CLaC @ FinMMEval 2026 Task 3: Sentiment-Augmented Deep Reinforcement Learning for Active Trading – An Alpha-Reward Approach

Andrei Neagu*, Eeham Khan and Leila Kosseim

Department of Computer Science and Software Engineering, Concordia University, Montréal, Canada

Abstract

This paper presents our system for Task 3 of the CLEF 2026 FinMMEval Lab, which requires participants to submit a daily trading decision (LONG, FLAT, or SHORT) for Bitcoin (BTC) and Tesla (TSLA) based on news articles and historical market pricing data. We frame the problem as a discrete-action Markov Decision Process and train four Deep Reinforcement Learning (DRL) algorithms: Policy Gradient (PG), Proximal Policy Optimization (PPO), Deep Q-learning (DQL), and Deep Deterministic Policy Gradient (DDPG), using a rich feature set that combines technical indicators (EMA, RSI, MACD, BollingerB, volume change), cyclical date encodings, and daily sentiment scores derived from news articles using LLaMA 3.2 1B. To reduce overfitting and align the training objective with the goal of outperforming a buy-and-hold baseline, we introduce an *alpha reward* that replaces the raw log-return with excess return over the market, and we randomize episode start days during training. Hyperparameters are optimized over 180 trials per algorithm-asset pair using Ray Tune, with model selection based on validation Sharpe ratio (SR) and early stopping. Evaluation on the CLEF Task 3 test set demonstrates that DDPG yields the best performance across both assets. Note, however, that DQL was selected a priori for the live competition endpoint based on its highest validation Sharpe ratio; the endpoint model was deliberately selected blind to the test period to avoid selection bias. For TSLA, DDPG and DQL achieved cumulative returns of 54.96% and 52.62% respectively, substantially beating the 16.45% buy-and-hold baseline. On the more challenging BTC test set, DDPG mitigated severe market losses, achieving a positive return of 1.58% against a baseline decline of -34.27%. Results reveal a pronounced validation-to-test generalization gap, pointing to the difficulty of adapting policies trained in a bull market validation period to bear market test conditions.

Keywords

Deep Reinforcement Learning, Algorithmic Trading, Sentiment Analysis, Large Language Models, Policy Gradient, Proximal Policy Optimization, Deep Deterministic Policy Gradient, Deep Q-Learning

1. Introduction

Financial markets are among the most complex sequential decision-making environments, combining noisy price dynamics and information sources such as global and asset specific news and economic indicators. Traditional quantitative strategies rely on hand-crafted rules or statistical models that struggle to capture non-linear interactions in multi-modal settings. Deep Reinforcement Learning (DRL) methods offer an alternative better suited to learning an optimal trading strategy adapted to such interactions. In DRL, an agent learns a policy that maps market observations directly to trading actions by maximizing a cumulative reward signal [1].

Task 3 of the CLEF 2026 FinMMEval Lab [2, 3, 4] aims at exploring dynamic market environments through real-time evaluation. Participants provide a web endpoint that receives daily market context consisting of prices, news articles, momentum, and optional SEC filings, and must respond with one of three actions: LONG, FLAT, or SHORT. Positions are updated daily and performance is assessed by cumulative return (CR), Sharpe ratio (SR), maximum drawdown (MDD), and volatility on two assets: Bitcoin (BTC) and Tesla (TSLA).

Our work tackles this challenge using four DRL algorithms trained and validated on historical data spanning from 2010 to 2024. We make three principal contributions:

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author: andrei.neagu@concordia.ca

andrei.neagu@concordia.ca (A. Neagu); eeham.khan@concordia.ca (E. Khan); leila.kosseim@concordia.ca (L. Kosseim)

1. We demonstrate that a multi-modal feature representation combining price-derived technical indicators, LLM-based daily news sentiment scores, and cyclical calendar encodings provides an effective state signal for DRL-based trading agents.
2. We introduce and analyze an alpha reward formulation that explicitly trains agents to outperform the *buy-and-hold* baseline rather than to maximize absolute portfolio value, and show that it (i) equals log-alpha when summed over an episode, (ii) shares optimal policies with the raw log-return reward under exogenous market dynamics, and (iii) acts as a variance-reducing control variate on the reward signal. Combined with random episode starts, our alpha reward reduces overfitting and aligns the training objective with the evaluation criterion.
3. We provide a comprehensive comparison of four DRL algorithms (PG, PPO, DQL, and DDPG) with hyperparameter optimization and Sharpe-based early stopping, and show that DDPG achieves the most robust performance across both assets while validation-based model selection remains sensitive to regime shifts between bull and bear markets.[1]

2. Related Work

Early work applying Reinforcement Learning (RL) to market trading dates back to Moody et al. [5], who proposed the use of recurrent neural networks trained on cumulative return. Mnih et al. [1] later showed that Deep Q-Networks (DQN) could match or surpass human performance in Atari games, inspiring a wave of applications to financial markets. Jiang et al. [6] formulated cryptocurrency portfolio management as a continuous-action Deep Reinforcement Learning (DRL) problem, showing that an LSTM-based deterministic Policy Gradient (PG) agent could outperform traditional statistical rebalancing strategies.

Actor-critic methods have also seen growing adoption in trading. Building on foundational actor-critic architectures, Mnih et al. [7] introduced Asynchronous Advantage Actor-Critic (A3C), which stabilizes learning and improves sample efficiency by deploying multiple parallel workers to interact with their own environments asynchronously. To simplify this architecture, a synchronous variant, Advantage Actor-Critic (A2C), was then developed to coordinate parallel workers and update the global network synchronously while maintaining comparable performance. Additionally, Schulman et al. [8] introduced Proximal Policy Optimization (PPO), a stable on-policy method widely adopted for financial applications due to its clipped surrogate objective. Lillicrap et al. [9] proposed Deep Deterministic Policy Gradient (DDPG), which extends Deep Q-Learning (DQL) to continuous action spaces via a deterministic policy with soft target updates. Liu et al. [10] applied DDPG to stock portfolio management, and demonstrated that it outperformed the Dow Jones buy-and-hold strategy in backtesting.

Following these results, Yang et al. [11] employed an ensemble approach, combining PPO, DDPG, and A3C, to trade a portfolio comprised of the 30 constituent stocks of the Dow Jones Industrial Average. The authors constantly evaluate each individual algorithm, switching to the best performing one after N steps. They showed that, while their ensemble method produces a higher Sharpe ratio (SR), PPO achieves a higher cumulative return. Ye et al. [12] have also employed an ensemble method using the DDPG, A2C and PPO algorithms, but only switch to the best performing algorithm when sentiment scores exceed a certain threshold. Their results show that this selection method outperforms the N-step selection method. More recently, Du and Shen [13] employed DQL with sentiment analysis for trading a portfolio comprised of Chinese stocks, finding that their method outperforms both standard DQL and traditional statistical methods.

The integration of natural language information into trading agents has gained significant interest due to the increasing availability of pre-trained and large language models (LLMs). Araci [14] proposed FinBERT, a BERT model fine-tuned on financial corpora that produces sentiment scores that can be used as features in prediction models. Recent research has also explored the potential for LLMs to act

[1] All code, trained model checkpoints, hyperparameter configurations, and data preprocessing scripts (including the regex filters, deduplication, and sentiment-scoring pipeline) will be released upon paper acceptance to ensure full reproducibility.

as financial decision-makers [15, 16]. Our work follows this line of research, as we use an open-source LLM for article-level sentiment scoring combined with DRL algorithms for action selection.

3. Background

We model the trading problem as a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$ [17], where:

- $\mathcal{S}$ is the state space: a vector of market features observed at each trading day.
- $\mathcal{A} = \{\text{SHORT } (-1),\ \text{FLAT } (0),\ \text{LONG } (1)\}$ is the discrete action space.
- $\mathcal{R}(s, a, s')$ is the reward function associated with taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and transitioning to the next state $s' \in \mathcal{S}$.
- $\mathcal{P}(s'|s, a)$ is the (unknown) market transition probability, representing the stochastic evolution of the financial environment. We assume the agent's actions do not influence future market states (price-taker).
- $\gamma \in [0, 1)$ is the discount factor, determining the present value of future rewards and ensuring that the expected return G_t remains finite.

At each time step $t \in [1, T]$, the agent observes state s_t, selects action $a_t \sim \pi_\theta(s_t)$, transitions to s_{t+1}, and receives a reward r_t. The objective is to find the policy π_θ that maximizes the expected discounted return $G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$.

To translate these discrete actions into financial performance, we evaluate the agent based on a portfolio that evolves multiplicatively. At each time step t, the agent's previously chosen action dictates its current market position $p_t \in \{-1, 0, 1\}$ (LONG, FLAT, or SHORT). As the market transitions, the agent observes the daily price return $return_t = (c_t - c_{t-1})/c_{t-1}$, where c_t is the closing price on day t. Consequently, the portfolio value V_t updates as follows:

$$V_t = V_{t-1} \cdot (1 + p_{t-1} \cdot return_t) \cdot \left(1 - \delta \cdot \mathbb{1}_{\{a_t \neq p_{t-1}\}}\right) \tag{1}$$

where $\delta = 0.002$ is the transaction cost, and $\mathbb{1}_{\{a_t \neq p_{t-1}\}}$ is an indicator function that equals 1 when the agent's new action a_t differs from its previous position p_{t-1}, i.e. when a trade is executed, and 0 otherwise, so that the cost is incurred only on position changes.

4. Methodology

Figure 1 illustrates the overall architecture of our proposed trading framework. The system processes historical pricing and textual news data (see Section 4.1), extracts daily sentiment using a zero-shot LLM (see Section 4.3), and constructs a 15-dimensional state feature vector (see Section 4.2). This state representation informs a custom Markov Decision Process trading environment, where one of four DRL agents is trained to maximize a market-relative alpha reward (see Section 4.5).

4.1. Data

External Data. In order to supplement the provided training data, we augment it with externally sourced data. Daily closing prices and trading volumes are downloaded via the Yahoo Finance (`yfinance`[2]) library. For TSLA, the data spans 2,835 trading days from 2010-06-29 to 2022-12-30, covering the asset's entire history up to the validation cutoff. For BTC-USD, the data covers 2,664 trading days from 2014-09-17 (the earliest date available on Yahoo Finance) to 2022-12-30. We align each trading day with news articles from Brian Ferrell's Financial News Multi-Source Hugging Face dataset[3] [18]. The official CLEF Task 3 evaluation set is drawn from the CLEF Task 3 Hugging Face[4]

[2] https://pypi.org/project/yfinance/
[3] https://huggingface.co/datasets/Brianferrell787/financial-news-multisource
[4] https://huggingface.co/datasets/TheFinAI/CLEF_Task3_Trading

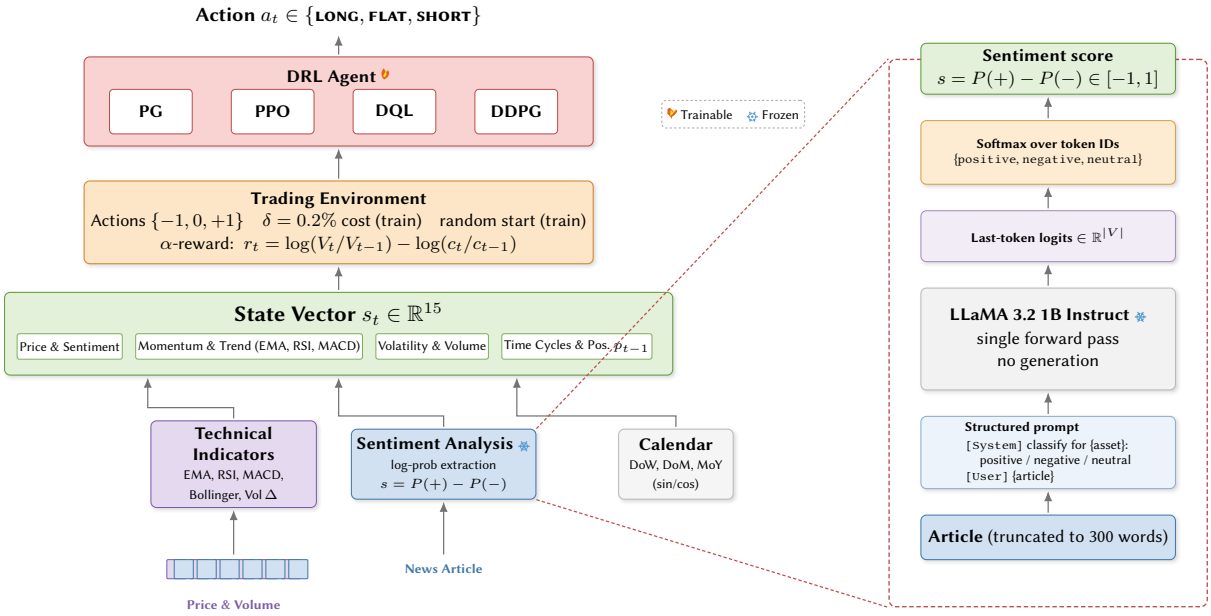


Figure 1: System overview. **Left:** three parallel processing streams, technical indicators derived from price and volume (purple) and daily news sentiment from a frozen LLaMA-3.2-1B Instruct model (blue), together with cyclical calendar encodings, compose a 15-dimensional state vector s_t. The state is fed as input by a custom trading environment whose α-reward incentivises outperforming the buy-and-hold baseline, and by one of four DRL agents (PG, PPO, DQL, DDPG) that outputs a daily action. **Right:** Detail of the sentiment scoring module, which performs a single forward pass of the LLaMA backbone model, extracts the logits at the final token position, and applies a softmax restricted to the token IDs for {positive, negative, neutral}, yielding a bounded score $s \in [-1, 1]$.

dataset [3], which provides daily closing prices and pre-linked news for the testing phase. Volume data taken from Yahoo Finance is then injected into this data split to compute volume-based features.

To isolate asset-specific information, news articles are filtered using regular expressions. We retain articles that contain at least two mentions of the target asset's keywords (*Tesla*/*TSLA* or *Bitcoin*/*BTC*), or at least one mention within the first 100 words (the lede). Following standard journalistic structure, an early mention likely indicates the asset is the primary subject of the article. This constraint is imposed to prevent the sentiment classifier from scoring noisy texts where the target asset is not a major detail.

The text is then normalized only for the purposes of filtering and removing duplicates by converting to lowercase, removing punctuation, and stripping excess whitespace. We remove duplicates from the dataset by computing an MD5 hash of this normalized text; when duplicates are found, priority is given to records containing explicit publisher metadata. To preserve the semantic nuance, financial metrics, and tone required for accurate LLM inference, we retain the original unnormalized text for the sentiment classification step. Finally, we structure this input text by prepending available publisher information (e.g., `[Publisher: Name] Article Text`) to provide structured context for the sentiment model.

Temporal Alignment and Aggregation. All article timestamps are in UTC. To prevent look-ahead bias on TSLA, articles published after 16:00 UTC (i.e., after the U.S. equity market close) are shifted and linked to the following trading day. For BTC, which trades 24/7, we instead anchor each trading day to its UTC calendar date and link every article timestamped on day t to day t's closing price. Because trading decisions for day $t+1$ are conditioned strictly on information available at or before the close of day t, no look-ahead is introduced in either setting.

Following the article-level sentiment scoring (detailed in Section 4.3), the data is aggregated on a daily basis. For each trading day, we calculate the mean sentiment score across all relevant articles and record the final closing price and volume. Lastly, standard percentage returns and log returns are computed to serve as inputs for the DRL environment.

Data Splits. We partition the augmented dataset chronologically into three distinct splits: training (prior to January 1, 2023), validation (January 1, 2023 to August 1, 2024), and testing (the official CLEF Task 3 evaluation period). This allocation yields 324 validation days for TSLA and 457 for BTC. To eliminate cold-start artifacts from our rolling-window technical indicators (e.g., EMA_{50}), the first 60 days of the training set are used only for indicator initialization and subsequently discarded. Similarly, to ensure the test set receives fully warmed-up features on its very first day, we prepend the final 60 days of the validation set to the test data prior to computing the indicators, truncating these warmup rows before evaluation.

4.1.1. Market Regime Analysis

The three temporal splits differ substantially in their market dynamics, which is central to interpreting the generalisation results in Section 6. Figures 2a and 2b show the prices (in log scale) over time for TSLA and BTC respectively. Identified are also the training, validation and test splits. Table 1 summarises statistics for each split and asset.

Figure 2: Asset prices (in log scale) across time and data splits.

(a) TSLA prices (in log scale) across time and data splits.

(b) BTC prices (in log scale) across time and data splits.

As shown in Figure 2, the *training period* encompasses multiple complete market cycles for both assets. TSLA's history includes the rapid growth phase of 2020–2021 (share price rising from \$30 to a peak of \$407), followed by a severe correction throughout 2022 (-65%). BTC experienced the 2017 speculative bubble and subsequent crash throughout 2018 (-83%), the COVID-19 drawdown in March 2020, the 2021 all-time high near \$69,000, and the 2022 bear market (-64% year-over-year). The resulting high annualised volatility (60–65%) (see Table 1) and long-time horizons expose agents

Table 1
Market statistics across data splits. Ann. Ret., Ann. Vol., and SR are annualised; Cumul. Ret. is the buy-and-hold return over the full split. The Sharpe ratio (SR) uses a zero risk-free rate. Article counts for the test split are provided live by the competition.

Asset	Split	Period	Days	Articles	Cumul. Ret. (%)	Ann. Ret. (%)	Ann. Vol. (%)	SR
TSLA	Training	2010–2022	2835	57,741	7,634.2	47.2	60.3	0.94
	Validation	2023–2024	324	7,089	105.9	75.4	60.2	1.24
	Test†	2025–	256	—	16.5	16.2	34.9	0.61
BTC	Training	2014–2022	2664	29,307	3,518.3	40.4	65.0	0.85
	Validation	2023–2024	457	2,861	288.7	111.4	43.3	1.95
	Test†	2025–	256	—	−34.3	−33.8	39.1	−0.87

† Measured to April 12th 2026; test set is continuously updated.

to a diverse set of market conditions, providing broad coverage of both trending and mean-reverting regimes.

The *validation period* (2023–2024) coincides with a sustained bull market recovery for both assets. BTC rebounded from its 2022 lows and reached new all-time highs, producing a cumulative return of 288.7% over the period. TSLA also recovered strongly, yielding 105.9%. The uniformly positive and high-momentum nature of the validation environment means that model selection based on validation Sharpe ratio may favour policies that have learned to hold long positions persistently, potentially at the expense of robustness to downturns.

The *test period* (from August 2025) presents a divergent regime for the two assets. TSLA continued its general upward trend at markedly reduced volatility (34.9% annualised versus 60.2% in validation), yielding a 16.5% cumulative return (annualised SR of 0.61), well below the validation SR of 1.24. BTC, by contrast, entered a bear phase with a -34.3% cumulative return (annualized SR of -0.87), representing a sharp distributional shift from the 288.7% bull market validation environment (SR of 1.95). This regime divergence is the primary driver of the generalisation gap discussed in Section 6, particularly for BTC, where policies selected for strong bull market performance were exposed to sustained downward price pressure not well represented in the validation period.

4.2. Feature Representation

Table 2 lists the 15-dimensional state vector presented to each RL agent on each trading day. All technical indicators are normalized to be approximately zero-centered. We describe the rationale for each feature group below.

Price (LogReturn). The daily log return, computed as $\log(c_t/c_{t-1})$, provides a stationary, scale-invariant measure of the asset's immediate price movement over the last 24 hours. In quantitative finance, log returns are preferred over simple percentage returns because they are time-additive and symmetric, meaning positive and negative movements of the same magnitude cancel each other out mathematically. This feature anchors the agent to the most recent, immediate price action before the slower moving averages and momentum oscillators have time to react.

Agent State (Position). The agent's position from the previous time step, $p_{t-1} \in \{-1, 0, 1\}$, represents its internal state. Providing the agent with its own prior action is critical in environments with transaction fees. Without knowing its current portfolio allocation, the agent cannot evaluate the penalty of executing a new trade versus holding an existing position. Including this feature enables the DRL policy to learn "inertia", avoiding excessive trading churn and only triggering a state change when the expected market return exceeds the 0.2% transaction cost barrier.

Table 2
State feature vector (14 market dimensions + previous position = 15 total inputs).

	#	Category	Feature	Formula	Range
	1	Price	LogReturn	$\log(c_t/c_{t-1})$	$\mathbb{R}$
	2	Agent State	Position	p_{t-1}	$\{-1, 0, 1\}$
Technical Indicators	3	Calendar	DoW_sin	$\sin(2\pi \cdot \text{DoW}/7)$	$[-1, 1]$
	4		DoW_cos	$\cos(2\pi \cdot \text{DoW}/7)$	$[-1, 1]$
	5		DoM_sin	$\sin(2\pi \cdot \text{DoM}/31)$	$[-1, 1]$
	6		DoM_cos	$\cos(2\pi \cdot \text{DoM}/31)$	$[-1, 1]$
	7		MoY_sin	$\sin(2\pi \cdot \text{MoY}/12)$	$[-1, 1]$
	8		MoY_cos	$\cos(2\pi \cdot \text{MoY}/12)$	$[-1, 1]$
	9	Trend	EMA_{10}	$\text{EMA}_{10}/c_t - 1$	$\mathbb{R}$
	10		EMA_{50}	$\text{EMA}_{50}/c_t - 1$	$\mathbb{R}$
	11	Momentum	RSI_{14}	$\text{RSI}_{14}/50 - 1$	$\mathbb{R}$
	12		MACD	$(\text{EMA}_{12} - \text{EMA}_{26} - \text{signal})/c_t$	$\mathbb{R}$
	13	Volatility	BollingerB	$(c_t - \text{SMA}_{20})/(2\sigma_{20})$	$\mathbb{R}$
	14	Volume	LogVolumeChange	$\log(V_t/V_{t-1})$	$\mathbb{R}$
	15	Sentiment	Sentiment	$p_+ - p_-$ (LLaMA 3.2 1B)	$[-1, 1]$

Calendar Features (Cyclical Encodings). The six sinusoidal encodings in Table 2 represent day-of-week (DoW), day-of-month (DoM), and month-of-year (MoY) as sine/cosine pairs, providing the agent with a continuous, cycle-aware representation of calendar time. Financial markets exhibit well-documented intra-week and intra-month seasonality: the day-of-week effect [19] and turn-of-the-month patterns in equity returns are two examples. Encoding calendar variables as sine/cosine pairs rather than as raw integers preserves the cyclic continuity and avoids imposing an ordinal ranking on categorical time features.

4.2.1. Technical Indicators

Trend Indicators (EMA_{10}, EMA_{50}). Exponential Moving Averages (EMAs) assign exponentially decaying weights to past prices, making them more responsive to recent changes than simple moving averages [20]. EMA_{10} captures short-term momentum over approximately two trading weeks, while EMA_{50} reflects the medium-term trend over roughly 2.5 months. Expressing each EMA relative to the current close ($\text{EMA}/c_t - 1$) produces a scale-invariant feature that is directly comparable across assets with vastly different price levels (e.g. BTC at \$60,000 versus TSLA at \$300). The combination of a fast and a slow EMA also encodes an implicit cross-signal: when $\text{EMA}_{10} > \text{EMA}_{50}$, price is in a short-term uptrend relative to the medium-term baseline, a classical bullish signal [21].

Momentum Indicators (RSI_{14}, MACD). The Relative Strength Index (RSI_{14}) [22] is a bounded oscillator computed as the ratio of average gains to average losses over a 14-day lookback. We re-centre it from $[0, 100]$ to $[-1, 1]$ via $\text{RSI}_{14}/50 - 1$, so that the neutral level is zero. Values above 0.4 ($\text{RSI}_{14} > 70$) signal overbought conditions associated with potential reversals, while values below -0.4 ($\text{RSI}_{14} < 30$) signal oversold conditions, providing a mean-reversion counterpoint to the trend-following EMA features.

Moving Average Convergence Divergence (MACD) [23] measures the histogram between the MACD line ($\text{EMA}_{12} - \text{EMA}_{26}$) and its 9-day exponential signal line, normalised by the closing price to remove scale dependence. The MACD histogram captures trend acceleration and deceleration, providing a leading signal for potential reversals before they are evident in price alone. Normalisation by c_t ensures the feature remains stationary across different price regimes.

Volatility Indicator (BollingerB). BollingerB [24] positions the current price within a band defined as two standard deviations above and below a 20-day simple moving average (SMA_{20}):

$$\text{BollingerB} = \frac{c_t - \text{SMA}_{20}}{2\sigma_{20}}.$$

Unlike RSI, whose thresholds are fixed regardless of volatility regimes, BollingerB adapts its thresholds dynamically to the prevailing volatility regime, making it particularly informative for assets like BTC that alternate between low-volatility consolidation periods and high-volatility breakouts. Values near ± 1 indicate price at the band extremes and are commonly associated with reversal or continuation signals depending on the volume context.

Volume Indicator (LogVolChange). The log ratio of consecutive daily volumes $\log(V_t/V_{t-1})$ captures day-over-day shifts in market participation. Trading volume is widely recognized as a confirmation signal: price movements accompanied by abnormally high volume are considered more reliable indicators of trend continuation than those occurring on thin volume [25]. Using a log ratio rather than raw volume renders the feature invariant to the absolute scale of trading activity, which differs by orders of magnitude between TSLA (equity market) and BTC (24/7 cryptocurrency market).

4.3. Sentiment Analysis

As shown in Figure 1, we score daily news sentiment using the LLaMA 3.2 1B Instruct model [26] in a zero-shot setting. We selected this model because modern instruction-tuned LLMs demonstrate strong zero-shot reasoning capabilities that often match or exceed older domain-specific models (e.g., FinBERT), while its 1B parameter footprint ensures computational efficiency for daily inference. To capture the core market signal and avoid diluting the sentiment with irrelevant body content, each article is truncated to its first 300 words, in order to focus on the headline and the lede. The truncated text is then wrapped in a structured conversational prompt:

> **[System]** *You are a financial sentiment classifier for {asset}. Respond with exactly one word: positive, negative, or neutral.*
>
> **[User]** *{article text}*

Rather than relying on standard autoregressive text generation, which can be computationally expensive and subject to formatting hallucinations, we employ a log-probability extraction approach. We perform a single forward pass through the model and extract the logits at the final token position corresponding to the model's output. To avoid brittleness to a single tokenization, for each class label we enumerate the BPE token-ID variants with and without a leading space, and with lower- and title-case (e.g., `positive`, `␣positive`, `Positive`, `␣Positive`). A softmax is then applied strictly over this union of token IDs, and the resulting probabilities are summed per class to yield $P(\text{positive})$, $P(\text{negative})$, and $P(\text{neutral})$.

The final sentiment score for a given article is calculated as the difference between the aggregated probabilities of the positive and negative classes:

$$Sentiment = P(\text{positive}) - P(\text{negative}) \tag{2}$$

Although the neutral class probability $P(\text{neutral})$ does not appear explicitly in this equation, it plays an important structural role. Because the values are derived from a softmax distribution where $P(\text{positive}) + P(\text{negative}) + P(\text{neutral}) = 1$, a high probability assigned to the neutral class restricts the available probability mass for the polar classes. Therefore, a strong neutral prediction acts as an automatic confidence weight, dampening the final sentiment score and driving it toward zero. This formulation yields a bounded continuous score in the range $[-1, 1]$.

4.4. Alpha Reward

An important design choice is the *alpha reward*: instead of rewarding absolute portfolio log-return, we reward excess return over the market:

$$r_t = \log\left(\frac{V_t}{V_{t-1}}\right) - \log\left(\frac{c_t}{c_{t-1}}\right). \tag{3}$$

Under a passive buy-and-hold strategy with no costs, every step yields $r_t = 0$. The agent is therefore incentivised to outperform the market rather than merely profit from rising prices, directly aligning the training objective with the evaluation criterion. Model selection on the validation set uses the checkpoint with the highest Sharpe ratio computed on raw portfolio values, matching the task's evaluation metric.

Theoretical Properties of the Alpha Reward. Beyond its interpretation as market-relative performance, the alpha reward has three properties worth noting. Let $m_t = \log(c_t/c_{t-1})$ denote the market log-return, so that $r_t = \log(V_t/V_{t-1}) - m_t$.

1. Cumulative alpha identity. Over an undiscounted episode of length T, the per-step rewards telescope to:

$$\sum_{t=1}^{T} r_t \;=\; \log\left(\frac{V_T}{V_0}\right) - \log\left(\frac{c_T}{c_0}\right) \;=\; \log(\alpha_T), \tag{4}$$

where $\alpha_T = (V_T/V_0)/(c_T/c_0)$ is the terminal outperformance ratio over buy-and-hold. The return-to-go G_t is therefore the log-alpha of the strategy, directly aligning the training signal with the CLEF Task 3 evaluation criterion.

2. Equivalent optima to the raw reward. Under the price-taker assumption (see Section 3), the market return sequence $\{m_t\}$ is independent of the policy π_θ. For any discount factor $\gamma \in [0, 1]$, the alpha-reward objective therefore differs from the raw log-return objective only by a policy-independent constant $\mathbb{E}[\sum_t \gamma^{t-1} m_t]$, and the two share the same optimal policies. This is a weaker condition than the potential-based shaping of [27], but suffices here because m_t depends only on the state transition.

3. Variance reduction. Although the optima coincide, the learning dynamics do not. For a long position with no transaction cost, $\log(V_t/V_{t-1}) = m_t$, so the raw reward has variance $\mathrm{Var}(m_t)$ whereas the alpha reward has variance zero; the agent receives a clean signal about its deviation from buy-and-hold rather than a noisy signal dominated by market moves. More generally, subtracting m_t acts as a control variate on the reward, reducing gradient-estimator variance with no bias cost. On assets with high market volatility, TSLA and BTC both exceed 60% annualised training volatility (see Table 1), this variance reduction is substantial.

4.5. DRL Algorithms

We compare four DRL algorithms that span the main families of policy- and value-based approaches.

Policy Gradient (PG). PG [28] estimates the policy gradient as $\nabla_\theta J = \mathbb{E}[\nabla_\theta \log \pi_\theta(a|s) \cdot G]$ using Monte-Carlo returns G normalized within each episode for variance reduction. An entropy bonus $\beta H(\pi)$ discourages premature collapse to a deterministic policy. We parameterized π_θ as an LSTM network to capture temporal dependencies across the trading sequence.

Proximal Policy Optimization (PPO). PPO [8] improves stability over vanilla PG by clipping the probability ratio:

$$r_t(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \tag{5}$$

within $[1 - \varepsilon, 1 + \varepsilon]$. We use an LSTM actor-critic network with Generalized Advantage Estimation (GAE-λ) and truncated back-propagation through time (TBPTT) to manage gradient flow over long sequences.

Deep Q-Learning (DQL). DQL [1] learns an action-value function $Q(s, a; \theta)$ via temporal-difference learning with experience replay and a target network. We adopt the double DQL [29] update to reduce overestimation bias. The network is a feed-forward architecture (FFNN) since the replay buffer breaks temporal ordering, making LSTM less appropriate.

Deep Deterministic Policy Gradient (DDPG). DDPG [9] extends DQL to continuous action spaces with a deterministic actor $\mu_\theta(s)$ and a critic $Q_\phi(s, a)$. We discretize the continuous output to $\{-1, 0, 1\}$ via thresholds: $a < -0.33 \Rightarrow$ SHORT, $a > 0.33 \Rightarrow$ LONG, else FLAT. The threshold $1/3$ partitions the actor's output range $[-1, 1]$ into three equal-width intervals, giving each discrete action equal prior volume and avoiding an implicit bias toward any of the three positions. Exploration is driven by decaying Ornstein-Uhlenbeck noise. Like DQL, the actor and critic are FFNNs due to the use of a replay buffer.

5. Experimental Setting

5.1. Hyperparameter Optimisation

We use Ray Tune to run random search over 180 configurations per algorithm-asset pair (8 combinations) on a 192-core CPU cluster (Trillium, Digital Research Alliance of Canada), running all 180 trials in parallel. The search spaces are summarized in Table 3. All trials train for up to 2000 episodes with validation every 50 episodes; a trial terminates early if the validation SR does not improve for 10 consecutive evaluations. The configuration with the highest validation performance is selected.

Table 3
Hyperparameter search space.

Parameter	Values
hidden_dim	$\{64, 128, 256\}$
num_layers	$\{1, 2, 3\}$
γ	$\{0.95, 0.99\}$
lr	$[10^{-4}, 10^{-2}]$
entropy coeff (PG/PPO)	$[10^{-3}, 5 \cdot 10^{-2}]$
clip ε (PPO)	$\{0.1, 0.2, 0.3\}$
τ (DDPG)	$\{0.001, 0.005, 0.01\}$

5.2. Training Details

We implement a custom trading environment modeled after the OpenAI Gym interface. Each training episode starts at a randomly sampled day (with at least 100 days remaining) forcing the agent to encounter diverse market conditions rather than memorizing a fixed sequence. During evaluation, the episode always begins from the first day of the period.

Hyperparameter optimization uses a fixed train/validation split: agents are trained on data prior to January 1st 2023 and evaluated on data between January 2023 and August 2024. Once the best configuration is identified, each model retrains from scratch on the same training split (pre-2023) with early stopping based on the validation SR (patience 10). The best validation SR checkpoint from this final run is saved and used for all reported test results. We apply L2 regularization, with weight decay 10^{-4}, to all Adam optimizers and gradient clipping norm 1.0.

Transaction costs of 0.2% are applied during training only, causing the policy to internalize an inertia incentive, while evaluation is cost-free to exactly match the CLEF Task 3 submission protocol, which itself does not deduct execution costs. We acknowledge that this difference in transaction costs application reflects the task's evaluation convention and that realistic deployment would require

re-introducing costs at inference; we view this as a limitation of external validity rather than of the method.

The alpha reward and random episode start are enabled during training only. Validation and test evaluation always start from the first day of the respective period. Additionally, model selection uses the Sharpe ratio computed on the raw portfolio value series (matching how test SR is defined). Our test backtest mirrors the CLEF live-submission protocol (daily actions, cost-free, identical metric definitions); the only procedural difference is that we evaluate on a fixed snapshot of the test window rather than the continuously-updated leaderboard window.

6. Results

Table 4 reports the final performance of each trained agent on the CLEF Task 3 test set. The TSLA buy-and-hold baseline for the test period is 16.45% (SR 0.61, MDD 29.93); the BTC baseline is -34.27% (SR -0.87, MDD 49.72).

Table 4
Test set return and risk metrics on CLEF Task 3. Test CR = cumulative return (%), Test SR = Sharpe ratio, Val CR and Val SR are measured on the validation set, MDD = maximum drawdown (%), DV = daily volatility (%), AV = annualised volatility (%). Model selection uses the highest Val SR.

Asset	Model	Test CR (%)	Test SR	Val CR (%)	Val SR	MDD (%)	DV (%)	AV (%)
TSLA	B&H	16.45	0.61	105.94	1.24	29.93	2.19	34.80
	PG	13.96	0.55	101.76	1.21	29.93	2.19	34.74
	PPO	2.54	0.24	169.50	1.75	28.38	**2.01**	**31.93**
	DQL	52.62	1.38	**830.30**	**3.29**	35.76	2.19	34.69
	DDPG	**54.96**	**1.44**	262.10	2.03	**19.20**	2.15	34.09
BTC	B&H	−34.27	−0.87	**288.69**	1.95	49.72	2.46	39.00
	PG	−14.53	−0.33	109.33	1.43	**29.98**	**1.98**	**31.51**
	PPO	−33.80	−0.85	190.07	1.58	59.18	2.46	39.00
	DQL	−23.22	−0.55	276.71	**2.11**	37.41	2.25	35.72
	DDPG	**1.58**	**0.23**	259.53	1.87	35.30	2.40	38.12

For TSLA, as detailed in Table 4, both DDPG (CR 54.96%, SR 1.44) and DQL (CR 52.62%, SR 1.38) outperform the buy-and-hold (B&H) baseline (CR 16.45%, SR 0.61), demonstrating that RL agents can learn to time the TSLA market effectively. PG (CR 13.96%, SR 0.55) falls just below the baseline, and PPO (CR 2.54%, SR 0.24) trails behind significantly despite a large validation return (Val CR 169.50%). Examining the risk metrics, DDPG again stands out with the lowest maximum drawdown (MDD 19.20%), well below the buy-and-hold drawdown (MDD 29.93%), while DQL incurs a higher drawdown (MDD 35.76%) despite its competitive return. PG precisely matches the buy-and-hold drawdown (MDD 29.93%), reflecting a policy that largely tracks the market.

BTC presents a substantially harder generalisation challenge. The B&H baseline itself has a Sharpe ratio of -0.87 and a maximum drawdown of 49.72%, reflecting a severe bear market in the test period. All four models outperform buy-and-hold on cumulative return and Sharpe ratio. DDPG is the only model to achieve a positive return (CR 1.58%, SR 0.23, MDD 35.30%), significantly outperforming buy-and-hold (CR -34.27%, SR -0.87, MDD 49.72%). PG (CR -14.53%, SR -0.33, MDD 29.98%) is the second-best, markedly reducing both losses and drawdown relative to the baseline. DQL ranks third (CR -23.22%, SR -0.55, MDD 37.41%), while PPO (CR -33.80%, SR -0.85, MDD 59.18%) ranks last, still outperforming buy-and-hold in terms of returns but incurring a substantially larger maximum drawdown, effectively replicating the market trajectory without the capital preservation benefit seen in the other models. DQL records the highest validation SR on BTC (2.11) yet finishes with a test SR of -0.55, illustrating how strong bull market validation performance can fail to generalise to a bear-market test regime.

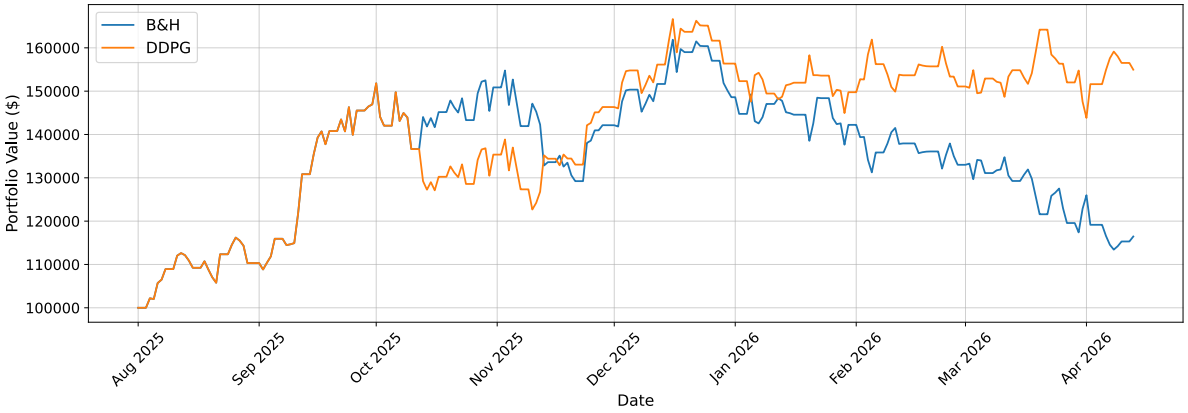
6.1. Algorithm Analysis

DDPG is the standout algorithm, ranking first on both assets in return and Sharpe ratio. Its continuous-action actor-critic architecture, combined with a replay buffer and soft target updates appears to provide a more stable optimisation landscape under the alpha reward signal than recurrent on-policy methods (PG and PPO) or the purely off-policy DQL. On TSLA, DQL matches DDPG closely in return (52.62% vs 54.96%) but with a substantially larger drawdown (35.76% vs. 19.20%), suggesting riskier positioning from DQL. On BTC, no recurrent on-policy method (PG, PPO) achieves a positive return, and PPO nearly replicates the buy-and-hold trajectory with a higher drawdown.

Figures 3a and 3b show the DDPG portfolio trajectory against the buy-and-hold on the test set for each asset.

Figure 3: DDPG vs B&H portfolio values on the TSLA and BTC test sets.

(a) DDPG vs B&H portfolio values on the TSLA test set.



(b) DDPG vs B&H portfolio values on the BTC test set.

On TSLA (Figure 3a), DDPG leads buy-and-hold throughout most of the test period. Both series rise together from August to December 2025, peaking over $\$160,000$ for buy-and-hold. After that peak, buy-and-hold retraces sharply as TSLA gives back gains, while DDPG preserves capital by reducing its long exposure, ending the period near $\$155,000$ versus $\$116,000$ for buy-and-hold. This sustained outperformance reflects DDPG's ability to time partial exits during the downswing, consistent with its best maximum drawdown of 19.20%.

On BTC (Figure 3b), the two series track each other closely during the flat August–October 2025 phase, indicating that DDPG holds near-neutral positioning while the market drifts. From November 2025 onward, DDPG diverges sharply upward, reaching around $\$130,000$ by December, while buy-and-hold begins a sustained decline that eventually drops below $\$80,000$ by early 2026. The agent sustains a

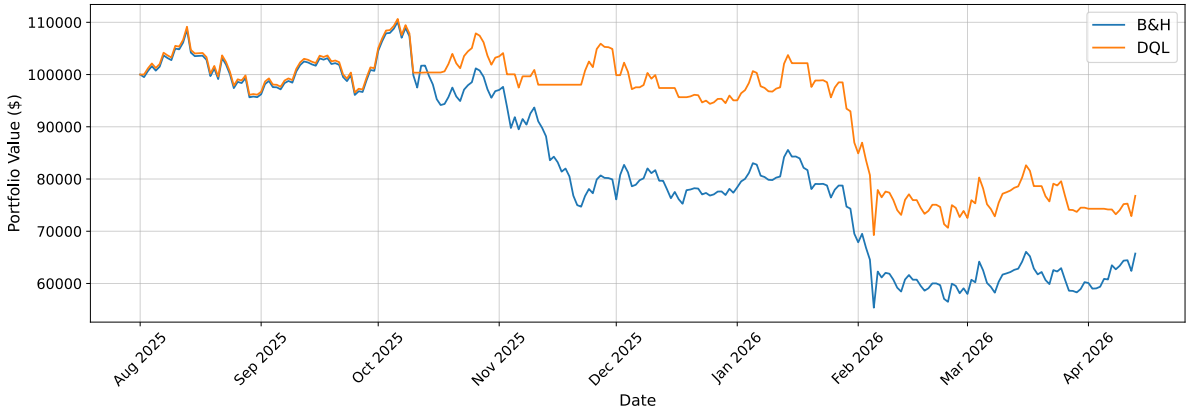
sharp drawdown in February 2026, pulling back to just over $\$90,000$ before stabilizing, but ultimately ends the period close to breakeven ($\$101,580$ vs $\$65,730$ for buy-and-hold). The contrast illustrates how DDPG's continuous actor effectively shifts from long to flat or short during sustained bearish periods, limiting participation in the worst of the drawdown.

DQL achieved the highest validation Sharpe ratio on both assets (TSLA: 3.29, BTC: 2.11) and was therefore selected as our submission model for the live CLEF competition rankings. We retain this choice despite DDPG's stronger offline backtest: our selection criterion was fixed a priori, and re-selecting the endpoint model based on a single realized test path would constitute selection bias. The gap between DQL's validation ranking and its backtest performance is instead reported as evidence of the regime-shift sensitivity analyzed in Section 6.2. Figures 4a and 4b show its test trajectories.

Figure 4: DQL vs B&H portfolio values on the TSLA and BTC test sets.

(a) DQL vs B&H portfolio values on the TSLA test set.

(b) DQL vs B&H portfolio values on the BTC test set.



On TSLA (Figure 4a), DQL takes an initially incorrect short position while TSLA rises, falling to roughly $\$75,000$ by November 2025 and accounting for the agent's high maximum drawdown (35.76%). It then reverses to a long position and steadily recovers, closing at around $\$152,000$, above buy and hold's $\$116,000$, for an overall gain of 52.62%.

On BTC (Figure 4b), DQL provides modest protection relative to buy-and-hold, staying closer to breakeven while B&H steadily declines. Both suffer the February 2026 sell-off, after which DQL stabilizes around $\$75,000$ versus $\$65,000$ for buy-and-hold (CR -23.22% vs -34.27%). The contrast between DQL's strong validation Sharpe ratio and its comparatively volatile test path highlights the regime-dependence of model selection discussed in Section 6.2.

A persistent training artefact is the massive gap between training-episode returns and test returns: PPO reports training-period portfolio gains of $114{,}462\%$ on TSLA and $256{,}989\%$ on BTC. These

inflated figures bear no direct relationship to the test performance and underscore the importance of Sharpe-based model selection over return-based selection.

6.2. Validation Sharpe Ratio vs. Test Performance

The relationship between validation Sharpe ratio and test performance is asset-dependent. As shown in Table 4, on TSLA, the two highest validation SR models (DQL: 3.29, DDPG: 2.03) are also the two best performers (52.62% and 54.96% respectively), suggesting that on a moderately trending asset, validation Sharpe ratio provides a useful signal. On BTC, the pattern breaks: DQL leads on validation SR (2.11) but ranks third on the test set (CR −23.22%, SR −0.55). DDPG, with a lower validation SR (1.87), is the only model to achieve a positive test return. This divergence arises because the BTC validation period (January 2023–August 2024) was a sustained bull market recovery, while the test period is a bear market; policies selected for strong upward momentum performance may not generalise to downturns. Overall, validation-based model selection eliminates the weakest checkpoints but remains subject to distributional shift between market regimes.

A structural contributor to this gap is the use of a single temporal validation window that happened to coincide with a bull market for both assets. Walk-forward cross-validation, which would create multiple rolling train/validate splits that collectively span bull runs, bear markets, and sideways regimes, would yield a more regime-balanced model selection criterion. Implementing this technique would likely have reduced the observed validation-to-test gap. Ultimately, walk-forward cross-validation was not implemented due to time constraints as this method is very computationally demanding.

7. Conclusion and Further Work

We presented a DRL-based trading system for CLEF 2026 FinMMEval Task 3 that integrates LLM-derived news sentiment with technical indicators and calendar features in a custom trading environment. Our key contributions are the alpha reward formulation applied during training, paired with Sharpe-based model selection on the validation set, and random episode start, which together reduced overfitting to the training sequence and aligned the optimisation objective with the competitive criterion of beating buy-and-hold. DDPG was the most robust algorithm across both assets, achieving a 54.96% cumulative return on TSLA (SR 1.44, MDD 19.20%) and 1.58% on BTC (SR 0.23, MDD 35.30%). However, our live endpoint uses DQL, selected on validation Sharpe ratio without reference to the test-period data. On TSLA, both DDPG (CR 54.96%, SR 1.44) and DQL (CR 52.62%, SR 1.38) significantly outperformed the baseline, while PG and PPO fell short. On BTC, DDPG was the only model to achieve a positive return while PG substantially reduced losses relative to buy-and-hold. The relationship between validation SR and test performance is asset-dependent: on TSLA, the top two validation SR models were also the top two test performers; on BTC, regime shift from a bull validation period to a bear test period caused DQL (highest validation SR at 2.11) to underperform DDPG (lower validation SR at 1.87), exposing the limits of validation-based model selection under distributional shifts.

Several directions remain for future work. First, replacing LLaMA 3.2 1B with the domain-specialised FinBERT [14] or a larger LLM may improve sentiment quality. Second, incorporating SEC 10-K/10-Q filings available in the task's daily context could provide longer-horizon signals. Third, walk-forward cross-validation over multiple regime-diverse validation windows would yield a more robust model selection criterion, reducing the bull market bias embedded in our single 2023–2024 validation split. Fourth, hierarchical or multi-timescale RL, learning both short-term momentum policies and longer-term regime-detection strategies, may help bridge the validation-to-test generalisation gap. Fifth, ensemble methods that combine the action outputs of multiple agents could smooth individual model noise and reduce the variance observed across algorithm-asset pairs. Finally, a systematic ablation study, selectively removing individual feature groups (technical indicators, sentiment scores, and calendar encodings) or replacing the LLM sentiment score with a simpler lexicon-based baseline, would quantify each component's marginal contribution to agent performance and clarify whether the observed gains are primarily driven by price-derived signals or by the news sentiment score.

Acknowledgments

This research was supported by funding from the Natural Sciences and Engineering Research Council of Canada (NSERC). We also gratefully acknowledge the Digital Research Alliance of Canada for providing the high-performance computing resources utilized throughout this project.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (2015) 484–489.

[2] Z. Xie, Y. Dai, R. Elbadry, V. Jani, X. Peng, L. Qian, G. Georgiev, D. Dimitrov, F. Zhang, J. Huang, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, S. Liu, P. Nakov, Overview of FinMMEval 2026: Multilingual and multimodal financial evaluation, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Springer Lecture Notes in Computer Science LNCS, Jena, Germany, 2026.

[3] Z. Xie, L. Qian, G. Georgiev, D. Dimitrov, R. Elbadry, F. Zhang, X. Peng, J. Huang, V. Jani, Y. Dai, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, S. Liu, P. Nakov, Overview of the FinMMEval 2026 task 3: Financial decision making, in: CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026.

[4] L. Qian, X. Peng, Y. Wang, V. J. Zhang, H. He, H. Smith, Y. Han, Y. He, H. Li, Y. Cao, Y. Yu, A. Lopez-Lira, P. Lu, J.-Y. Nie, G. Xiong, J. Huang, S. Ananiadou, When agents trade: Live multi-market trading benchmark for LLM agents, 2025. arXiv:2510.11695.

[5] J. Moody, L. Wu, Y. Liao, M. Saffell, Performance functions and reinforcement learning for trading systems and portfolios, Journal of Forecasting 17 (1998) 441–470.

[6] Z. Jiang, D. Xu, J. Liang, A deep reinforcement learning framework for the financial portfolio management problem, 2017. arXiv:1706.10059.

[7] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: Proceedings of the 33rd International Conference on Machine Learning (ICML), volume 48 of *Proceedings of Machine Learning Research*, PMLR, New York, NY, USA, 2016, pp. 1928–1937.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, 2017. arXiv:1707.06347.

[9] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, in: Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2016.

[10] X.-Y. Liu, Z. Xiong, S. Zhong, H. Yang, A. Walid, Practical deep reinforcement learning approach for stock trading, in: Proceedings of the NeurIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services, Montréal, Canada, 2018.

[11] H. Yang, X. Liu, S. Zhong, A. Walid, Deep reinforcement learning for automated stock trading: An ensemble strategy, in: Proceedings of the ACM International Conference on AI in Finance (ICAIF), New York, NY, USA, 2020.

[12] A. Ye, J. Xu, V. Veedgav, Y. Wang, Y. Yu, D. Yan, R. Chen, V. Chaudhary, S. Xu, Learning the market: Sentiment-based ensemble trading agents, 2024. arXiv:2402.01441.

[13] S. Du, H. Shen, A stock prediction method based on deep reinforcement learning and sentiment analysis, Applied Sciences 14 (2024). URL: https://www.mdpi.com/2076-3417/14/19/8747. doi:10.3390/app14198747.

[14] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, 2019. arXiv:1908.10063.

[15] X. Yu, Z. Chen, Y. Lu, Harnessing LLMs for temporal data – a study on explainable financial time series forecasting, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP): Industry Track, Singapore, 2023, pp. 739–753.
[16] A. Lopez-Lira, Y. Tang, Can ChatGPT forecast stock price movements? Return predictability and large language models, 2025. arXiv:2304.07619.
[17] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, second ed., The MIT Press, 2018.
[18] B. Ferrell, financial-news-multisource (revision b509ef6), 2025. doi:10.57967/hf/6432.
[19] K. R. French, Stock returns and the weekend effect, Journal of Financial Economics 8 (1980) 55–69.
[20] J. J. Murphy, Technical Analysis of the Financial Markets, New York Institute of Finance, 1999.
[21] W. Brock, J. Lakonishok, B. LeBaron, Simple technical trading rules and the stochastic properties of stock returns, Journal of Finance 47 (1992) 1731–1764.
[22] J. W. Wilder, New Concepts in Technical Trading Systems, Trend Research, 1978.
[23] G. Appel, Technical Analysis: Power Tools for Active Investors, Financial Times Prentice Hall, 2005.
[24] J. Bollinger, Bollinger on Bollinger Bands, McGraw-Hill, 2001.
[25] A. W. Lo, H. Mamaysky, J. Wang, Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation, Journal of Finance 55 (2000) 1705–1765.
[26] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, et al., The LLaMA 3 herd of models (2024). arXiv:2407.21783.
[27] A. Y. Ng, D. Harada, S. J. Russell, Policy invariance under reward transformations: Theory and application to reward shaping, in: Proceedings of the Sixteenth International Conference on Machine Learning (ICML), Morgan Kaufmann, 1999, pp. 278–287.
[28] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (1992) 229–256.
[29] H. van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), Association for the Advancement of Artificial Intelligence, Phoenix, Arizona, 2016, p. 2094–2100.