# Large Language Model-guided Reinforcement Learning for Algorithmic Trading

Michał Andrzejewski

January 2025

## 1 Introduction

In the past few years, the field of Natural Language Processing (NLP) has turned the world upside down, driven largely by advances in Large Language Models (LLMs) built on the Transformer architecture. Models such as OpenAI's GPT series, Meta's LLaMA, and Google's BERT and their derivatives have redefined machine learning (ML), enabling systems to understand and generate human-like text with remarkable accuracy. These breakthroughs have not only enhanced traditional NLP tasks like translation, but have also altered what we perceived to be possible in the landscape of artificial intelligence (AI).

For reasons we will elaborate on here, the integration of LLMs with ML paradigms marks one of the most promising and novel frontiers in AI research. Reinforcement Learning (RL), which excels at training agents to make sequential decisions in dynamic environments, has by necessity relied on structured numerical data. The recent surge of LLMs offers a compelling opportunity to enrich RL by incorporating insights derived from unstructured textual data. This hybrid approach would leverage the interpretive power of LLMs to process vast amounts of qualitative information, such as financial news, social media sentiment, or corporate reports, which can then guide RL algorithms in making more informed and adaptive decisions.

In this essay, we will explore the potential of LLM-guided RL for algorithmic trading, demystifying its technical underpinnings, current research efforts, and the concerns that must be addressed. Given the nascent state of this hybrid approach, with only a handful of pioneering studies available, we aim to critically assess its feasibility and potential impact. Algorithmic trading, as one of the most data-intensive and dynamically evolving applications, provides a compelling testbed for this emerging bridging of ML paradigms.

## 2 Background Overview

In order to tackle the recent advances in the interconnectedness between Large Language Models and Reinforcement Learning, it is essential first to define and analyse their independent working mechanisms. By understanding the fundamental principles of these frameworks separately, we can better discern both the challenges and opportunities that arise when combining them, particularly in the context of dynamic and complex environments such as financial markets.

### 2.1 Reinforcement Learning and Its Applications

Reinforcement Learning offers a robust framework for training agents to learn optimal policies through iterative interactions with an environment. By focusing on maximizing cumulative rewards, RL is inherently suited to solving sequential decision-making problems, which is why its adaptability has enabled significant advancements across domains such as robotics, healthcare, and finance.

In financial markets, RL has facilitated the development of algorithmic strategies capable of adapting to volatile and unpredictable conditions demonstrating its effectiveness in portfolio optimization, trade execution, and risk management. Despite its capabilities, traditional RL methods often face scalability challenges and inefficiencies when applied to high-dimensional or unstructured data environments, limiting their real-world applicability in complex financial scenarios.

## 2.2 Foundations of Reinforcement Learning

At its core, Reinforcement Learning is grounded in the framework of a Markov Decision Process (MDP), as formalized by Bellman's foundational work on dynamic programming [3]. In general, a MDP is defined by the following components:

- $S$: State space
- $A$: Action space
- $P(s'|s,a)$: Transition probability function
- $R(s,a)$: Reward function
- $\gamma$: Discount factor, where $\gamma \in [0,1]$

The primary objective of RL is to derive an optimal policy $\pi^*$ that maximizes the expected cumulative reward:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]. \tag{1}$$

In practice, the definitions of these MDPs can vary depending on their objectives, as different use cases aim to incorporate various factors. For instance, as exemplified in the work by Hambly, Xu, and Yang (2022) in ”Recent Advances in Reinforcement Learning in Finance”[11], we can discern two main approaches used in financial markets, where one is the infinite time horizon with discounted rewards, typically used for portfolio optimization, while on the other hand, the finite time horizon with terminal reward is commonly applied in high-frequency trading.

On these MDPs we can apply RL algorithms, such as the widely used Q-Learning, that works by iteratively updating the Q-value function:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right], \tag{2}$$

where $\alpha$ represents the learning rate, $\gamma$ denotes the discount factor, $R(s,a)$ is the reward received after taking action $a$ in state $s$, and $s'$ is the next state resulting from action $a$.

This algorithm aims at approximating $Q^*$, the optimal Q-function, converging towards an optimal trading policy through repeated updates of the Q-value function. However, it has limitations in financial markets where the action space evolves every second [4], necessitating manual adjustments of MDP parameters and limiting full autonomy in real-time trading.

## 2.3 Deep Reinforcement Learning

The limitations of standard Reinforcement Learning, particularly in handling high-dimensional state-action spaces and multifaceted environments, have led to the emergence of Deep Reinforcement Learning (DRL) [7]. By incorporating deep neural networks with RL to approximate complex value functions or policies, DRL has managed to become increasingly relevant in financial applications, where dynamic, high-dimensional decision-making is crucial.

While these algorithms are still quite new as well, they have quickly gained traction across multiple domains, particularly in finance. As shown in Table 1, three principal paradigmatic algorithms define the foundation of DRL: DQN, PPO, and SAC; each of these algorithms possessing unique features that address specific types of tasks, making them highly adaptable and versatile.

These foundational algorithms not only highlight their individual strengths but also serve as the building blocks for more advanced, purpose-driven algorithms. Consequently, each is tailored to excel in different categories; for example, DRL has proven highly effective in portfolio optimization, where dynamic asset allocation strategies are employed to maximize returns while minimizing risk in volatile market conditions. In algorithmic trading, DRL is used to automate trade execution, optimize order placement, and reduce transaction costs, enabling seamless adaptation to high-frequency trading environments. Similarly, it enhances liquidity provision and inventory management in market-making scenarios, where managing market uncertainty is critical. Additionally, DRL offers robust solutions in option pricing by providing real-time, data-driven valuations of derivatives that can adapt to ever-changing market dynamics.

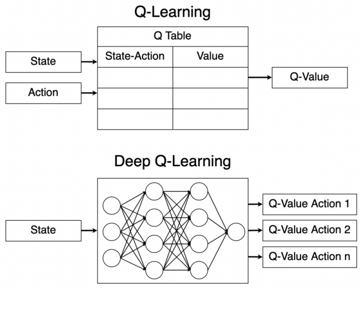


Figure 1: Comparison of Q-learning vs Deep Q-learning [15]

DRL holds significant promise, nevertheless its application in finance is fraught with challenges: financial markets are inherently volatile, making it onerous to strike a balance between exploration and exploitation strategies. The data used in there is often noisy, sparse, and non-stationary, complicating

Table 1: Prominent DRL Algorithms and Their Features

| **Algorithm** | **Key Features** | **Applications in Finance** |
|---|---|---|
| **Deep Q-Network (DQN)** | Utilizes neural networks to approximate $Q(s, a)$; employs experience replay and target networks for stabilization [7] | Portfolio optimization, where discrete actions (e.g., buy/sell/hold) are common |
| **Proximal Policy Optimization (PPO)** | Policy-gradient method that optimizes a clipped surrogate objective for stable and efficient learning [16] | High-frequency trading, where continuous adjustments are needed in real-time |
| **Soft Actor-Critic (SAC)** | Balances expected return and entropy to encourage exploratory and robust behavior in uncertain environments [6] | Market making, where managing uncertainty and inventory is critical |

both the training and evaluation processes. Furthermore, crafting reward functions that accurately align with aggregated financial objectives is a formidable task. DRL models also face the persistent risk of overfitting to historical data, resulting in poor generalization when deployed in live markets.

## 2.4 Large Language Models and Their Capabilities

The hurdles of DRL highlight the need for advanced methodologies that can effectively address the labyrinthine nature of financial data and other high-stakes decision-making contexts. Might integrating state-of-the-art NLP models, especially now with modern LLMs, be the way to go?

The Transformer architecture adopted by LLMs allows them to capture intricate contextual relationships and long-range dependencies in textual data, enabling tasks like sentiment analysis, summarization, and trend detection with unprecedented accuracy. Unlike earlier architectures such as recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), Transformers process sequences in parallel rather than sequentially, which not only accelerates computation but also improves the ability to model intricate dependencies in lengthier sequences.

At the core of LLMs lies the Transformer, a model said to have revolutionised NLP with its self-attention mechanism, a method that dynamically evaluates the relevance of each input token to every other token in a sequence. This approach, which we will try to detail the mathematical foundations of, ensures that even long-range dependencies within textual data are effectively captured, eliminating the limitations of sequential processing.

First, the Scaled Dot-Product Attention is calculated resulting in normalized attention scores that determine how much ”focus” the model places on each token relative to others:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

Here: - $Q$ (Queries), $K$ (Keys), and $V$ (Values) are learned projections of the input embeddings. - $d_k$ is the dimensionality of the keys, scaling the dot product to prevent large values.

Next, the Multi-Head Attention extends single-head attention by projecting $Q$, $K$, and $V$ into multiple subspaces:

$$\text{MH}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{4}$$

Where each head in 4 captures different types of relationships, enabling the model to understand the data from various perspectives, and is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

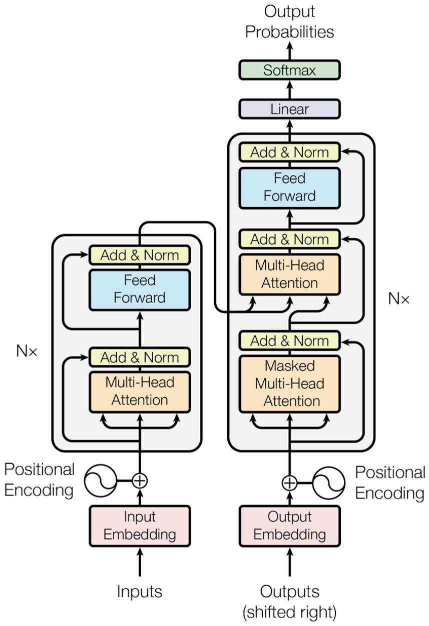


Figure 2: The Transformer - model architecture [12].

The learned weight matrices $W_i^Q$, $W_i^K$, and $W_i^V$ allow each head to specialize in capturing different patterns, such as syntactic dependencies or semantic relationships.

After attention has been computed, the model applies a Feed-Forward Neural Network (FFNN) to each token's representation. This step adds non-linear transformations, enriching the information extracted from the attention mechanism. It is computed as:

$$\text{FFNN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{6}$$

The first layer introduces non-linearity (using the ReLU activation function), while the second refines the representation. This combination allows the model to learn more complex patterns.

The results come as a set of output probabilities for different possible interpretations of the input. This architecture plays a great role in the complete workflow of the operations of an LLM that can be broken down into the following stages, as depicted in Figure 3:

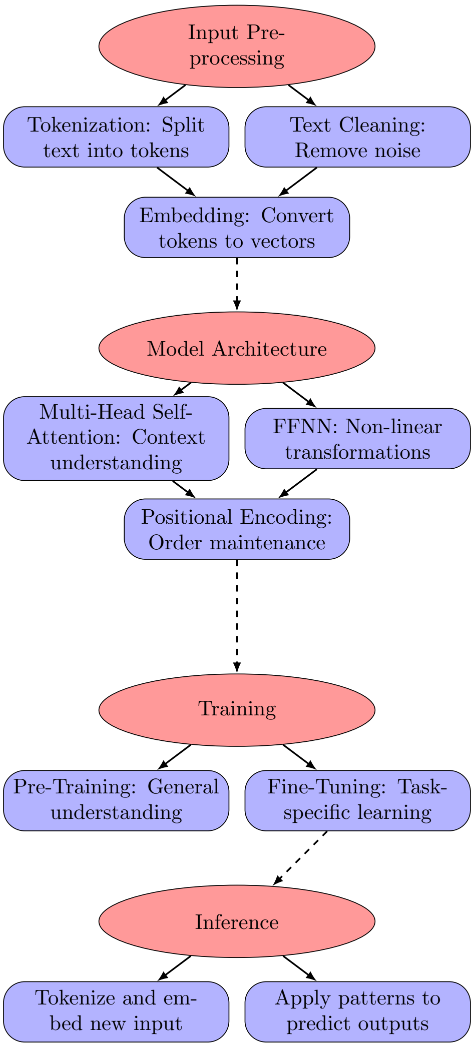

Input Pre-processing
Tokenization: Split text into tokens
Text Cleaning: Remove noise
Embedding: Convert tokens to vectors
Model Architecture
Multi-Head Self-Attention: Context understanding
FFNN: Non-linear transformations
Positional Encoding: Order maintenance
Training
Pre-Training: General understanding
Fine-Tuning: Task-specific learning
Inference
Tokenize and embed new input
Apply patterns to predict outputs

Figure 3: Workflow of a LLM based on Transformer architecture.

**1. Input Preprocessing**: The workflow of LLM begins with input preprocessing, an essential step that ensures the raw text is ready for computational processing. Text is divided and broken down into smaller units called tokens, which might represent entire words, parts of words, or even individual characters. This tokenization process allows the model to work with manageable pieces of information while retaining meaningful patterns within the text. Noise, such as irrelevant symbols or extraneous elements, is also removed during this step to create a cleaner input. Each token is then converted into a high-dimensional numerical vector, or embedding, designed to capture semantic relationships. This transformation ensures that tokens with similar meanings are positioned close to one another in the embedding space, forming a foundation for understanding linguistic context.

**2. Model Architecture**: Once the input is prepared, the model architecture takes over. This is where the Transformer model comes into play. The Multi-Head Self-Attention mechanism analyzes the relationships between all tokens in the input. By assigning attention scores, the model dynamically determines how much influence each token should have on every other token. This capability allows the model to grasp context not only within local sequences but also across long distances in the text, addressing limitations of earlier sequential models. Complementing this is the FFNN, which applies non-linear transformations to refine the contextual representations derived from attention. Positional Encoding is another pivotal component, adding information about the order of tokens. Since Transformers lack an inherent sense of sequence, these encodings ensure that the model understands the relative position of words, enabling it to handle structured text effectively.

**3. Training**: Models are trained on massive text corpora to learn general language understanding. Models are then refined on task-specific datasets to specialize for particular applications, such as sentiment analysis or summarization. During training, optimization algorithms minimize a loss function to refine the model's parameters to its applications, while regularization prevents overfitting.

**4. Inference**: New input text is tokenized and embedded. The model leverages its trained weights to generate predictions or outputs, such as classification labels or text completions.

By integrating these stages, LLMs achieve their ability to understand, analyze, and generate human-like text, delivering predictions or completions with remarkable accuracy and contextual depth. Thanks to the flexibility of the training phase, they can be tailored to suit different situations without losing their aptitude.

# 3 Large Language Model Integration With Reinforcement Learning Algorithms

## 3.1 Motivation and Developpment of LLM-Guided Reinforcement Learning

As we have seen, Transformers' ability to process large amounts of unstructured textual data and extract nuanced contextual relationships makes them particularly suited to address deficiencies in DRL. Since LLMs excel at interpreting qualitative data, understanding ambiguous contexts, and generating human-like textual predictions, while simultaneously RL algoritms perform the best at sequential decision-making neither framework alone can fully address the challenges of dynamic, unstructured environments like financial markets. Their integration seems to represent a natural evolution, leveraging the strengths of each to overcome their respective limitations.

The convergence of LLMs and RL, LLM guided RL (LGRL), aims to achieve two primary objectives that significantly enhance decision-making capabilities in complex environments like financial markets:

The first is incorporating updated qualitative data into the RL agent's MDP to simulate their environment. While RL algorithms traditionally rely on structured, quantitative data, much of the most valuable information can be found in qualitative sources. This could include news articles, social media posts, or expert opinions - all of which carry critical insights but have not been interpretable by normal RL algorithms. This is where LLMs come into play. They process this qualitative data, extracting meaning and translating it into a form that is actionable for RL agents.

For example, financial news articles or sentiment derived from social media posts from platforms like Twitter or Reddit can provide a deep understanding of market sentiment, or even predictive signals about market movements. In the beginning of 2021, the importance of analyzing such insights became evident, with the irrational buildup of stock price associated with a subreddit's, called WallStreetBets, speculation on GameStop's shares that led to a bubble that overestimated GameStop's real valuation by a margin of more than 20 times. Starting at $19.03 on January 4, 2021, the price reached $43.03 by January 21, a rise of over 100% without new company information. In the next five days, the price surged tenfold to peak at $483 on January 28 [5], showcasing the significant role of emotions displayed on social media and the potential opportunities this brings. LLMs can assess this unstructured data and translate it into quantitative values, such as a sentiment score or a trend prediction. These insights can then be fed into the RL agent's reward structure and used to guide policy updates, allowing the agent to adjust its trading strategies dynamically, responding to evolving market conditions in real time. Research highlights, such as those by the QuantAgent Research Group [14] and, independently, Ananya Unnikrishnan [13], emphasize how integrating LLMs with RL can managing portfolios. Their work demonstrates the capability of RL systems to process, interpret, and incorporate diverse, unstructured data—such as news, social media sentiment, and financial reports—directly into trading strategies.

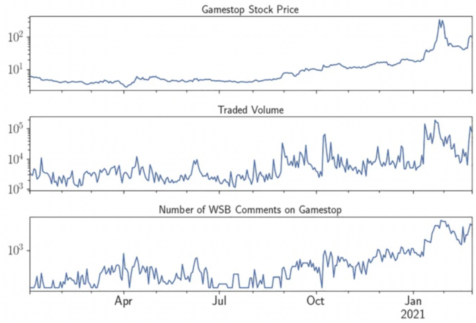
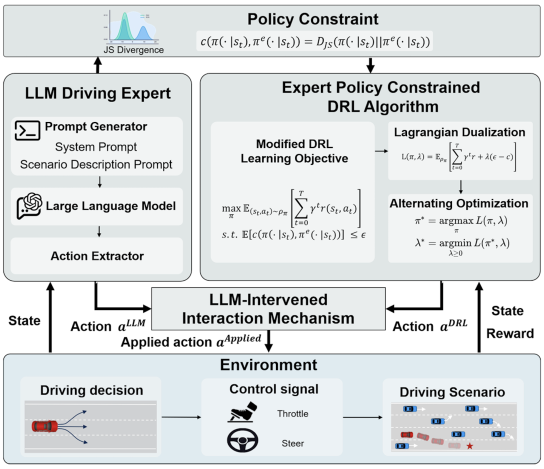


Figure 4: Logarithmic Chart of GameStop's Stock Price, Traded Volume and Reddit WSB Comments[5].



Policy Constraint
JS Divergence
$c(\pi(\cdot|s_t),\pi^e(\cdot|s_t)) = D_{JS}(\pi(\cdot|s_t)||\pi^e(\cdot|s_t))$
LLM Driving Expert
Prompt Generator
System Prompt
Scenario Description Prompt
Large Language Model
Action Extractor
Expert Policy Constrained DRL Algorithm
Modified DRL Learning Objective
$\max_{\pi} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{T}\gamma^t r(s_t,a_t)\right]$
$s.t.\ \mathbb{E}[c(\pi(\cdot|s_t),\pi^e(\cdot|s_t))] \le \epsilon$
Lagrangian Dualization
$L(\pi,\lambda) = \mathbb{E}_{\rho_\pi}\left[\sum_{t=0}^{T}\gamma^t r + \lambda(\epsilon - c)\right]$
Alternating Optimization
$\pi^* = \operatorname{argmax}_{\pi} L(\pi,\lambda)$
$\lambda^* = \operatorname{argmin}_{\lambda\ge 0} L(\pi^*,\lambda)$
State
Action $a^{LLM}$
LLM-Intervened Interaction Mechanism
Applied action $a^{Applied}$
Action $a^{DRL}$
State
Reward
Environment
Driving decision
Control signal
Throttle
Steer
Driving Scenario

Figure 5: LLM-guided DRL System[8].

Secondly, the other objective is to supervise the learning process of RL agents while minimizing the workload required from humans during the training phase [9], thereby reducing labor costs. DRL usually demands extensive computational resources to achieve a high-quality policy in complex scenarios, largely because of its quite low learning efficiency. Moreover, relying on human experts to guide the learning process incurs prohibitively high labor costs, limiting its practical applications. Recent research, such as the LGDRL framework proposed for autonomous vehicles [8], offers a novel approach to address this. Its' framework, illustrated in Figure 5, integrates an LLM-based driving expert into DRL systems to provide intelligent guidance during the learning process and intervening in agent actions, thereby enhancing efficiency, as well as and reducing dependency on human feedback.

The application of LLM-guided RL is relatively recent, with only a few pioneering studies demonstrating its feasibility. These advancements can be attributed to the technical progress made in recent years in the development and complexity of algorithms and models. Earlier attempts to integrate language understanding into RL systems faced significant problems, primarily due to the limitations of the models and techniques available at the time. As a result, these models were not adequately equipped to handle the intricate tasks required for such integrations. The field has since advanced considerably, enabling the discussion and realization of merging LLMs with RL.

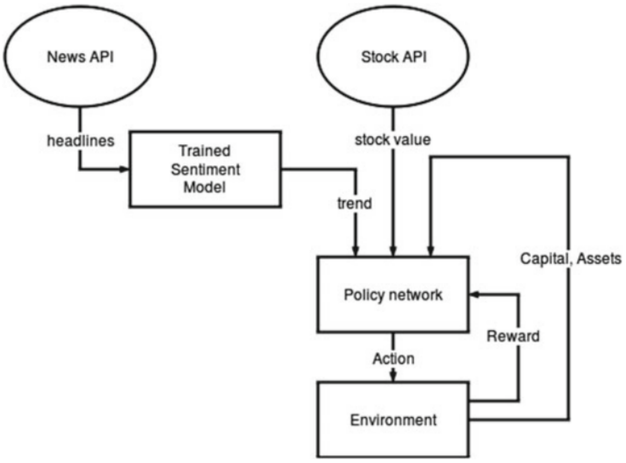


Figure 6: Early recurrent convolutional neural network (RCNN) for classification of news sentiment integration in DRL Workflow Framework. [1]

- **Limitations of Early Models**: Early attempts, such as those described by Azhikodan et al. [1], lacked both LLMs and advanced RL techniques like Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC). These methods, often based on Recurrent Convolutional Neural Networks (RCNNs), represented initial efforts to integrate sentiment analysis into trading strategies. However, they were constrained by their inability to process nuanced language patterns or dynamically adapt to changing market conditions. Figure 6 illustrates such an early workflow, which laid the groundwork for subsequent advancements leveraging LLMs.
- **Advancements with Modern Models**: Recent models have significantly improved upon these limitations by incorporating LLMs for contextual sentiment analysis and leveraging advanced computational capabilities. Unlike early RCNN-based models that relied on predefined feature extraction, modern LLM-guided RL systems dynamically process nuanced sentiment signals, enabling adaptive responses to evolving market conditions. For example, LLMs can distinguish between neutral and negative sentiment in ambiguous contexts, a capability that traditional models often lack.

## 3.2 Financial Applications: Methodologies, Challenges, and Concerns

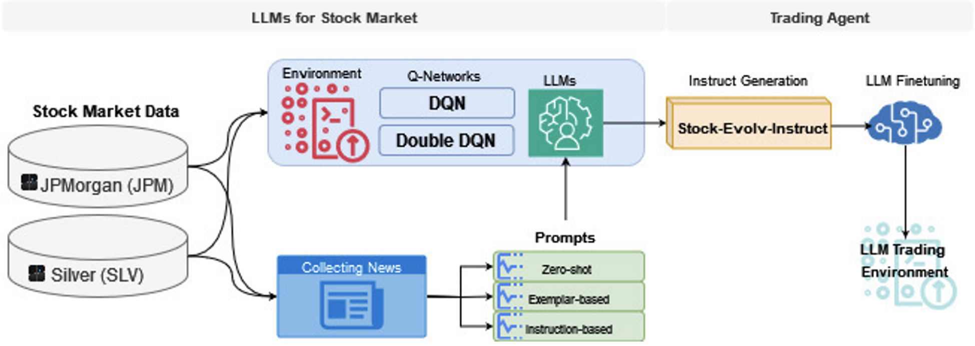


Figure 7: Proposed Framework for LGRL in Algorithmic Trading [18]

### 3.2.1 LGRL Agents in Algorithmic Trading

Adapting workflows from studies such as [8] and [13], the integration of LLMs with RL in financial applications introduces domain-specific refinements to address the unique challenges of volatile and data-rich environments. Below, we detail the tailored system workflow for financial applications based on the work [18] that introduces an algorithmic trading framework presented for reference in figure 7.

The workflow begins with the integration of both structured and unstructured data, each contributing unique insights to the decision-making process.

**Structured Data:** Key financial inputs include market prices, volatility indices, macroeconomic indicators, and historical performance trends. These data points undergo preprocessing steps such as normalization, scaling, and detrending to ensure compatibility with RL models.

**Unstructured Data:** Data is drawn from sources such as financial news articles, analyst reports, earnings call transcripts, and social media platforms. LLMs process this data to generate actionable insights. Outputs include sentiment scores, event impact probabilities, and risk sentiment metrics, which together provide a richer contextual understanding of market behavior. For instance, financial-specific LLMs like FinBERT can be used identifying nuanced market sentiment and assessing the impact of event-driven risks.

LLMs then play a key role in **feature engineering**, where they extract domain-specific signals. These include quantifying bullish or bearish sentiment and evaluating the potential market impact of geopolitical or economic events. By focusing on sentiment scoring and event impact evaluation, these models prioritize the most relevant signals for financial decision-making.

**Temporal modeling** further enhances the system's adaptability, by analyzing time-series data using techniques such as Fourier transforms and attention-based encodings to capture sequential dependencies. Temporal embeddings are then employed to improve the RL agent's responsiveness to market volatility, allowing it to adapt dynamically to evolving patterns.

The **reward structures** in these systems are designed to balance two key objectives: maximizing returns and minimizing risks. Insights from LLMs dynamically shape these rewards, such as penalizing strategies that overexpose portfolios to volatility or rewarding hedging actions in response to shifts in sentiment.

State-of-the-art RL algorithms like PPO and SAC are adapted to financial applications. These algorithms facilitate continuous-action spaces, enabling precise asset allocation and dynamic risk adjustments tailored to market conditions.

As we could observe, financial markets share similarities with other dynamic systems, we discussed before, such as autonomous driving environments, as highlighted in studies like [17, 9]. For example, financial markets, much like road traffic systems, exhibit unpredictable behaviors that require agents to adapt continuously.

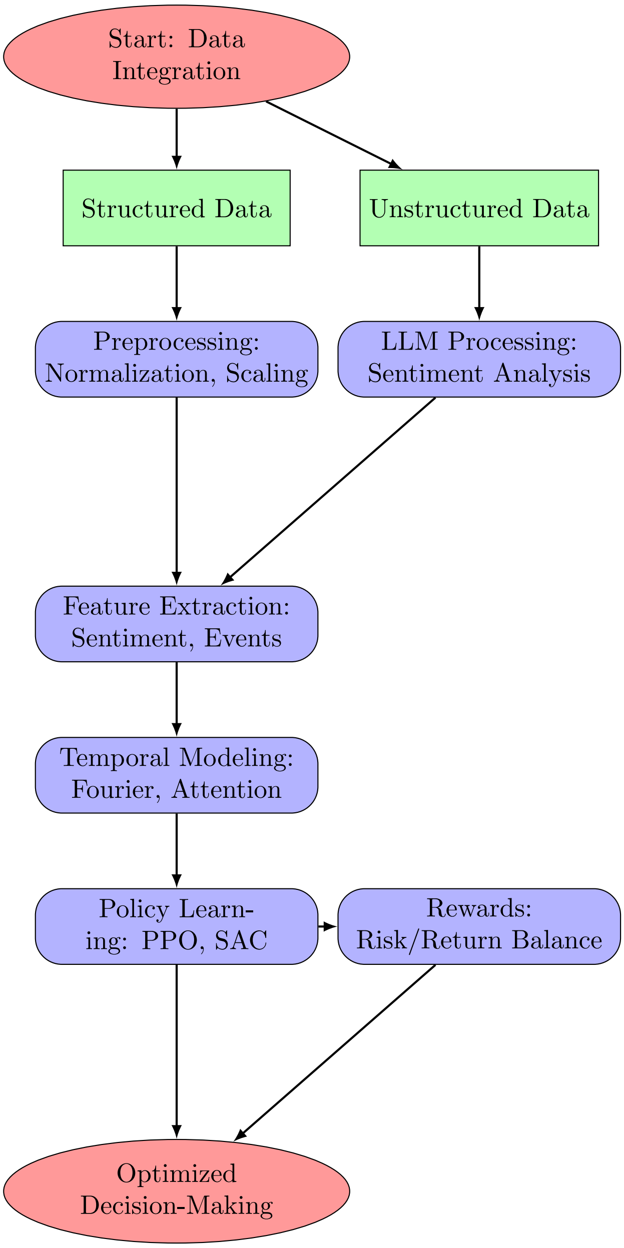

Start: Data Integration
Structured Data
Unstructured Data
Preprocessing: Normalization, Scaling
LLM Processing: Sentiment Analysis
Feature Extraction: Sentiment, Events
Temporal Modeling: Fourier, Attention
Policy Learning: PPO, SAC
Rewards: Risk/Return Balance
Optimized Decision-Making

Figure 8: Workflow for Data Integration and Decision-Making

In this context, **market simulation** serves as the environment, with RL agents leveraging LLM insights to navigate sudden market shocks. This is akin to how autonomous systems respond to unexpected hazards on the road, demonstrating the versatility and robustness of LGRL agents in handling real-world complexities.

### 3.2.2 Technological Bottlenecks and Scalability

More maintainable than human feedback, LLM-guided RL systems would still require substantial computational resources to process high-dimensional financial data in real time. As emphasized in [17], this scalability issue limits their deployment across diverse financial instruments and volatile markets.

These systems promise enhanced decision-making, however their opacity and dependence on data-driven patterns raise concerns about interpretability and reliability, reflecting the broader epistemological question of whether humans can control the tools they create, or if these tools might eventually begin to control them.

Algorithmic strategies, when widely adopted, have the potential to influence market dynamics in unpredictable ways. Flash crashes and other emergent phenomena highlight the systemic risks posed by collective algorithmic behavior [18].

This raises questions about the philosophical implications of designing systems that could destabilize the very markets they aim to optimize. While promising, the integration of LLMs with RL introduces several technical and operational challenges. Computational overhead is a significant concern, as LLMs are resource-intensive, and combining them with RL systems may exacerbate latency issues, particularly in high-frequency trading environments. Moreover, issues of generalization, as discussed in [9], remain critical, as LLMs trained on static datasets might struggle to adapt to rapidly shifting market dynamics.

It is very important to note that the research addressing this topic is very new. The few papers directly discussing the concept of LGRL are all from 2024 and have not yet been fully validated or widely published. For instance, [18] is currently under double review for an upcoming ICLR conference. Reviews on OpenReview reveal diverging opinions—some highly positive and others critical. This raises a debate: can we discuss this innovation confidently, or is it premature? Alternatively, perhaps this is the ideal moment to contribute to this emerging field, laying the groundwork for future breakthroughs.

### 3.2.3 Concerns in Integrating LLMs with RL

Ethical concerns also arise, especially in scenarios where LLMs and RL agents might inadvertently reinforce biases in financial markets. It's true that the "black-box" nature of AI systems often raises important concerns about transparency and accountability, and rightfully so. As mentioned in [2], these issues can complicate regulatory compliance and have significant societal impacts. One of the key challenges is the potential for AI systems, including LLM-guided RL systems, to amplify biases present in historical data, leading to discriminatory outcomes, sometimes such that we would not be able to identify before seeing the results.

The paper [19] underscores this problem by demonstrating how sentiment-based strategies can inadvertently reinforce gender biases in stock valuation predictions, which in turn attracts regulatory scrutiny. This example highlights the critical need for thorough evaluation and mitigation of biases when applying AI tools. To ensure responsible and ethical use of AI, it's essential to implement robust mechanisms for bias detection, transparency, and accountability. Continuous monitoring and evaluation, along with collaboration between AI developers, users, and regulators, can help mitigate the risks and harness the benefits of these advanced technologies.

## 4 Conclusion

As LLM-guided RL systems mature, they might hold the potential to completely change financial markets into adaptive, resilient ecosystems. Integrating technical rigor with ethical foresight ensures that these systems amplify human potential rather than undermining it.

The integration of LLM-guided RL into financial systems is more than a technical endeavor — it is a philosophical reflection on humanity's aspirations and limitations. By embracing both innovation and introspection, we can design systems that serve collective welfare while safeguarding against unintended consequences.

Adaptation to novel market scenarios remains a fundamental challenge on multiple dimensions. LLMs, while proficient at encoding historical data, often struggle with generalization in dynamic environments [8]. The tendency to overfit specific patterns jeopardizes robustness during rare events, such as financial crises, and could potentially bring catastrophic results.

From a philosophical standpoint, this fragility exposes the limitations of artificial systems when compared to human cognition, which is inherently more flexible and context-aware. This aligns with

Shoshana Zuboff's critique in [20], which emphasizes the ethical risks of opaque, algorithm-driven systems. Zuboff warns that without deliberate human-centric design, financial automation could prioritize efficiency and profit over transparency and accountability. Integrating human oversight ensures that these systems remain aligned with societal values, fostering trust and preventing potential misuse of technological capabilities.

Increasing interpretability of LLM-guided RL outputs is critical. Techniques such as explainable AI (XAI) can ensure that financial decisions remain transparent to stakeholders including, especially the public, who can be directly impacted by the outcomes of these systems' decisions. Accountability frameworks must trace the actions of autonomous agents, aligning with the ethical imperative of responsibility.

## References

[1] Azhikodan, Akhil Raj, Anvitha G. K. Bhat, and Mamatha V. Jadhav (2019). "Stock Trading Bot Using Deep Reinforcement Learning". Lecture Notes in Electrical Engineering, vol. 505. Springer. DOI: https://doi.org/10.1007/978-981-10-8201-6_5.

[2] Azzutti, Alessio et al. (2021). "Machine Learning, Market Manipulation and Collusion on Capital Markets: Why the 'Black Box' Matters". European Banking Institute Working Paper Series 2021, no. 84. University of Pennsylvania Journal of International Law, vol. 43, no. 1, SSRN. https://doi.org/10.2139/ssrn.3788872.

[3] Bellman, R. (1957). *The Theory of Dynamic Programming.* Princeton University Press.

[4] Bacoyannis, V. et al. (2018). "Idiosyncrasies and Challenges of Data-Driven Learning in Electronic Trading". arXiv preprint. https://arxiv.org/abs/1801.01234.

[5] A. Betzer, J.P. Harries, *How online discussion board activity affects stock trading: the case of GameStop*, Financ. Mark. Portf. Manag., vol. 36, pp. 443–472, 2022. https://doi.org/10.1007/s11408-022-00407-w

[6] Haarnoja, T., Zhou, A., Abbeel, P., Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv preprint arXiv:1801.01290.*

[7] Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. "Human-level control through deep reinforcement learning." *Nature* 518, no. 7540 (2015): 529-533. DOI: https://doi.org/10.1038/nature14236

[8] Hao Pang, Zhenpo Wang, Guoqiang Li. *Large Language Model guided Deep Reinforcement Learning for Decision Making in Autonomous Driving.* arXiv preprint arXiv:2412.18511v1, 2024. https://arxiv.org/abs/2202.03070.

[9] S. Zhang, S. Zheng, S. Ke, Z. Liu, W. Jin, J. Yuan, Y. Yang, H. Yang, Z. Wang, *How Can LLM Guide RL? A Value-Based Approach*, arXiv preprint arXiv:2402.16181, 2024. https://arxiv.org/abs/2402.16181

[10] Watkins, C. J. C. H., and Peter Dayan (1992). "Q-learning". Machine Learning, 8(3–4), 279–292. https://doi.org/10.1007/BF00992698.

[11] Hambly, B., Xu, R., Yang, H. (2022). "Recent Advances in Reinforcement Learning in Finance. February 3, 2022."

[12] Vaswani, A., Jones, L., Shazeer, N., Parmar, N., Gomez, A. N., Uszkoreit, J., Kaiser, Ł., & Polosukhin, I. (2023). Attention Is All You Need. *arXiv preprint arXiv:1706.03762v7.*

[13] Financial News-Driven LLM Reinforcement Learning for Portfolio Management, arXiv preprint arXiv:2411.11059, 2024. https://doi.org/10.48550/arXiv.2411.11059

[14] Wang, S., Yuan, H., Ni, L. M., and Guo, J. (2024). "QuantAgent: Seeking Holy Grail in Trading by Self-Improving Large Language Model." arXiv preprint arXiv:2402.03755v1 [cs.AI].

[15] Sebastianelli, A., Tipaldi, M., Ullo, S., and Glielmo, L. (2021). "A Deep Q-Learning Based Approach Applied to the Snake Game."

[16] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347.*

[17] Z. Liu, H. Guo, H. Hu, S. Ke, S. Zhang, B. Liu, Z. Wang, *Reason for Future, Act for Now: A Principled Framework for Autonomous LLM Agents with Provable Sample Efficiency*, arXiv preprint arXiv:2309.17382v3, 2024. `https://arxiv.org/abs/2309.17382`

[18] Authors (2024). "Advancing Algorithmic Trading with Large Language Models: A Reinforcement Learning Approach for Stock Market Optimization." ICLR 2025 Conference Submission. arXiv preprint arXiv:2402.03755v1 [cs.AI].

[19] Nakagawa, K., Hirano, M., and Fujimoto, Y. (2024). "Evaluating Company-specific Biases in Financial Sentiment Analysis using Large Language Models." arXiv preprint arXiv:2411.00420v1 [q-fin.CP].

[20] Zuboff, S. (2019). "The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power." Profile Books. ISBN 9781781256855.