# Nparam Bull V1 Large Stock Model

John Lins

**Abstract** — Numerical data is the foundation of quantitative trading. However, qualitative textual data often contain 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 as ultra-high-dimensional embeddings, learned from decades of historical press releases paired with corresponding daily stock price percentage changes. We present Nparam Bull, a 360M+ parameter LSM designed for fast inference, which predicts instantaneous stock price fluctuations of many companies in parallel from raw textual market data. Nparam Bull surpasses both equal-weighting and market-cap-weighting strategies, marking a breakthrough in high-frequency 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. Any further disclosures will occur in person.

## 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 inputs. For this initial version, Nparam Bull V1 was trained to predict price fluctuations for 90 top stocks and 10 major ETFs. Each prediction is generated by processing textual input and outputting an expected percentage shifts in stock prices, enabling rapid inference suitable for high-frequency trading environments. 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. Nparam Bull V1 was trained exclusively on data from periods prior to 2023, encompassing decades worth of paired text-price examples from earlier years. Evaluation was then performed only on data from 2023—a year entirely unseen during training. This backtest demonstrated the model's ability to generalize, achieving a remarkable +318% return when using model-weighted portfolios, far surpassing equal-weighting (+35%) portfolio. 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 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

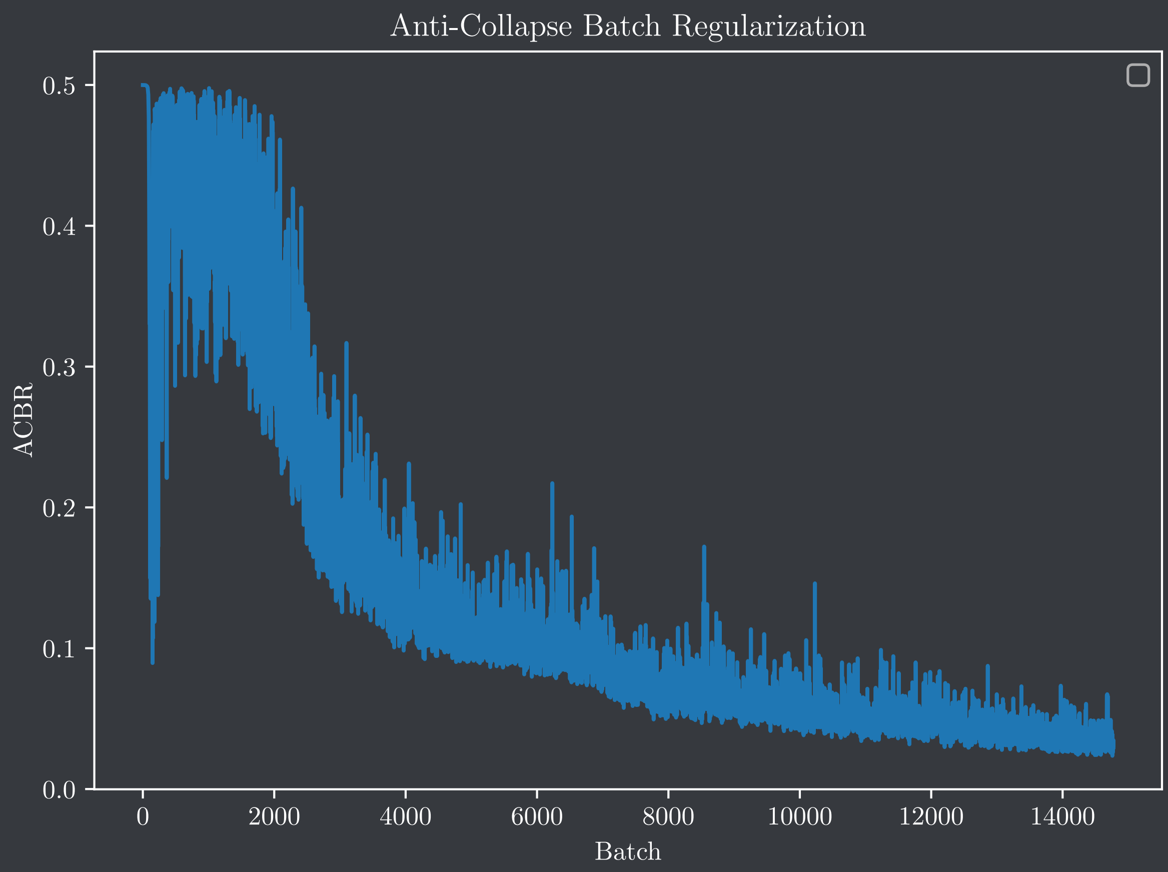



Figure 1: The ACBR term initially increased as the gradient balanced a dual-objective optimization, but eventually decreased once the other objective (MSE) stabilized.

# 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 distances 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

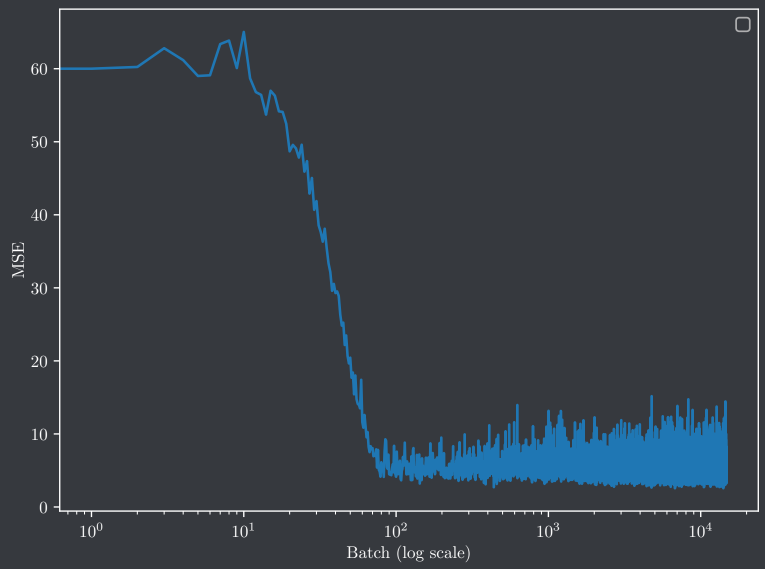



Figure 2: The training loss dropped sharply early in the first epoch and later oscillated slightly.

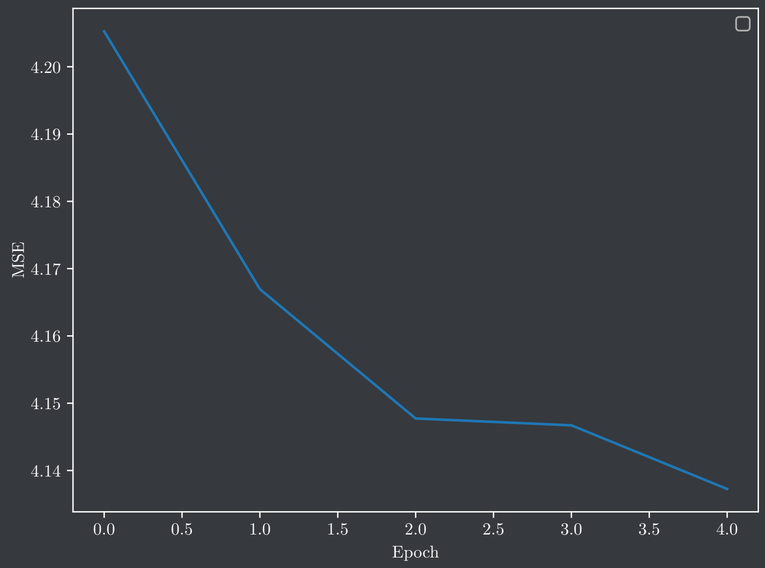


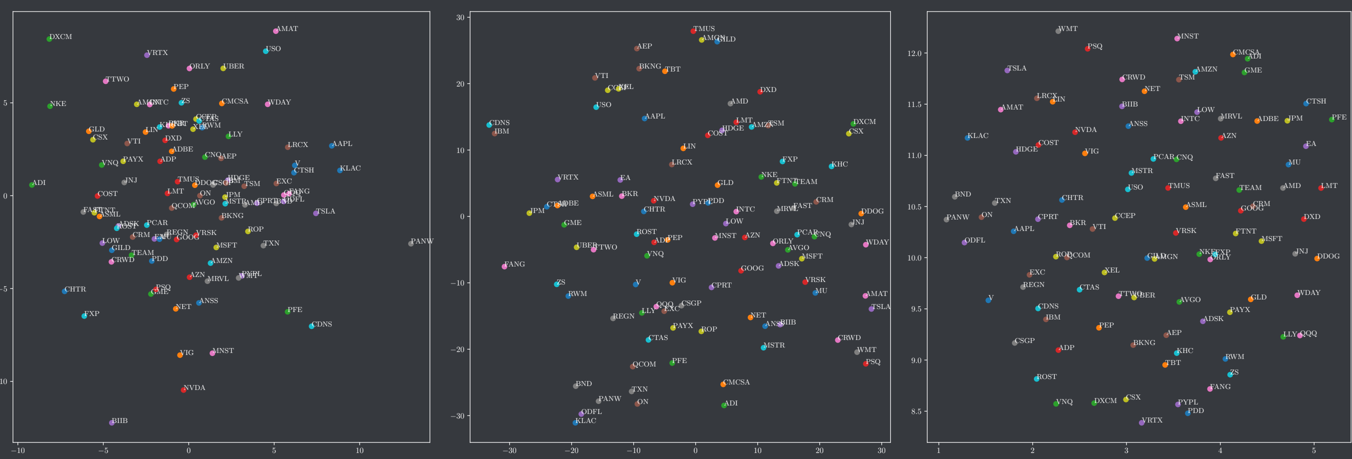

Figure 3: The test loss decreased steadily for five epochs.

### 4.3 Remove Historical Bias Post-training

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

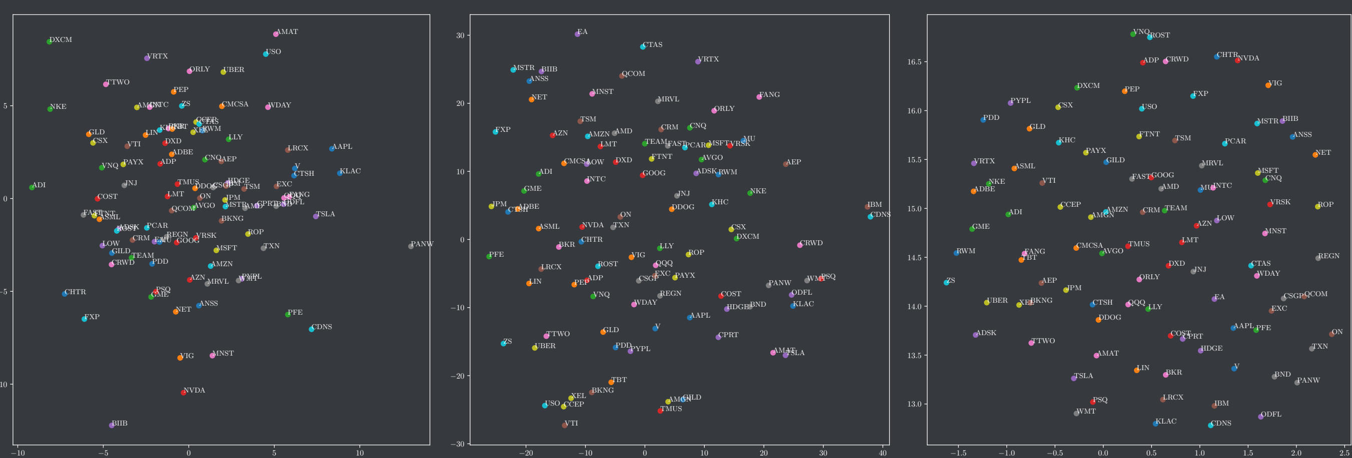
$$\widehat{\%\Delta_{\text{daily}}} \leftarrow \widehat{\%\Delta_{\text{daily}}} - \overline{\%\Delta_{\text{daily}}}$$

### 4.4 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.

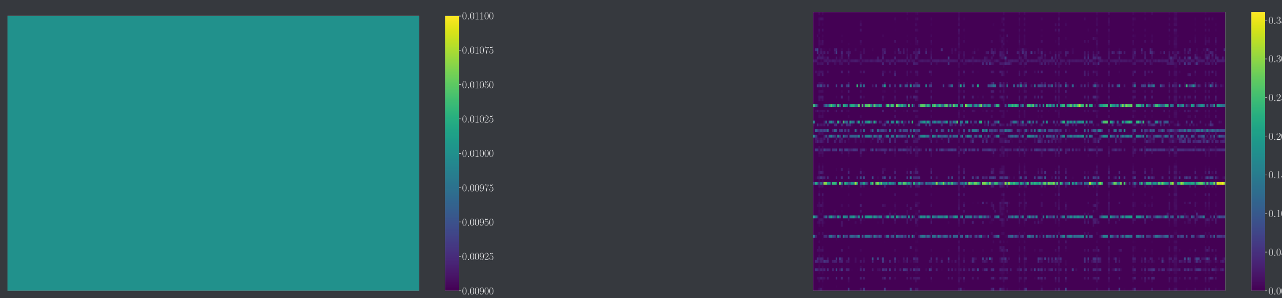
## 5 Training Specifications

NDA Protected

## 6 Results

We performed a backtest on our test dataset of 14K articles. The test dataset includes only news in the year 2023, this model was never training on any data in the year 2023.


This backtest includes both data that is already priced in (pre-market) and data that is not priced in yet (after-market), as well as a grey area, during market hours.




(a) Equal Weights (b) Model Weights

Figure 6: This figure compares trade diversity of equal- versus model-weighted assets. The x-axis shows the date, and the y-axis shows tickers, arranged alphabetically from top to bottom.

$$w_i = \begin{cases} \frac{\Delta p_i}{\sum_{j:\Delta p_j > 0} \Delta p_j} & \text{if } \Delta p_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Where $w_i$ is the portfolio weight for asset $i$, and $\Delta p_i$ is its predicted percent change.

### 6.1 Backtest

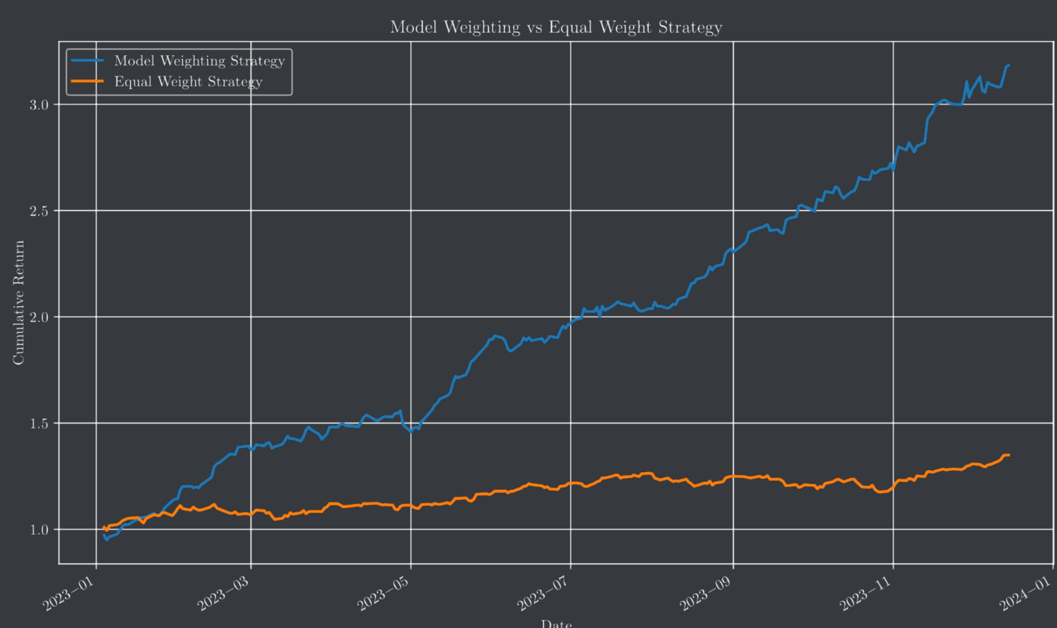



Figure 7: This remarkable backtest illistrates a +318% increase for the model-weighted compaired to just a +35%

## 7 Conclusion

Nparam Bull V1 represents a pioneering advancement in the application of large-scale models to high-frequency quantitative trading, bridging the gap between qualitative textual data and precise numerical forecasting. Our Large Stock Model (LSM) achieves superior performance in predicting instantaneous price fluctuations, outperforming traditional benchmarks such as equal-weighting strategies with a remarkable +318% return in backtesting. 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. 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 market data and daily market returns.

## 8 Future Work

**NDA Protected**