Nparam Bull V1 Large Stock Model

John Lins — john@johnlins.com

Abstract — Numerical data is the foundation of quantitative trading. However, qualitative textual data often contains highly impactful, nuanced signals that are not yet priced into the market. Nonlinear dynamics embedded in qualitative textual sources such as interviews, hearings, news announcements, and social media posts often take humans significant time to digest. By the time a human trader finds a correlation, it may already be reflected in the price. While large language models (LLMs) might intuitively be applied to sentiment prediction, they are notoriously poor at numerical forecasting and too slow for real-time inference. To overcome these limitations, we introduce Large Stock Models (LSMs), a novel paradigm tangentially akin to transformer architectures in LLMs. LSMs represent stocks with high-dimensional embeddings, learned from decades of historical press releases paired with corresponding daily stock price percentage changes. We present Nparam Bull V1, a 360M+ parameter LSM designed for fast inference, which predicts short-horizon stock price fluctuations of many companies in parallel from raw textual market data. Nparam Bull V1 surpasses equal-weighting strategies, marking a breakthrough in short-horizon quantitative trading.

1 NDA Notice

Developing a stable model was a lengthy, year-long endeavor requiring extensive trial and error. Certain stability-related insights and methodologies will be publicly shared in this technical report; however, the majority of substantive procedures will remain redacted. For access to additional specifics, please contact the author.

2 Introduction

We introduce Nparam Bull V1, a pioneering Large Stock Model (LSM) with over 360 million parameters, designed specifically for predicting stock price fluctuations from raw textual market data [5, 1]. Each prediction is generated by processing textual input and outputting an expected percentage shift in stock prices, enabling rapid inference suitable for short-horizon trading environments. For this initial version, Nparam Bull V1 was trained to predict price fluctuations for 90 top stocks (primarily chosen from the NASDAQ 100) and 10 major ETFs. This selected universe may have survivorship bias; however, when performing the backtest, both weight schemes use the identical universe, so survivorship bias partially cancels in the model-vs-equal-weighted comparison. A key consideration in deploying Nparam Bull V1 is the efficient market hypothesis, which posits that all publicly available information is already reflected in current prices. Consequently, one cannot generate profitable trades with the model by feeding it data that has already been priced into the market, such as historical or widely disseminated news during active trading hours. Instead, profitability hinges on inputting data that is not priced in yet [3]. Nparam Bull V1 was trained exclusively on data from periods prior to 2023, encompassing decades worth of paired text-price examples from earlier years. Validation was then performed only on 2023 and 2024, years entirely unseen during training. This backtest demonstrated the model's ability to generalize, achieving higher returns when using model-weighting compared to the equal-weighted baseline of the same stocks. Beyond its core predictive capabilities, Nparam Bull V1 opens avenues for advanced applications in quantitative finance. For instance, a quant firm or hedge fund could integrate the model into a pipeline by feeding multiple textual inputs—such as a batch of concurrent news articles, regulatory filings, or social media sentiment aggregates—into the system simultaneously. By averaging the resulting predictions across these inputs, users can derive a composite signal that mitigates noise from individual sources. This averaged output could then inform daily portfolio weightings, where each stock or ETF's allocation is adjusted based on the predicted price fluctuation magnitude and direction. For example, assets with positive predicted shifts might receive overweight positions, while those with negative forecasts are underweighted or shorted. Such strategies could extend to ensemble methods, combining Nparam Bull V1 with other models for hybrid forecasting. Future iterations will scale coverage to thousands of assets, further enhancing its utility for institutional trading.

3 Methods

3.1 Model Architecture

3.2 Dataset

NDA Protected

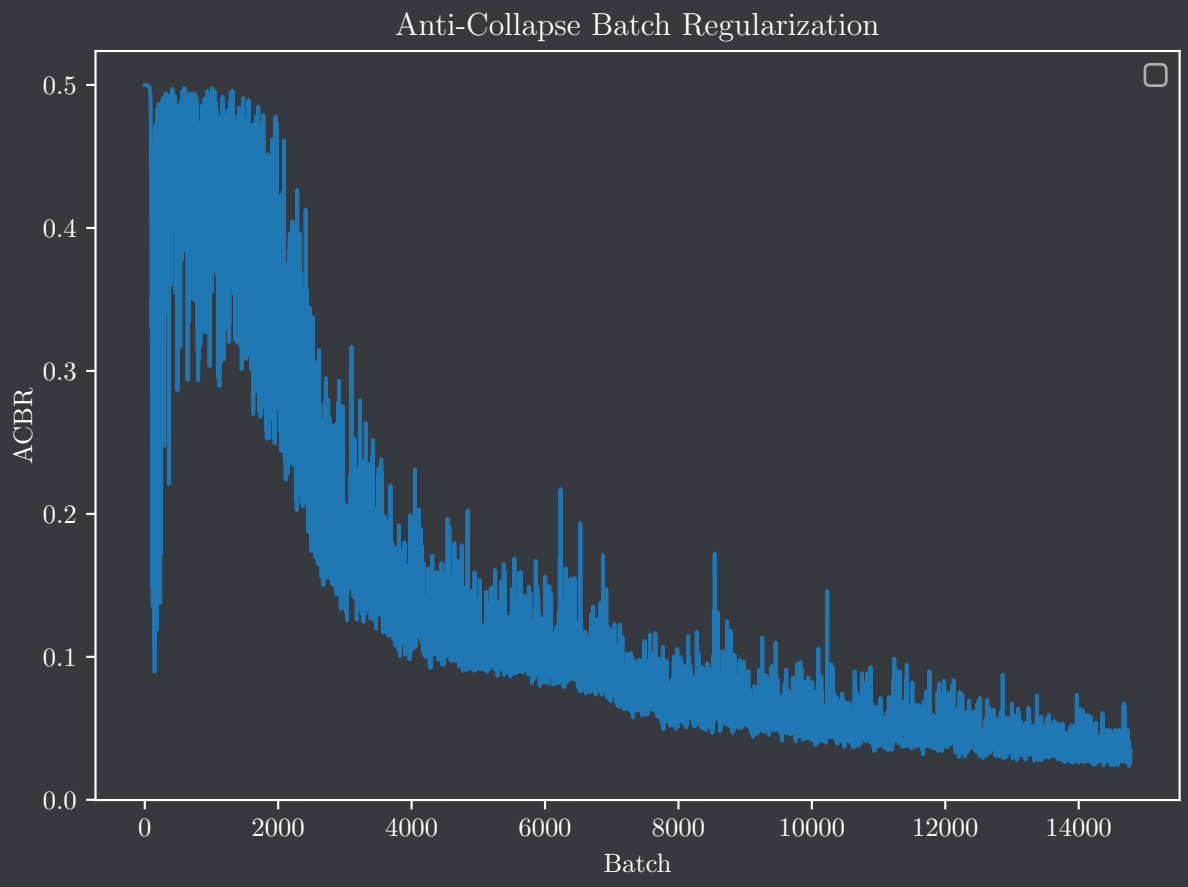


Figure 1: The ACBR term initially increased as the gradient balanced a dual-objective optimization, but eventually decreased once the other objective (MSE) stabilized.

4 Training

4.1 Anti-Collapse Batch Regularization (ACBR)

Due to the high variation in stock data, the model may lose motivation to learn the correlations when it is easier to simply ignore the input signal and converge to an average—an effective way to exploit the loss objective. To address this issue, we promote diversity among the output vectors within each training batch by incentivizing distance between all pairs in the batch. Initially, we experimented with negative Euclidean distance, but the model exploited this by shrinking output magnitudes. We then settled on cosine similarity, which proved highly effective and more robust.

4.1.1 ACBR Term

$$\text{ACBR}(\mathbf{P}) = \frac{1}{\binom{b}{2}} \sum_{1 \le i < j \le b} \text{ReLU}\left(\frac{\mathbf{p}_i \cdot \mathbf{p}_j}{\|\mathbf{p}_i\|_2 \|\mathbf{p}_j\|_2} \right)$$

where $\mathbf{P} \in \mathbb{R}^{b \times s}$ is the output matrix with b rows $\mathbf{p}_1, \ldots, \mathbf{p}_b$, one for each vector in the batch ($b = 32$) and s columns, one for each ticker ($s = 100$). In essence, this formula is *the average of all positive cosine similarity between every pair of vectors in the batch.*

4.1.2 ACBR Optimization

$$\mathcal{L} = \mathcal{L}_{MSE} + \gamma \text{ACBR}(\mathbf{P}) + \lambda \|\mathbf{w}\|_2^2, \; \gamma = 0.5, \; \lambda = 10^{-3}$$

4.2 Loss

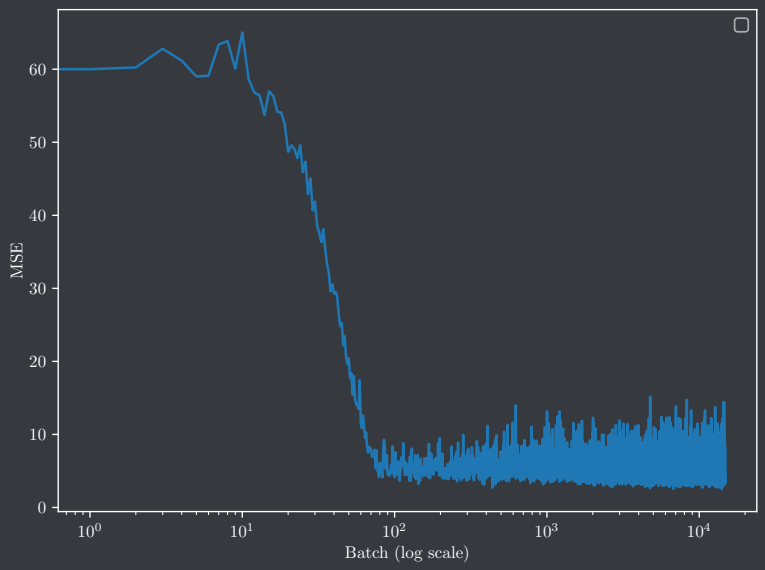


Figure 2: The training loss dropped sharply early in the first epoch and later oscillated slightly.

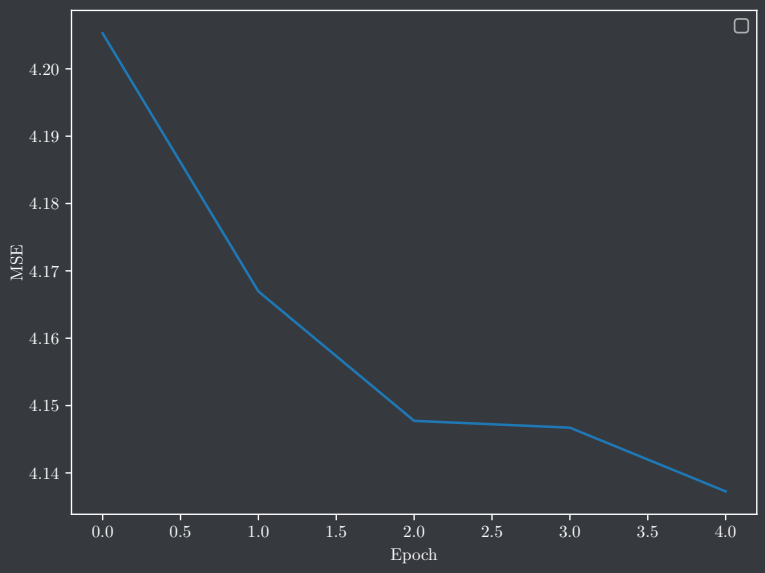
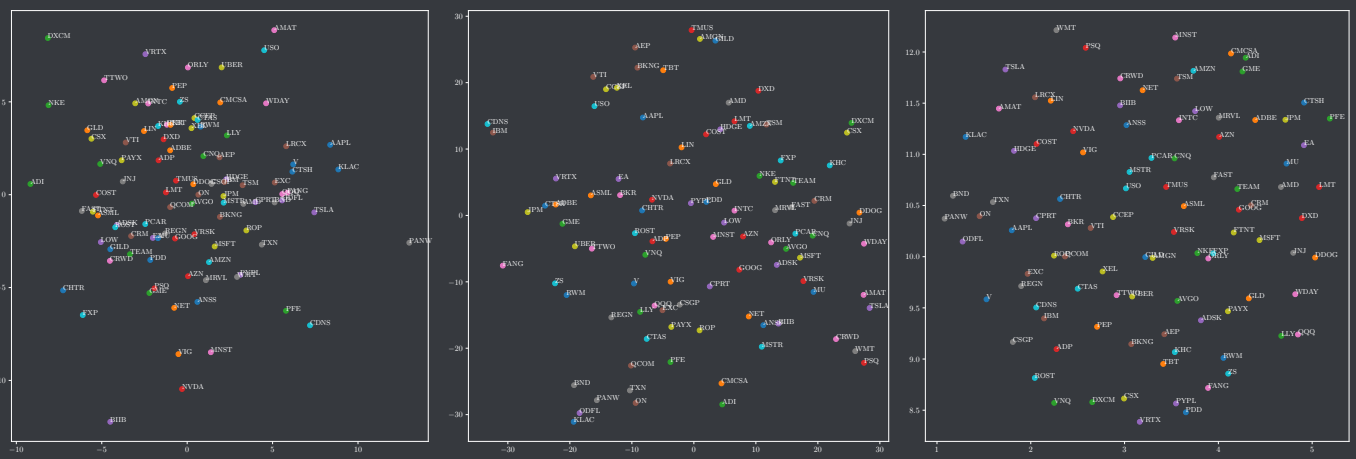


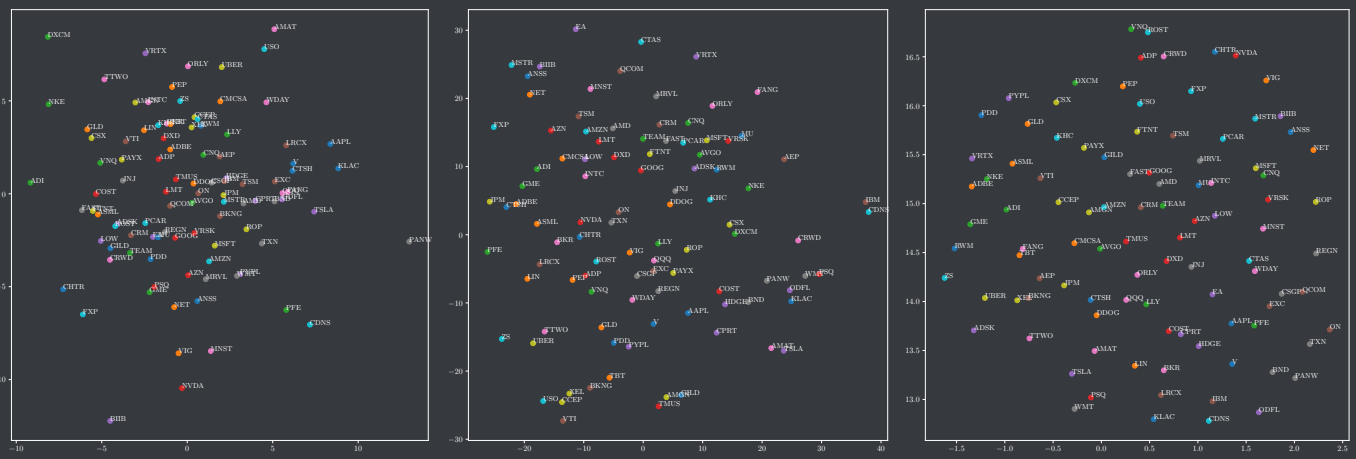
Figure 3: The test loss decreased steadily for five epochs.

4.3 Stock Embeddings



PCA t-SNE UMAP

Figure 4: Visualization of stock embeddings before training.



PCA t-SNE UMAP

Figure 5: Visualization of stock embeddings after training.

The initial embeddings were sufficiently linearly separable that their primary role was simply to identify which ticker the model should predict. Although each vector encodes only a small amount of additional information after training, that signal is present—indicating there is potential to further enrich these embeddings to capture significantly more data as the model scales.

5 Training Specifications

NDA Protected

6 Backtests

6.1 Portfolio Weighting

$$w_i = \begin{cases} \frac{\widehat{\%\Delta_i}}{\sum_{j:\widehat{\%\Delta_j}>0} \widehat{\%\Delta_j}} & \text{if } \widehat{\%\Delta_i} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where w_i is the portfolio weight for asset i, and $\widehat{\%\Delta_i}$ is its predicted percent change. These weights are assigned at the start of the trading window when the stocks are bought, and they are then all sold at the end of the trading window for a given day.

6.2 Hindsight Backtest

Year: 2023
Dataset: FNSPID [2]
Bias: Lookahead (model makes trades at market open based on news published throughout that day)
Hold time: Held for 24 hours and reweighted at market open
Average number of input data points per day: 2614

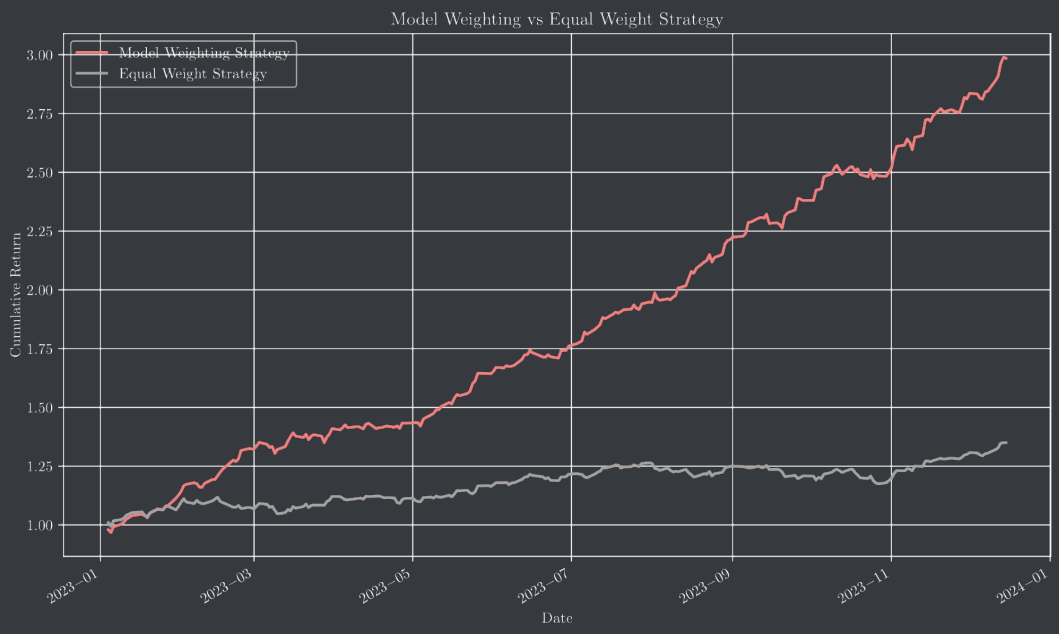


Figure 6: A hindsight backtest illustrates how well a model can perform given all the information, including future information. This hindsight backtest illustrates a +198% increase for the model-weighted portfolio compared to just +35% for the equal-weighted portfolio. Here, the model is given news from throughout the entire day to make trade decisions at market open. While these conditions are impossible in the real world, this backtest serves as an upper bound on achievable performance if all relevant information were available at decision time.

6.3 Feasible Backtest

NDA Protected

6.3.1 Bias

The hindsight backtest intentionally suffers from lookahead bias, which the feasible backtest does not.
One could also argue that both backtests suffer from moderate selection bias, more specifically survivorship bias. This is because the universe of stocks used for training Nparam Bull V1 was not randomly selected; most of the stocks are taken from the current (2025) NASDAQ 100, which may have changed since 2023/2024. Also, some stocks were manually replaced by other companies to inject diverse coverage of different asset types.
The baseline equal-weighted portfolio is constructed using the same universe and therefore shares these sources of bias. As a result, common components of survivorship and selection bias partially cancel in relative comparisons, making the difference between the model portfolio and the equal-weighted portfolio a more reliable measure of performance than absolute returns.
It is also important to acknowledge that these backtests do not account for slippage; slippage was not accounted for here because the 100 assets being traded are highly liquid.
These backtests also do not account for exchange fees; however, for both backtests, re-weighting only occurs once per day.

6.3.2 Mean Shifting

Naturally, the model will learn to overpredict stocks that have historically done well and underpredict stocks that have historically done poorly. We know that past price performance does not reliably predict future price performance. Removing this bias from the prediction is easy: we simply subtract the prior belief for a given stock across all data points. This prior is the average change for each stock in the training dataset.

$$\widehat{\%\Delta_i} \leftarrow \widehat{\%\Delta_i} - \overline{\%\Delta_i} \text{ for each asset } i.$$

This process may lead to more accurate results, but the way these backtests are set up may lead to less diverse portfolios when provided limited data.
Shifting increased the performance of the hindsight backtest from +198% to +218% but decreased the performance of the feasible backtest from +17.94% to +17.14% and increased the max drawdown from -5.29% to -7.55%. This occurred because the feasible backtest had much less data available to it.

7 Conclusion

Nparam Bull V1 represents a pioneering advancement in the application of large-scale models to short-horizon quantitative trading, bridging the gap between qualitative textual data and precise numerical forecasting. Our Large Stock Model (LSM) achieves superior performance in predicting short-horizon price fluctuations, outperforming baselines such as equal-weighted strategies. The introduction of Anti-Collapse Batch Regularization further enhances training stability, enabling the model to capture nuanced market dynamics without succumbing to collapse or overfitting.
It is widely believed that price reactions are not instantaneous, but rather propagate through a series of corrections over time [6]. While most alpha may be short-lived for trivial reactions, there are many less-obvious reactions that come from the nonlinear relationships between textual market data and the market; these take longer to be realized. Nparam Bull V1 may have the ability to find these connections during inference. This work underscores the potential of LSMs to automate and accelerate decision-making in financial markets. The process of scaling this model further is underway, and future versions will learn to establish even more nuanced nonlinear relationships between textual data and daily market returns.

8 Future Work

NDA Protected

References

[1] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

[2] Zihan Dong, Xinyu Fan, and Zhiyuan Peng. Fnspid: A comprehensive financial news dataset in time series. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4918–4927, 2024.

[3] Eugene F. Fama. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417, 1970.

[4] Polygon.io. Polygon.io financial market data, 2024. https://polygon.io.

[5] Paul C. Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168, 2007.

[6] Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The journal of finance*, 63(3):1437–1467, 2008.