacing the audio inputs to their CNNs with our graphical data G. We changed the models to support our price data M. They were trained over 15 epochs, and train batch size of 16, and all other experimental settings identical to those used in the original TempStock experiments.

MOSI models on TempStock	$F1$	P	R
TEASEL	0.6228	0.6148	0.5745
UniMSE	0.7343	0.7238	0.7315
MEANT-large	0.8351	0.8399	0.8343

Table 12: Models which performed well on MOSI, ran on TempStock. Results use Precision (P), Recall (R), F-1 scores and testing accuracy (Acc).

The models which perform at such a high level on MOSI fail to perform as well on the TempStock task, as seen in 12. Ideally, one architecture could tackle both of these sorts of problems. In future work, we would like to make our temporal mechanism more robust to dependencies across the time dimension of the entire input. One method would be to extend our *Query-Targeting* mechanism to learn a parameterized selection of the best target components, or to learn which parts of the input the other auxiliary dependencies need to be collected in relation to. This could involve a separate temporal matrix, as in Rosin and Radinsky (2022), or some sort of softmax query weighting prior to the attention computation. Creating a mechanism which can perform at the highest level on any temporally dependent benchmark remains an open problem.

A.6 TempStock Dataset Details

The tables below show the number of lag periods used in TempStock for each ticker.

Ticker	Count
DHI	32
HWM	31
PCG	30
LEN	28
DG	28
IR	27
EL	27
AVGO	26
CTRA	26
IEX	26
XRAY	26
TER	26
KR	26
UPS	25
PAYC	25
META	25
L	25
PGR	25
FITB	25
BKR	25
LYV	25
DRI	25
MET	25
WYNN	25
SHW	25
APTV	25
SEE	25
AMCR	24
ADI	24
ANSS	24
HUM	24
DXC	24
CRM	24
SBNY	24
STLD	24
CMI	24
PWR	24
MKTX	24
LUV	24
REGN	24
RTX	24
MNST	24
CDW	24
MHK	24
VRTX	24
TMUS	23
TRGP	23
WAB	23
APH	23
FTNT	23
GRMN	23
FDX	23

Table 13: TempStock Companies Chunk 1

Ticker	Count
FE	23
JNPR	23
INTU	23
HBAN	23
NOC	23
CLX	23
LVS	23
SBUX	23
JPM	23
NOW	23
DGX	23
LOW	23
PNC	23
PPG	23
ECL	23
ZTS	23
TMO	23
XYL	23
EPAM	22
DAL	22
LUMN	22
MRO	22
MGM	22
MTCH	22
ENPH	22
HSY	22
GIS	22
OTIS	22
NRG	22
WRB	22
EVRG	22
NDSN	22
NVR	22
CHD	22
CBOE	22
HCA	22
CDNS	22
SWKS	22
PEP	22
LW	22
TYL	21
RL	21
SWK	21
FANG	21
PTC	21
QCOM	21
DUK	21
MTD	21
AEP	21
LLY	21
MMM	21
ABT	21

Table 14: TempStock Companies Chunk 2

Ticker	Count
ZBH	21
UNP	21
TSCO	21
TFC	21
LHX	21
HIG	21
HON	21
KEYS	21
KDP	21
CBRE	21
CMS	21
MSFT	21
NSC	21
VMC	21
AIG	21
GM	21
FOX	21
BAC	21
TTWO	21
BIO	21
ETSY	21
ZION	20
MCK	20
NVDA	20
CHRW	20
CAG	20
LKQ	20
BBY	20
BIIB	20
HLT	20
NEM	20
CCI	20
FTV	20
CARR	20
ODFL	20
PCAR	20
WBA	20
PEG	20
PSX	20
HII	20
GL	20
SJM	20
CI	20
FSLR	20
TJX	20
MAR	20
CSGP	20
UAL	20
T	20
SNPS	20
AEE	20
DTE	20

Table 15: TempStock Companies Chunk 3

Ticker	Count
ETN	20
WHR	20
GOOGL	19
GOOG	19
SYK	19
DLR	19
AES	19
ADP	19
AIZ	19
ADSK	19
AKAM	19
KEY	19
TRMB	19
UDR	19
JNJ	19
IBM	19
ILMN	19
CF	19
SCHW	19
CB	19
CINF	19
PAYX	19
PYPL	19
IVZ	19
FOXA	19
EFX	19
OXY	19
TECH	19
VRSK	19
HPE	19
NDAQ	19
NTRS	19
CNC	19
CMA	19
CSCO	19
ALL	19
ABBV	19
LNT	19
VFC	18
VTRS	18
AAL	18
AMGN	18
YUM	18
CEG	18
C	18
ON	18
NKE	18
NXPI	18
AAP	18
EXR	18
EQT	18
CE	18

Table 16: TempStock Companies Chunk 4

Ticker	Count
ORLY	18
JCI	18
MPC	18
CVS	18
GE	18
K	18
TXN	18
HD	18
MOS	18
CVX	18
CL	18
HPQ	18
ITW	18
WMT	18
PM	18
MU	18
MPWR	18
MSCI	18
MAS	18
TEL	18
BAX	18
VZ	18
WMB	18
SLB	18
DFS	18
WST	18
MCD	18
MRK	18
DXCM	18
SYY	18
AMAT	18
AFL	17
A	17
MRNA	17
NTAP	17
NWSA	17
NEE	17
MAA	17
CSX	17
DHR	17
IRM	17
J	17
DE	17
CPT	17
OGN	17
ED	17
LIN	17
CAT	17
BSX	17
F	17
BEN	17
EXPD	17

Table 17: TempStock Companies Chunk 5

Ticker	Count
EA	17
EOG	17
CTSH	17
KLAC	17
CMG	17
FCX	17
FMC	17
IPG	17
BK	17
BKNG	17
TROW	17
PNR	17
CRL	17
WAT	17
WFC	17
RMD	17
BLK	17
EIX	17
EW	17
D	17
WDC	17
STX	17
SNA	17
RHI	17
SBAC	17
V	17
AXP	17
AMT	17
VLO	16
PSA	16
BBWI	16
BDX	16
TGT	16
TDY	16
WRK	16
WY	16
WTW	16
XEL	16
WBD	16
TSN	16
LRCX	16
LMT	16
BMY	16
GPN	16
GS	16
HSIC	16
CTVA	16
LYB	16
MA	16
GPC	16
GILD	16
CTLT	16

Table 18: TempStock Companies Chunk 6

Ticker	Count
ROK	16
MKC	16
ADM	16
ACGL	16
ANET	16
AZO	16
ALLE	16
ELV	16
ETR	16
EXC	16
XOM	16
EMR	15
EQR	15
ESS	15
ZBRA	15
ACN	15
ATO	15
AMP	15
CTAS	15
PARA	15
ROP	15
GLW	15
MTB	15
MLM	15
DPZ	15
DD	15
NFLX	15
BRO	15
KMX	15
GD	15
USB	15
SRE	15
STT	15
CME	15
CMCSA	15
INCY	15
IFF	15
RSG	15
FDS	15
BWA	15
BXP	15
TFX	15
NI	15
NUE	15
ORCL	15
PNW	15
PLD	15
IT	15
AVB	15
AWK	15
AJG	14

Table 19: TempStock Companies Chunk 7

Ticker	Count
UHS	14
VICI	14
FIS	14
GEN	14
NCLH	14
DLTR	14
IP	14
INVH	14
RJF	14
NWL	14
HST	14
PKG	14
CPB	14
COF	14
GNRC	14
EMN	14
MCHP	14
MDLZ	14
MS	14
CNP	14
PH	14
CCL	14
DVA	14
DVN	14
SPG	14
TSLA	14
ULTA	14
KMB	14
KHC	14
RCL	14
BALL	14
SYF	14
APD	13
MO	13
AMZN	13
AVY	13
EQIX	13
CZR	13
JBHT	13
PXD	13
VNO	13
RF	13
PFE	13
ISRG	13
ICE	13
INTC	13
LDOS	13
COST	13
MDT	13
COP	13
AON	13
AOS	13

Table 20: TempStock Companies Chunk 8

Ticker	Count
STE	13
VTR	13
WM	13
DIS	13
JKHY	12
PG	12
IQV	12
AMD	12
FRT	12
ALB	12
KIM	12
MSI	12
ROST	12
URI	12
ES	12
HRL	12
O	12
GWW	12
PFG	12
PPL	12
ADBE	12
HAL	12
DOW	12
ARE	11
BR	11
IDXX	11
AAPL	11
BA	11
OKE	11
VRSN	11
WELL	11
TDG	11
SPGI	11
EBAY	11
CPRT	11
UNH	10
SO	10
NWS	10
KMI	10
REG	9
DOV	9
HES	1

Table 21: TempStock Companies Chunk 9