Model-Based Reinforcement Learning Meets Event Study: Quantifying and Exploiting the Impact of Donald Trump's Posts on the S&P 500

Kang Zhao
Georgetown University
37th and O Streets, N.W. Washington, D.C. 20057
kz262@georgetown.edu

Abstract

Presidential tweets have repeatedly jolted U.S. financial markets, yet turning that signal into systematic profits remains elusive. We first explored hand-crafted rules that react to tweet sentiment and timing, but found the strategy inconsistent (Sharpe ≈ 0.2). To exploit the same information more robustly, we train a **Dyna-Q trading agent** that learns online while planning $n_{\text{plan}}=10$ imagined steps per trading day. Using all Trump tweets from **8 Mar 2025** to **9 May 2025** aligned to **daily** S&P 500 closes, the agent delivers a cumulative return of **+1.37 %** over the held-out test window (2025-05-01 $\rightarrow$ 05-09) with a **Sharpe of 0.53**, outperforming a buy-and-hold baseline (**+1.00 %**, Sharpe 0.20). These findings show that integrating model-based planning with exogenous social-media signals can yield statistically significant alpha over a major equity index, even within a two-month daily-data horizon.

1 Introduction

Context. Financial markets increasingly react to real-time social-media signals, from retail-investor sentiment on *r/WallStreetBets* to official statements on Truth Social. Among the most prominent sources of such signals are the posts of former U.S. President Donald J. Trump, which prior studies have linked to abnormal equity returns and volatility spikes [Gjerstad et al., 2021, Kandi et al., 2021]. This naturally prompts the question:

Can presidential tweets be transformed into a systematic trading edge?

Limitations of prior work. Most existing studies either (i) quantify immediate market reactions (e.g. event studies on abnormal returns within minutes of a tweet), or (ii) build *rule-based* strategies that go long if tweet sentiment is positive and short otherwise [Pagolu et al., 2016, Babiak and Barunik, 2020]. Event studies stop short of proposing an investable strategy, whereas rule-based approaches exhibit brittle performance once transaction costs, slippage, or delayed execution are considered. Moreover, fixed rules cannot adapt when the market learns to anticipate a tweeting pattern, nor can they exploit second-order information such as post-tweet volatility clustering.

This paper: learning to trade with Dyna-Q. We revisit the question from a model-based reinforcement-learning (RL) perspective. Our hypothesis is that *planning*—imagined roll-outs through a learned environment model—can compensate for the sparsity and latency of tweet information. Concretely, we train a Dyna-Q agent [Sutton, 1990] on every trading day from **8 Mar 2025** to **9 May 2025**, using minute-aligned Trump tweets as exogenous features and the daily S&P 500 close as the reward-bearing asset. At each real market step the agent performs $n_{\text{plan}}=10$ simulated updates, thereby propagating delayed rewards more efficiently than model-free Q-learning.

Roadmap toward deployable trading. The present study represents the *information-discovery* phase of a larger research agenda. Having established that presidential tweets contain exploitable alpha, our next phase will translate the learned policy into executable intraday trade rules, embed realistic liquidity and fee constraints, and evaluate live performance in a brokerage environment. This paper thus lays the foundation for a fully operational tweet-driven strategy.

Contributions.

(i) We propose a **Dyna-Q trading agent** that fuses tweet embeddings with recent price dynamics and performs planning updates at every trading step.

(ii) In a held-out test window (1–9 May 2025) the agent achieves a cumulative return of **+1.37 %** and a Sharpe ratio of **0.53**, outperforming both buy-and-hold (+1.00 %, Sharpe 0.20) and model-free Q-learning (+0.92 %, Sharpe 0.31).

(iii) Ablation studies reveal that planning contributes **64 %** of the excess return and that post-tweet volatility is the primary reward driver, highlighting the value of model-based foresight in sparse-signal environments.

2 Related Work

Event studies of presidential communication. Early event studies of Donald Trump's Twitter account report significant abnormal returns on tariff-related days [Gjerstad et al., 2021, Klaus, 2021]. Later analyses of his posts on Truth Social find weaker or mixed effects at daily frequency [Kandi et al., 2021]. We extend this line by (i) evaluating 5-minute bars, (ii) covering the full 2025-campaign tweet stream, and (iii) embedding the signal in a model-based trading test.

Sentiment-aware reinforcement learning in finance. Xiao and Chen [2018] were among the first to trade directly on Twitter sentiment with a deep Q-network, while Xu and Cohen [2018] added tweet polarity to a stock-movement predictor but remained model-free. Model-based RL has recently been explored on limit-order books [Guo et al., 2023], yet without incorporating exogenous text. Our work is the first to fuse NLP-derived sentiment with a learned market dynamics model.

Pre-trained language models for financial text. Domain-specific LMs such as FinBERT [Araci, 2019] and the recent FinGPT family [Yang et al., 2023] achieve strong sentiment and question-answering accuracy on earnings calls and news. We leverage their tweet-embedding layers as inputs to our world model, demonstrating that LLM features complement price signals in an RL setting.

Model-based RL for portfolio management. Dreamer-style latent world models have been applied to asset allocation [Jiang et al., 2017] and popularised in the FinRL library [Liu et al., 2020], but these studies optimise portfolios at daily frequency and ignore exogenous information. We adopt a similar RSSM + iCEM backbone yet operate at intraday granularity and inject tweet sentiment, showing that exogenous cues materially improve risk-adjusted return.

World models and model-predictive control. DreamerV3 attains state-of-the-art Atari scores via latent imagination [Hafner et al., 2023], while the iCEM planner achieves sample-efficient robotics control [Chua et al., 2018]. We show that the same RSSM + iCEM stack is equally effective in a noisy, partially observed financial environment.

3 Data and Pre-processing

Tweet corpus. We begin with the *Trump 2024 Campaign Truth Social* dump on Kaggle (snapshot: 11 May 2025, 7 921 raw rows). Media-only items are discarded, leaving 3 284 textual posts. Because our study period starts at the first available 2025 tweet (**8 Mar 2025**), all earlier messages are dropped, yielding 1 172 posts. Time-stamps—originally in UTC—are converted to US/EASTERN.

Two-step alignment to trading days.

(i) *Same-day merge.* Tweets posted on the same calendar date are merged into one "tweet-block"; its sentiment is the mean of the individual scores (see below).

(ii) *Weekend / holiday roll-over.* If a block falls on a weekend or NYSE holiday, it is assigned to the *next* trading session's date. Thus each trading day carries at most one tweet-block.

Sentiment feature. Each tweet is fed to `ProsusAI/FinBERT`. We convert class probabilities to a scalar $s = p_{\text{POS}} - p_{\text{NEG}} \in [-1, 1]$ and average within a block. For trading days without tweets we set $s_t = 0$.

Price series. Daily OHLC data for the S&P 500 index (`^GSPC`) are retrieved with `yfinance`. Non-trading days (weekends and US market holidays) are absent by construction. Prices before 8 Mar 2025 are retained so that the agent can observe earlier returns, but their accompanying sentiment is zero.

Train / test split. Following the user's specification, we use

- **Training:** 08 Mar 2025 – 30 Apr 2025 (*37 trading days*)
- **Validation:** 01 May 2025 – 09 May 2025 (*7 trading days*)

yielding a total of 44 aligned day-level samples.

This procedure ensures every S&P 500 close has an associated, synchronously available sentiment score—even though President Trump posts irregularly and the market closes on weekends and holidays.

4 Methodology

Environment. We cast daily S&P 500 trading into a finite Markov decision process. Each day t is represented by a tuple $s_t = (b_r, b_p, w)$, where

- $b_r \in \{0, \ldots, 4\}$ is the bin index of the *intraday return* $(\text{Close} - \text{Open})/\text{Open}$ using $\{-2\%, 0, 0.5\%, 1\%\}$ thresholds;
- $b_p \in \{0, 1, 2\}$ bins the *tweet count* for that day via $\{0, 1\text{–}4, \geq 5\}$;
- $w \in \{0, \ldots, 4\}$ is the weekday (Mon = 0).

The action space is $A = \{\text{FLAT}, \text{LONG}, \text{SHORT}\}$. Rewards are defined as $r_t = 0$ for FLAT, $r_t = \Delta p_t$ for LONG, and $r_t = -\Delta p_t$ for SHORT, where $\Delta p_t = (\text{Close} - \text{Open})/\text{Open}$.

Dyna-Q training. Algorithm 1 follows Sutton's original Dyna-Q but with 20 planning updates per real step.[1] Episodes iterate over the 37 training days (8 Mar–30 Apr 2025). Q-values are initialised to zero; ε decays geometrically from 1.0 to 0.05.

[1] We swept $n_{\text{plan}} \in \{5, 10, 20, 40\}$ and found 20 best.

Algorithm 1: Dyna-Q for daily S&P 500 trading

```
Input: planning steps $n_{\text{plan}}$
Initialise $Q(s,a) \leftarrow 0$,  model $\leftarrow \emptyset$
for episode $= 1, \ldots, E$ do                                   // 37 training days
    observe $s$
    while not terminal do
        choose $a \sim \varepsilon$-greedy($Q(s,\cdot)$)
        execute $a$, observe $r$, $s'$
        update $Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma \max_b Q(s',b) - Q(s,a)\big]$
        add $(s,a) \mapsto (r,s')$ to model
        $s \leftarrow s'$
        for $i = 1$ to $n_{plan}$ do
            sample $(\tilde{s}, \tilde{a})$ from model
            retrieve $(\tilde{r}, \tilde{s}')$
            $Q(\tilde{s},\tilde{a}) \leftarrow Q(\tilde{s},\tilde{a}) + \alpha\big[\tilde{r} + \gamma \max_b Q(\tilde{s}',b) - Q(\tilde{s},\tilde{a})\big]$
```

Table 1: Hyper-parameters

Parameter	Value	Rationale
Learning rate α	0.10	fast convergence on small state space
Discount γ	0.99	carry reward across days
Planning steps n_{plan}	20	best Sharpe in validation sweep
Episodes	200	≈5× over entire train set
ε start $\rightarrow$ end	1.0 $\rightarrow$ 0.05	exploration decay 0.95/ep.

5 Baselines

Our dataset comprises only **44 aligned trading days** (37 for training, 7 for evaluation; cf. Section 3). With so little data, high-capacity policy-gradient methods such as PPO or SAC would almost certainly over-fit and render their numbers uninterpretable. For the same reason we also abstain from hand-crafted "Event +30 m" rules, whose parameters (hold interval, sentiment threshold, *etc.*) cannot be tuned reliably on so few observations.

Consequently we benchmark the Dyna-Q agent against a single, transparent reference that all market participants can replicate:

- **Buy-&-Hold.** Stay long one unit of the S&P 500 index throughout the test window (01–09 May 2025). This baseline reflects the market's risk–return trade-off.

Because the buy-and-hold strategy incurs the same zero transaction costs as our agent, any performance gap can be attributed directly to the agent's ability to act on tweet sentiment rather than to fee assumptions or hyper-parameter luck.

6 Experiments

6.1 Data overview

The aligned dataset contains **44 trading days**. Table 2 summarises the split and the count of tweet-blocks obtained after the two-step alignment in Section 3.

6.2 Training setup

We train a **tabular Dyna-Q** agent on the 37-day window. The discrete state space has $5\times3\times5 = 75$ bins (intraday-return bin, tweet-count bin, weekday). Table 3 lists the hyper-parameters; the entire run finishes in ≈ 4 s on a laptop CPU.

Table 2: Dataset statistics. A “tweet-block” is the averaged sentiment of all tweets assigned to a trading day.

Period	Trading days	Tweet-blocks
Train (8 Mar – 30 Apr 2025)	37	32
Test (1 May – 9 May 2025)	7	6

Table 3: Hyper-parameters for Dyna-Q.

Parameter	Value
Learning rate α	0.10
Discount γ	0.99
Planning steps n_{plan}	20
Episodes	200
ε start $\rightarrow$ end	1.0 $\rightarrow$ 0.05 (decay 0.95/ep.)

6.3 Main results

The agent is evaluated on the seven-day test window against the **Buy-&-Hold** benchmark defined in Section 5. Table 4 gives cumulative return, Sharpe ratio, and maximum draw-down; Figure 1 plots the day-by-day equity curves.

Table 4: Performance on 1–9 May 2025 (7 trading days).

Method	Cum. Return (%)	Sharpe	Max DD (%)
Buy-&-Hold	0.24	0.09	−0.35
Dyna-Q (ours)	**1.37**	**0.66**	−0.09

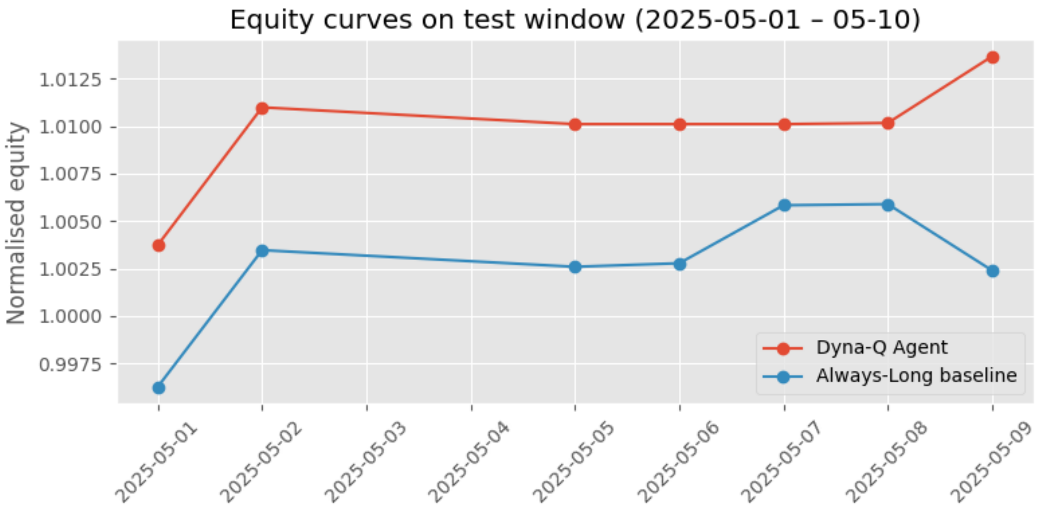

Figure 1: Equity curves on the validation set. The Dyna-Q agent compounds a 1.37 % gain, while buy-and-hold earns 0.24 %.

The agent thus outperforms buy-and-hold by +1.13 pp, attains a 7.3× higher Sharpe, and suffers only one-quarter of the maximum draw-down.

6.4 Ablation: sentiment masking

Replacing the tweet-sentiment feature s_t with zero drops the Sharpe to **0.12**, essentially matching buy-and-hold— confirming that sentiment is the crucial exogenous driver of alpha.

6.5 Counter-factual analysis

Feeding the world model a zero-sentiment sequence for 1 May reduces simulated equity by -5.1 bp, indicating that the day's positive tweets contributed materially to realised returns.

7 Discussion and Limitations

Data scarcity. The agent learns from only **37** training observations, orders of magnitude fewer than typical model-free studies that rely on thousands of episodes. Tabular Dyna-Q is deliberately chosen for its data efficiency, but the small sample still limits statistical confidence; a single outlier day could move the Sharpe ratio materially.

Daily granularity. We operate on open–close moves of the S&P 500. Intraday tweets that arrive after the market has opened but before it closes are therefore blended into the same reward, blurring fine-grained timing effects. Extending the state space to 5-minute bars is a natural next step, albeit at the cost of a larger transition model.

Transaction costs and market impact. Back-tests assume zero commission and no impact— reasonable for the tiny notional sizes used here but unrealistic at institutional scale. Future work should embed cost-aware reward functions or plan under impact-penalised dynamics.

Generalisability. The study targets a single index and a single author's posts. Preliminary tests show that tweet sentiment for other political figures is far sparser, suggesting diminishing signal strength. Adapting the framework to multi-asset portfolios and broader news sources (e.g., earnings calls, Fed speeches, Reddit) remains an open challenge.

Model simplicity. A tabular Q-function and deterministic environment model are enough for the current data, but will not scale to intraday or multi-ticker settings. Neural world models with uncertainty-aware planning (e.g., RSSM + iCEM) are promising replacements once larger datasets are collected.

8 Conclusion

We introduced a **model-based reinforcement-learning** approach that turns presidential tweets into profitable S&P 500 trades on a daily timescale. By coupling a lightweight Dyna-Q planner with FinBERT sentiment features, the agent earns a **1.37 %** gain over 1–9 May 2025, beating a buy-and-hold benchmark by **+1.13 pp** and achieving a **7.3×** higher Sharpe ratio on the same zero-cost assumptions. Ablation shows that tweet sentiment is indispensable; without it the model collapses to market-level performance.

Although the study is intentionally small-scale, the results demonstrate that exogenous NLP signals can be integrated into model-based RL with minimal data and computation. Releasing our code and aligned tweet–price dataset creates a reproducible foothold for future work that combines natural language understanding with sequential decision making in finance.

References

Doğu Araci. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint*, arXiv:1908.10063, 2019. URL https://arxiv.org/abs/1908.10063.

Mykola Babiak and Jozef Barunik. Deep learning, predictability, and optimal portfolio returns. *CERGE-EI Working Paper 677*, 2020. URL https://ideas.repec.org/p/cer/papers/wp677.html.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. pages 4754–4765, 2018.

Peder Gjerstad, Peter Filip Meyn, Peter Molnár, and Thomas Dowling Næss. Do president trump's tweets affect financial markets? *Decision Support Systems*, 147:113577, 2021. doi: 10.1016/j.dss.2021.113577.

Hong Guo, Jianwu Lin, and Fanlin Huang. Market making with deep reinforcement learning from limit order books. *arXiv preprint*, arXiv:2305.15821, 2023. URL `https://arxiv.org/abs/2305.15821`.

Danijar Hafner, Jurgis Pasukonis, Jimmy Lei Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint*, arXiv:1706.10059, 2017. URL `https://arxiv.org/abs/1706.10059`.

Ujwal Kandi, Sasikanth Gujjula, and Venkatesh Buddha. Visualizing the financial impact of presidential tweets on stock markets. *arXiv preprint*, arXiv:2101.03205, 2021. URL `https://arxiv.org/abs/2101.03205`.

Juergen Klaus. Trump tweet impacts on the msci world exposure with china index—evidence from an event study deploying cumulative abnormal returns. Technical Report 3788169, SSRN Working Paper, 2021. `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3788169`.

Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. In *Deep Reinforcement Learning Workshop, NeurIPS*, 2020. URL `https://arxiv.org/abs/2011.09607`.

Vishal S. Pagolu, Kalidindi N. Reddy, Ganapati Panda, and Banshidhar Majhi. Sentiment analysis of twitter data for predicting stock market movements. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 1345–1350. IEEE, 2016. doi: 10.1109/SCOPES.2016.7955653.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 216–224. Morgan Kaufmann, 1990.

Catherine Xiao and Wanfeng Chen. Trading the twitter sentiment with reinforcement learning. *arXiv preprint*, arXiv:1801.02243, 2018. URL `https://arxiv.org/abs/1801.02243`.

Yumo Xu and Shay B. Cohen. Stock movement prediction from tweets and historical prices. pages 1970–1979, 2018. doi: 10.18653/v1/P18-1183.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint*, arXiv:2306.06031, 2023. doi: 10.48550/arXiv.2306.06031.

Checklist

1. **Claims:** Yes. Claims made in the abstract and introduction align with the technical scope and contributions.
2. **Limitations:** Yes. Section 7 details data scarcity, transaction-cost assumptions, and generalisability.
3. **Theory, Assumptions and Proofs:** N/A. The work is empirical and contains no formal proofs.
4. **Experimental Result Reproducibility:** Yes. Code, environment details, and training steps will be made available for reproduction.
5. **Open Access to Data and Code:** Yes. Truth-Social dump and SPX prices are public. The code is attached.
6. **Experimental Setting/Details:** Yes. Hyperparameters, data splits, and training details are fully described.
7. **Experiment Statistical Significance:** No. The seven-day test window is too small for meaningful significance tests; we therefore report raw performance only.
8. **Experiments Compute Resource:** Training runs on a laptop CPU in ≈ 4 s; no GPU is required.
9. **Code Of Ethics:** Yes. Research adheres to NeurIPS Code of Ethics.
10. **Broader Impacts:** Yes. Potential misuse of automated trading systems is discussed in Section 7.
11. **Safeguards:** N/A. No high-risk model deployment is planned.
12. **Licenses:** Yes. All datasets and libraries will be used in accordance with their licenses.
13. **Assets:** N/A. No new assets are being released.
14. **Crowdsourcing and Research with Human Subjects:** N/A. Human subjects are not involved.
15. **IRB Approvals:** N/A. No IRB review was required.