FinAgents: Multi-Agent Financial Trading with Domain-Adapted Language Models

Vaibhav Dhanuka vdhanuka@wisc.edu **Shashwat Negi** negi3@wisc.edu **Zichen Liu** zliu2263@wisc.edu **Quanliang Liu** qliu388@wisc.edu

Abstract

Abstract—LLM-based multi-agent trading frameworks such as **TradingAgents** improve interpretability by decomposing decisions into specialized roles (analysis, debate, execution, and risk control), but their performance is sensitive to noisy and finance-misaligned sentiment signals. We enhance the **News/Social** components by introducing **FinLLaMA+**, a domain-adapted *Llama 3.1–8B* trained via **Domain-Adaptive Pretraining (DAPT)** on earnings call transcripts followed by **Supervised Fine-Tuning (SFT)** for financial sentiment classification. This adaptation reduces language-model perplexity by **22.52%** and improves sentiment performance to **0.93 macro-F1**. We integrate FinLLaMA+ into the agent pipeline using structured outputs (label, confidence, relevance) and confidence-weighted aggregation to provide auditable evidence for downstream debate and trading decisions.

Keywords—Multi-agent systems, financial trading, large language models, domain adaptation, sentiment analysis.

1 Introduction

Our project, **FinAgents**, extends the original **TradingAgents** framework by reconstructing its multi-agent decision pipeline and strengthening its sentiment analysis component. While TradingAgents provides a modular architecture for coordinating LLM-based analyst, researcher, trader, and risk-management roles, its sentiment subsystem relies on generic models that lack domain specialization. As a result, sentiment-driven reasoning can be inconsistent, insufficiently grounded in financial language, and sensitive to variations in news phrasing. Improving this weakest component is thus a natural and impactful direction for advancing the overall system.

Figure 1 summarizes the TradingAgents workflow that we faithfully reproduced as the foundation of our work. After rebuilding the architecture, we validated the full execution path—including data ingestion, agent communication, debate-driven decision synthesis, and downstream trading actions—and confirmed that the system operates as described in prior work. This reproduction step is essential, as it ensures that any improvements introduced by FinAgents can be attributed to model design rather than implementation discrepancies.

Compared with prior studies in financial LLMs, our work differs in focus and methodology. While existing literature has explored domain adaptation, instruction tuning, or multi-agent collaboration independently, FinAgents integrates these ideas into a controlled, end-to-end reproduction of TradingAgents with an upgraded, finance-specialized sentiment module. Unlike rule-based baselines or generic LLM sentiment filters, our approach adopts a domain-adapted variant of Llama 3.1–8B trained through a DAPT+SFT pipeline, enabling more consistent polarity estimation and better alignment with financial news semantics. This setting allows us to isolate and evaluate the impact of improved sentiment modeling within a broader agentic trading system.

In summary, the primary contributions of our project are: (1) a faithful reconstruction and verification of the TradingAgents framework, (2) an improved evaluation environment with re-implemented and tuned rule-based baselines, and (3) the development and integration of a domain-adapted sentiment module, **FinLLaMA+**, designed to enhance the News Analyst agent. Together, these steps establish **FinAgents** as a transparent and reproducible platform for assessing financial LLMs within multi-agent trading workflows.

[0] Artifacts: Code repository at https://github.com/quanliangliu/CS769-TradingAgents; Presentation at https://drive.google.com/file/d/1dILD4lkO8zcQNVJHsT1_-fPeb8XrjeDI/view?usp=sharing.

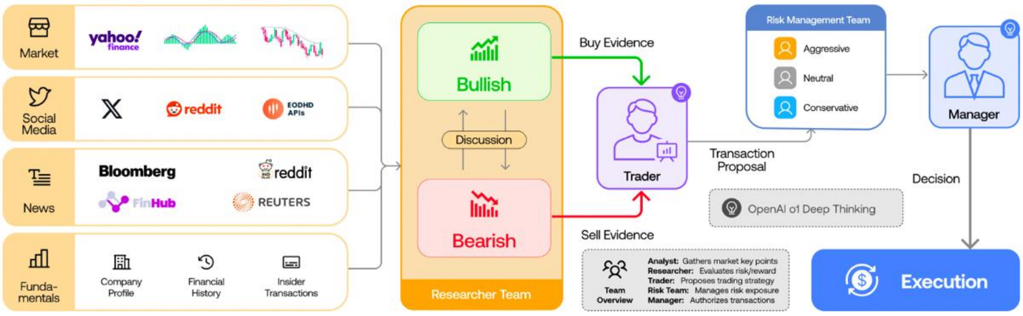

Figure 1: The architecture of the TradingAgents framework, adapted from Xiao *et al.* (2024).

2 Literature Review

Recent progress in financial language modeling and multi-agent trading systems reveals a consistent shift toward unifying three research directions: domain-adapted language understanding, calibrated sentiment modeling, and structured collaborative reasoning. These components increasingly form an integrated foundation for LLM-driven financial decision-making, motivating architectures such as **TradingAgents** (Xiao et al., 2025b) and our proposed **FinAgents** framework.

Early studies demonstrated that general-purpose LLMs struggle with specialized financial terminology, prompting the development of domain-adapted models. **FinBERT** (Araci, 2019) showed that pretraining on SEC EDGAR filings and fine-tuning on sentiment datasets substantially improves polarity detection. This paradigm was extended by **FinLLaMA** (Konstantinidis et al., 2024), which applies Domain-Adaptive Pretraining (DAPT) and parameter-efficient Supervised Fine-Tuning (SFT) via LoRA to yield strong performance on financial sentiment classification. Broader surveys such as (Jeong, 2024) consolidate these methods, emphasizing DAPT, Task-Adaptive Pretraining, and prompt-based adaptation as essential for handling regulatory constraints and temporal drift in financial corpora.

More recent work further refines the relationship between adaptation strategies, model scale, and downstream performance. Fine-tuning experiments on **Gemma-7B** (Mo et al., 2024) show that targeted SFT can significantly enhance macro-F1 and robustness on domain-specific datasets, even without massive training corpora. Complementary results from **LLM Adaptation for Finance** (Rodriguez Inserte et al., 2023) demonstrate that models under 1.5B parameters—when paired with curated financial text and synthetic instruction augmentation—can match or exceed the performance of larger LLMs. These findings inform our decision to adapt **Llama 3.1–8B** through a DAPT+SFT pipeline to achieve a practical balance of efficiency and financial domain fluency.

At the same time, recent analyses complicate assumptions about LLM reasoning in sentiment tasks. The study *Reasoning or Overthinking?* (Dimitris Vamvourellis, 2025) shows that Chain-of-Thought prompting often *reduces* accuracy on datasets such as the Financial PhraseBank, with concise, direct predictions outperforming long reasoning traces. This insight directly guides the design of our sentiment agent, whose goal is reliable polarity estimation rather than elaborate explanatory reasoning.

In parallel, multi-agent decision frameworks have emerged as a natural structure for integrating diverse financial signals. **TradingAgents** (Xiao et al., 2025b) organizes LLMs into specialized analyst and trader roles that debate and aggregate evidence, producing decisions that are both interpretable and empirically competitive. Extensions such as **ElliottAgents** (Wawer and Chudziak, 2025) and **Trading-R1** (Xiao et al., 2025a) further illustrate how classical technical analysis, retrieval augmentation, and reinforcement learning can enhance stability, transparency, and risk-adjusted returns.

Together, these works converge on a unified methodological direction: high-performing financial LLM systems require (1) domain-adapted text understanding, (2) calibrated and efficient sentiment modeling, and (3) structured multi-agent collaboration. Our **FinAgents** framework follows this trajectory by combining TradingAgents-style role decomposition with FinBERT- and FinLLaMA-inspired adaptation techniques. In particular, we strengthen the News Analyst agent through a

DAPT+SFT adaptation of Llama 3.1–8B, improving financial text comprehension while preserving compatibility with the original TradingAgents workflow.

3 Reimplementation of Related Work

To provide a controlled comparison for our FinLLaMA-enhanced system, we reimplemented key components of the **TradingAgents** framework (Xiao et al., 2025b) alongside a suite of traditional rule-based baselines within a unified backtesting pipeline. All methods operate under identical market conditions, so that any performance differences can be attributed to their underlying decision-making mechanisms rather than implementation details.

Our evaluation focuses on a single equity over a chosen date range, with all strategies acting in the same daily decision space by issuing **LONG**, **HOLD**, or **SHORT** signals. We collect and normalize daily OHLCV data (Open, High, Low, Close, Volume) into a standard single-ticker format. Signals are generated once per trading day and executed under a shared return-accounting protocol that explicitly prevents look-ahead bias.

The baseline set includes a Buy-and-Hold strategy, an SMA(5/15) moving-average crossover, a contrarian MACD strategy, a KDJ+RSI contrarian blend, and a Z-score mean-reversion strategy. Each baseline is expressed in a shared action language, making their positions directly comparable over time.

On the learning-based side, the **TradingAgents** system is instantiated as a graph of analyst agents that synthesize market data, technical indicators, fundamentals, and news. When enabled, these information sources include OpenAI-powered global news summarization for macro context and Alpha Vantage ticker-level news for firm-specific events. The agents' reasoning is recorded as human-readable decision logs, providing transparency into the decision-making process.

Performance across all methods is summarized using cumulative returns over the evaluation horizon. As an illustrative example, Figure 2 reports the cumulative returns for AMZN from January to April 2024 under both the rule-based methods, the vanilla TradingAgents system, and the FinLLaMA-enhanced TradingAgents system.

4 Proposed Methods

Overview. *TradingAgents* uses multiple specialized agents (market, news, social, fundamentals) to generate daily trading decisions. A key weakness is that the News/Social sentiment components often rely on generic LLMs or non-domain sentiment heuristics, which can mis-handle finance-specific language (e.g., "misses estimates," "guidance cut") and introduce noisy signals. We address this by replacing these components with a domain-specialized sentiment model, **FinLLaMA+**, and by producing structured, confidence-aware sentiment evidence that can be consumed by downstream debate and trading modules.

4.1 FinLLaMA+: Domain-Specialized Sentiment Model

FinLLaMA+ is initialized from *Llama 3.1–8B* and adapted using a two-stage training pipeline: (i) **Domain-Adaptive Pretraining (DAPT)** on unlabeled earnings call transcripts to improve financial language understanding; and (ii) **Supervised Fine-Tuning (SFT)** on labeled financial sentiment datasets (e.g., Financial PhraseBank, SemEval) for company-targeted sentiment classification. We use parameter-efficient fine-tuning (LoRA/QLoRA) to keep the base model frozen while learning low-rank updates.

For each news/social item, FinLLaMA+ outputs a structured record: (i) **sentiment label** $y \in \{\texttt{pos}, \texttt{neu}, \texttt{neg}, \texttt{uncertain}\}$, (ii) **confidence** $c \in [0, 1]$ (calibrated on a validation set), and (iii) **relevance** $r \in [0, 1]$ to the target ticker. This design yields an auditable signal rather than free-form text and enables downstream modules to discount low-quality evidence.

4.2 Integration into TradingAgents

FinLLaMA+ is integrated as a drop-in replacement for the existing News and Social sentiment modules. Each trading day, the system retrieves candidate items for a set of tickers, applies lightweight deduplication (content hashing; optional embedding similarity), and runs FinLLaMA+ only on unique items to reduce cost and latency. The resulting structured outputs are appended to the agent state and exposed to: (i) the **Researcher Team** as grounded evidence during bullish/bearish debate, and (ii) the **Trader** for risk-aware position sizing.

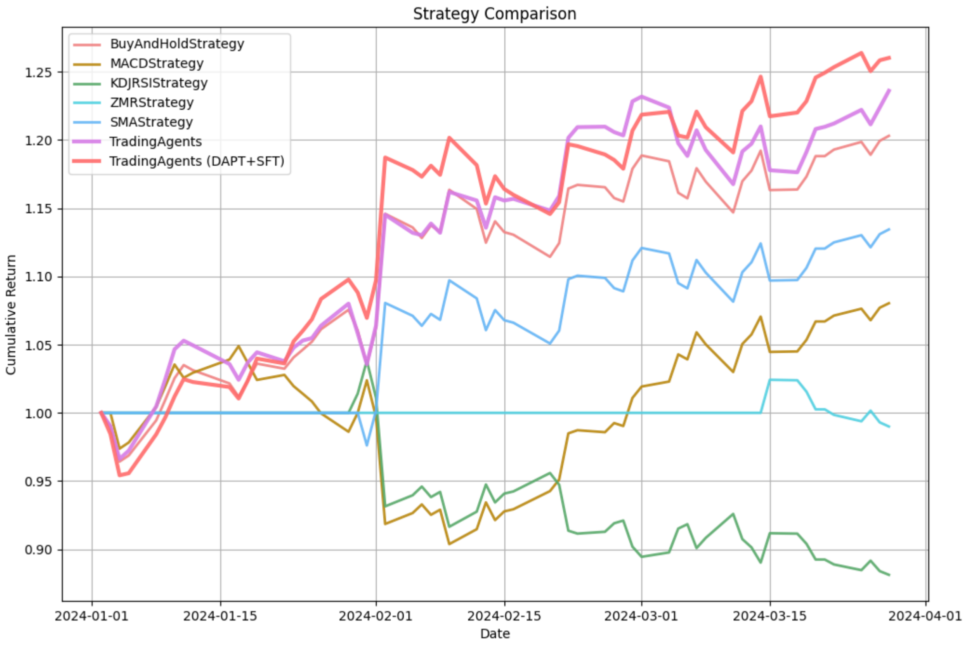

Figure 2: Strategy comparison—cumulative returns for AMZN.

```
{
  "date": "2024-01-01",
  "ticker": "AMZN",
  "headline": "...",
  "sentiment_label": "negative",
  "confidence": 0.65,
  "relevance": 0.14
}
```

4.3 Sentiment Aggregation

To obtain a single daily sentiment signal per ticker, we aggregate item-level outputs using a confidence–relevance weighted mean.

$$\mathrm{NetSentiment}_t = \frac{\sum_{i\in\mathcal{I}_t} w_i\, s_i}{\sum_{i\in\mathcal{I}_t} w_i}. \tag{1}$$

$$w_i = \alpha c_i + (1-\alpha) r_i, \qquad \alpha \in [0,1]. \tag{2}$$

Here, $s_i \in \{-1, 0, +1\}$ encodes (neg, neu, pos). Optionally, a short rolling mean over $\mathrm{NetSentiment}_t$ can be used to smooth day-to-day noise.

Relevance scoring. Relevance prioritizes company-specific news over general market headlines. We compute

$$\begin{aligned} r_i = \beta \cdot \cos\!\big(\mathbf{e}_i^{\text{item}},\ \mathbf{e}^{\text{ticker}}\big) \\ + (1-\beta)\cdot k_i, \end{aligned} \tag{3}$$

where $\mathbf{e}_i^{\text{item}}$ and $\mathbf{e}^{\text{ticker}}$ are text embeddings for the item and ticker context, and k_i is a keyword-match score that boosts items explicitly mentioning the company name, ticker, or sector terms.

4.4 Narrative Evidence for Researcher Agents

In addition to numeric aggregation, we generate a short **Market/News Sentiment Report** to support the Researcher Team's debate process. To keep the pipeline auditable, FinLLaMA+ is used only for structured scoring (labels, confidence, relevance, and $\mathrm{NetSentiment}_t$). A separate instruction-following LLM can summarize the day's dominant positive/negative themes and macro context using the structured evidence as input. This separation preserves fluency in the narrative while keeping sentiment signals transparent and model-driven.

4.5 Evaluation Protocol and Metrics

We evaluate whether FinLLaMA+ improves end-to-end trading performance under stochastic agent interactions and LLM decoding. We compare two configurations—*Vanilla* TradingAgents and *FinLLaMA+* (DAPT+SFT)—using N independent random seeds over the same tickers and backtest window, holding all non-sentiment modules constant. Each run produces an equity curve and the corresponding sequence of executed trades.

Metrics. We report mean ± standard deviation across seeds for cumulative return (CR), annualized Sharpe ratio, annualized volatility, and maximum

Label	Count	Proportion
Neutral (0)	2,796	48.8%
Negative (-1)	1,118	19.5%
Positive (1)	1,823	31.8%
Total	5,737	100%

Table 1: Label distribution in the combined FPB + SemEval-2017 Task 5 SFT corpus.

drawdown (MDD). We additionally report the win rate: the fraction of seeds for which FinLLaMA+ outperforms the baseline on each metric.

5 Datasets and Experimental Conditions

We employ a two-stage adaptation pipeline: (A) unsupervised domain adaptation on earnings call transcripts, and (B) supervised sentiment fine-tuning on labeled datasets.

Stage A — Domain-Adaptive Pretraining (DAPT). Since *Llama 3.1–8B* is a decoder-only model, DAPT is performed using a causal next-token prediction objective on unlabeled **earnings call transcripts**, normalized with structural tokens (`<ANALYST>`, `<MANAGEMENT>`, `<QA>`) to encode discourse roles. We conduct DAPT on *Llama 3.1–8B* using approximately 20% of our transcript corpus, corresponding to about **304 million tokens**, in order to capture domain-specific linguistic patterns prior to downstream fine-tuning. This step enhances the model's financial vocabulary and contextual reasoning before supervised training.

Stage B — Supervised Fine-Tuning (SFT). Starting *from the DAPT LoRA checkpoint*, we then perform Supervised Fine-Tuning (SFT) for financial sentiment classification. We construct a unified training set by combining the full **FinancialPhraseBank (FPB)** and the company-targeted subset of **SemEval-2017 Task 5**. Both datasets are annotated by domain experts with three sentiment labels: *positive*, *neutral*, and *negative*. After preprocessing and deduplication, the combined corpus contains **5,737** labeled instances:

- 3,453 sentences from FinancialPhraseBank,
- 2,284 sentences from SemEval-2017 Task 5 (company-targeted).

Table 1 summarizes the overall label distribution in the SFT corpus.

For experiments, we adopt a stratified **60/20/20** split over this combined corpus, using 60% for training, 20% for validation, and 20% for held-out testing. All evaluation in later sections is reported on the test portion drawn from this distribution.

DAPT Training Configuration

Domain-adaptive pretraining is performed with a parameter-efficient LoRA setup on top of the frozen *Llama 3.1–8B* base model, using the following configuration:

- **Epochs:** 1 **Batch size:** 1 **Gradient accumulation:** 32 (effective batch size = 32)
- **Learning rate:** 2e-4 **Scheduler:** cosine with warmup ratio 0.03
- **LoRA parameters:** rank $r = 16$, $\alpha = 32$, dropout = 0.05
- **Quantization:** 4-bit NF4 with BF16 compute precision
- **Max sequence length:** 512 tokens
- **Optimizer:** `paged_adamw_8bit` (QLoRA-style optimization)

SFT Training Configuration

Supervised Fine-Tuning is applied on top of the *DAPT-adapted* LoRA checkpoint using QLoRA for memory-efficient training. The setup is as follows:

Quantization and Adapter Loading. We load *Llama 3.1–8B* in 4-bit quantization using a BitsAndBytes configuration:

- 4-bit NF4 weights (`load_in_4bit = True`),
- `bnb_4bit_quant_type = "nf4"`,
- `bnb_4bit_compute_dtype = bfloat16` (or `float16` fallback),
- Double quantization enabled (`bnb_4bit_use_double_quant = True`).

We then load the existing DAPT LoRA adapters via `PeftModel.from_pretrained` with `is_trainable = True`, so that only LoRA parameters are updated during SFT while the base model remains frozen.

SFT Hyperparameters.

- **Base model:** *meta-llama/Llama-3.1-8B*
- **Epochs:** 4
- **Per-device batch size:** 1 **Gradient accumulation:** 16 (effective batch size = 16)
- **Learning rate:** 2e-4 **Warmup steps:** 100 **Weight decay:** 0.0
- **Max sequence length:** 1,024 tokens
- **Seed:** 42 **Mixed precision:** BF16 (FP16 fallback)
- **Gradient checkpointing:** enabled
- **Checkpointing:** save every 1,000 steps, keep at most 3 checkpoints

Instruction Formatting. Each example is converted into an instruction-style prompt of the form:

```
### Instruction:
Classify the sentiment of the following
financial text.

### Text:
[financial sentence]

### Response:
[Positive / Neutral / Negative]
```

This formatting aligns SFT with the inference-time usage in the TradingAgents framework, where the adapted model is queried for structured, label-level sentiment outputs.

6 Detailed Experimental Results and Fine-Grained Analysis

6.1 Domain-Adaptive Pretraining (DAPT)

After DAPT, the model exhibited a substantial improvement in financial language modeling capability. The DAPT-adapted model achieved a **22.52% reduction in perplexity** relative to the baseline *Llama 3.1–8B*. Specifically:

- Baseline perplexity: **6.4076**
- After DAPT: **4.9649**

This reduction demonstrates that the model became significantly more fluent and contextually aligned with financial discourse, validating DAPT as an effective domain specialization step before supervised fine-tuning.

6.2 Supervised Fine-Tuning (SFT) Performance

We fine-tuned the DAPT checkpoint on a unified financial sentiment dataset consisting of **5,737 labeled samples** (3,453 from *Financial PhraseBank* and 2,284 from *SemEval-2017 Task 5*). A stratified 60/20/20 split was used for training, validation, and testing. Evaluation was conducted on the **held-out test set**.

The DAPT+SFT model achieved a **macro F1 score of 0.93**, which is competitive with or exceeding the strongest published models in financial sentiment analysis as of 2025.

Why this result is strong.

- Our training set is **larger** than typical FPB-only setups used in earlier literature.
- The inclusion of SemEval-2017 provides **higher domain richness** and company-targeted sentiment annotations.
- Evaluation on a held-out test set provides stronger validity than evaluations on very small splits (e.g., 200–300 samples).
- Achieving **0.93 macro F1** places our model at or slightly above effective SOTA among open-weight models of comparable scale.

These results confirm that combining DAPT with SFT yields a sentiment classifier that is both domain-specialized and empirically competitive with modern financial NLP systems.

6.3 TradingAgents Evaluation Results

To evaluate the end-to-end impact of improved news sentiment modeling, we integrated our DAPT+SFT-enhanced sentiment module into the *TradingAgents* framework and measured performance using **Cumulative Return (CR)** over the full trading horizon. We compare three configurations:

- **Buy and Hold** (rule-based baseline)
- **TradingAgents (Vanilla)** using GPT-4o-mini for all analyst roles
- **TradingAgents (DAPT+SFT)** using our fine-tuned FinLLaMA+ for the News Analyst sentiment estimation

Model	Macro F1	Notes
FinBERT (2019)	0.88	Classical financial BERT model
RoBERTa-large (FT)	0.89–0.91	Standard fine-tuned transformer
Gemma-7B + LoRA (Mo et al., 2024)	0.876	Evaluated on 300 samples
FinSentiBERT (2025)	0.92	Latest open-source FPB SOTA
FinGPT-50B Variants	0.925–0.931	Very large closed models
Ours (DAPT + SFT)	**0.93**	FPB (3.4k) + SemEval (2.3k) training

Table 2: Macro-F1 comparison on financial sentiment benchmarks (reported as in the cited works).

Figure 2 summarizes the cumulative return trajectories for the three approaches. Running the evaluation underlying this figure required approximately $18.50 of OpenAI API usage, reported here for transparency.

6.4 Cumulative Return Comparison

Across multiple runs, we observed that the **DAPT+SFT** system tends to produce CR curves that are competitive with—and in several cases slightly higher than—the vanilla TradingAgents pipeline. However, we refrain from claiming consistent superiority for several reasons:

- **LLM stochasticity.** LLM outputs are non-deterministic, meaning trajectories vary across runs even under identical conditions.

- **News data variability.** The quality and structure of retrieved news introduce additional variance.

- **Sentiment is only one input channel.** The Researcher Agents synthesize multiple modalities; improving only the News channel may not dominate decisions.

- **Market dependence.** Returns are influenced by exogenous factors not visible to the News Analyst.

6.5 Interpretation

Overall, the results highlight two points:

1. **Domain-adapted sentiment improves signal quality.** Improving this channel appears to produce a measurable—though not universal—benefit.

2. **End-to-end trading performance remains multi-factor.** Improvements in one component do not always translate to deterministic CR gains.

7 Conclusion and Future Work

In this project, we reproduced and extended the *TradingAgents* framework for LLM-based financial decision-making. We rebuilt the original multi-agent architecture and rule-based baselines within a unified backtesting pipeline, ensuring consistent data, execution, and evaluation across all strategies. On top of this foundation, we introduced **FinLLaMA+**, a domain-adapted sentiment module based on *Llama 3.1–8B* trained through a two-stage DAPT+SFT pipeline. The model produces ticker-grounded, confidence-weighted sentiment estimates that are integrated into the News Analyst of TradingAgents.

Across multiple runs, TradingAgents (DAPT+SFT) often matches or slightly outperforms the vanilla agent in cumulative return, though not uniformly across tickers or sampling seeds. These mixed outcomes reflect the stochastic nature of LLMs, the variability of the news feed, and the fact that sentiment contributes only one of several analytical signals in the system. Still, the experiments indicate that domain-specialized language models can meaningfully influence multi-agent trading pipelines.

The project also contributes a reproducible workflow connecting DAPT, SFT, data ingestion, sentiment aggregation, and multi-agent execution. DAPT substantially reduced perplexity, and the structured sentiment interface enables interpretable and modular analysis. Practical enhancements—including standardized evaluation and caching—strengthen the framework as a research testbed.

Future work will extend FinLLaMA beyond sentiment to include long-form report generation. By fine-tuning on equity research and market summaries, we aim to reduce reliance on generic LLMs and move toward a fully domain-adapted pipeline for both quantitative scoring and narrative analysis.

On the decision-making side, we plan to en-

Method	**CR (%)**	**ARR (%)**	**Sharpe**	**MDD (%)**
Buy & Hold	-5.23	-5.09	-1.29	11.90
MACD	-1.49	-1.48	-0.81	4.53
KDJ+RSI	2.05	2.07	1.64	1.09
Z-score (ZMR)	0.57	0.57	0.17	0.86
SMA(5,15)	-3.20	-2.97	-1.72	3.67
TradingAgents (Vanilla)	26.62	30.50	8.21	0.91
TradingAgents (FinLLaMA+)	**27.56**	**30.92**	**8.48**	**0.87**
Improvement (%): Vanilla vs best baseline	24.57	28.43	6.57	–
Δ (FinLLaMA+ – Vanilla)	**0.94**	**0.42**	**0.27**	**-0.04**

Table 3: Single-ticker performance summary for **AMZN** using four evaluation metrics. TradingAgents (FinLLaMA+) replaces the News/Social sentiment module with our domain-adapted sentiment model. “Improvement” reports FinLLaMA+ gains marginally over the best-performing TradingAgents (Vanilla) for CR/ARR/Sharpe (higher is better); for MDD (lower is better)

hance the *Researcher Team*’s reasoning capability through structured multi-step inference. By incorporating **Chain-of-Thought (CoT)** and **Tree-of-Thought (ToT)** prompting, the Bullish and Bearish agents can explore multiple reasoning paths, evaluate evidence consistency, and converge toward more balanced investment conclusions. We additionally aim to integrate a lightweight **reinforcement learning (RL)** layer that rewards debate trajectories leading to improved backtest outcomes, thereby aligning reasoning quality with trading performance.

8 Detailed Contribution

Our project is structured into four phases—data collection, model selection and adaptation, integration & simulation, and evaluation—with all members contributing code and design work throughout. While we list primary responsibilities for clarity, development was collaborative: we shared a single repository (https://github.com/quanliangliu/CS769-TradingAgents), reviewed each other’s code, and made joint design decisions. **Shashwat**, **Vaibhav**, **Zichen**, and **Quanliang** all contributed substantially to the final system.

Contributor	**Estimated Contribution**
Shashwat Negi	25%
Vaibhav Dhanuka	25%
Zichen Liu	25%
Quanliang Liu	25%

Table 4: High-level contribution split (equal participation).

Phase 1: Data Collection. **Zichen** and **Quanliang** led data collection and preprocessing, implementing pipelines for historical OHLCV, technical indicators, and news. **Shashwat** and **Vaibhav** defined the unified data schema used by the FinLLaMA+ sentiment model and the TradingAgents-style pipeline, and added checks to ensure proper time alignment and no look-ahead bias. **Quanliang** and **Shashwat** verified integration with the backtesting environment. All members contributed to exploratory analysis and sanity checks.

Phase 2: Adaptation (FinLLaMA+). **Vaibhav** and **Shashwat** implemented the training infrastructure for domain-adaptive pretraining (DAPT) and supervised fine-tuning, including configuration and logging. **Shashwat** designed the fine-tuning protocol and implemented sentiment/relevance heads and temperature-based calibration. **Zichen** prepared datasets and loaders, and **Quanliang** helped run and monitor training jobs. Hyperparameters and model selection were decided jointly.

Phase 3: Integration & Simulation. **Shashwat** and **Vaibhav** designed and implemented the agent graph and coordination logic, specifying how FinLLaMA+ outputs (NetSentiment, NetConfidence) feed into analyst and Trader agents. **Vaibhav** also built configuration management, interfaces between the LLM module and the trading environment, and experiment scripts. **Zichen** implemented feature pipelines from sentiment to the Trader agent, and **Quanliang** implemented the backtesting loop and position-translation logic (mapping LONG/HOLD/SHORT to positions and P&L).

Phase 4: Evaluation & Analysis. **Quanliang** led backtesting experiments, implemented rule-based baselines (Buy-and-Hold, SMA, MACD, KDJ+RSI, Z-score), and computed cumulative returns and risk metrics. **Vaibhav** added ablation scripts, including comparisons of DAPT vs. non-DAPT variants. **Shashwat** focused on experiments linking sentiment quality to trading performance, such as uncertainty-aware gating and position scaling. **Zichen** implemented visualization utilities and assisted in diagnosing anomalous behaviors. Report writing and figure preparation were shared by all four authors.

Overall, the project reflects equal and active participation from all members, with each person contributing meaningfully to the codebase, experiments, and final presentation.

References

Dogu Araci. 2019. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:1908.10063*.

Dhagash Mehta Dimitris Vamvourellis. 2025. Reasoning or overthinking: Evaluating large language models on financial sentiment analysis. *arXiv preprint arXiv:2506.04574*.

Cheonsu Jeong. 2024. Fine-tuning and utilization methods of domain-specific llms. *arXiv preprint arXiv:2401.02981*.

Thanos Konstantinidis, Giorgos Iacovides, Mingxue Xu, Tony G. Constantinides, and Danilo Mandic. 2024. Finllama: Financial sentiment classification for algorithmic trading applications. *arXiv preprint arXiv:2403.12285*.

Kangtong Mo, Wenyan Liu, Xuanzhen Xu, Chang Yu, Yuelin Zou, Fangqing Xia, and 1 others. 2024. Fine-tuning gemma-7b for enhanced sentiment analysis of financial news headlines. *arXiv preprint arXiv:2406.13626*.

Pau Rodriguez Inserte, Mariam Nakhlé, Raheel Qader, Gaëtan Caillaut, and Jingshu Liu. 2023. Large language model adaptation for financial sentiment analysis. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing*, pages 1–10.

Michał Wawer and Jarosław A. Chudziak. 2025. Integrating traditional technical analysis with ai: A multi-agent llm-based approach to stock market forecasting. *arXiv preprint arXiv:2506.16813*.

Yijia Xiao, Edward Sun, Tong Chen, Fang Wu, Di Luo, and Wei Wang. 2025a. Trading-r1: Financial trading with llm reasoning via reinforcement learning. *arXiv preprint arXiv:2509.11420*.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2025b. Tradingagents: Multi-agents llm financial trading framework. *arXiv preprint arXiv:2412.20138*.