

# Evaluating Financial LLM Architectures on Consumer GPUs: Model Comparison and Temperature-Based Hyperparameter Tuning

MSc Research Project  
Artificial Intelligence

Bugra Kavdir  
Student ID: 24159972

School of Computing  
National College of Ireland

Supervisor: Mr. Arundev Vamadevan

**National College of Ireland  
Project Submission Sheet  
School of Computing**



|                             |                                                                                                                       |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>Student Name:</b>        | Bugra Kavdir                                                                                                          |
| <b>Student ID:</b>          | 24159972                                                                                                              |
| <b>Programme:</b>           | Artificial Intelligence                                                                                               |
| <b>Year:</b>                | 2025                                                                                                                  |
| <b>Module:</b>              | MSc Research Project                                                                                                  |
| <b>Supervisor:</b>          | Mr. Arundev Vamadevan                                                                                                 |
| <b>Submission Due Date:</b> | 11/12/2025                                                                                                            |
| <b>Project Title:</b>       | Evaluating Financial LLM Architectures on Consumer GPUs: Model Comparison and Temperature-Based Hyperparameter Tuning |
| <b>Word Count:</b>          | 5372                                                                                                                  |
| <b>Page Count:</b>          | 25                                                                                                                    |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

|                   |                   |
|-------------------|-------------------|
| <b>Signature:</b> | Bugra Kavdir      |
| <b>Date:</b>      | 28th January 2026 |

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

|                                                                                                                                                                                            |                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| Attach a completed copy of this sheet to each project (including multiple copies).                                                                                                         | <input type="checkbox"/> |
| <b>Attach a Moodle submission receipt of the online project submission</b> , to each project (including multiple copies).                                                                  | <input type="checkbox"/> |
| <b>You must ensure that you retain a HARD COPY of the project</b> , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | <input type="checkbox"/> |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

|                                  |  |
|----------------------------------|--|
| <b>Office Use Only</b>           |  |
| Signature:                       |  |
| Date:                            |  |
| Penalty Applied (if applicable): |  |

# Evaluating Financial LLM Architectures on Consumer GPUs: Model Comparison and Temperature-Based Hyperparameter Tuning

Bugra Kavdir  
24159972

## Abstract

This thesis investigates whether quantized, domain-specific Large Language Models (LLMs) deployed on consumer-grade hardware can generate profitable trading signals through technical analysis of cryptocurrency markets. We evaluate three models (Finance-LLM-13B, Finance-Llama-8B, and Phi-4) across 45 backtests spanning 15 sessions with heterogeneous hyperparameter configurations on Bitcoin/USDT 5-minute candles from November 2025. Finance-LLM-13B consistently achieves profitability against a -9.04% market baseline. We conduct controlled experiments isolating the effects of temperature and nucleus sampling (top-p) on trading signal quality. Critically, we demonstrate that domain-specific fine-tuning dominates model parameter scaling. Temperature-driven exploration benefits well-trained models but degrades weak models. Results emphasize that domain-specific training, rather than scale or sophisticated decoding strategies, is the dominant performance driver for financial reasoning. This work provides proof-of-concept evidence that LLM-based trading on consumer hardware is feasible, while establishing necessary conditions for robustness: multi-period validation and live-market validation before production deployment.

## 1 Introduction

### 1.1 Motivation and Research Gap

The intersection of artificial intelligence and financial markets presents a fundamental research opportunity: can Large Language Models (LLMs), trained to process and reason about natural language, effectively interpret technical analysis indicators and generate profitable trading signals? Traditional algorithmic trading systems encode trading decisions as rigid mathematical rules—they cannot incorporate qualitative market insights or adapt to unstructured information. Meanwhile, recent advances in LLMs demonstrate remarkable reasoning capability across diverse domains (medicine, law, mathematics) Brown et al. (2020), yet their application to technical trading analysis remains largely unexplored.

This gap is significant because cryptocurrency markets exhibit unique characteristics: continuous operation (24/7), extreme volatility (daily price swings of 5–10%), absence of fundamental valuation metrics, and pronounced sentiment-driven dynamics Urquhart (2016). Under these conditions, human traders heavily rely on technical analysis and

pattern recognition. The question naturally emerges: can LLMs match or exceed human-level technical analysis capability, and can they do so on consumer-grade hardware (RTX 3060 with 12 GB VRAM)?

The direct LLM interpretation of technical indicators—moving averages, RSI, MACD, Bollinger Bands—remains understudied. Additionally, quantization (reducing model precision from 16-bit to 4-bit) Frantar et al. (2022) offers a pathway to deploy large models on consumer hardware, but its impact on financial reasoning accuracy is unclear.

## 1.2 Research Questions

1. Can quantized open-source Large Language Models deployed on consumer-grade hardware generate statistically significant positive returns through technical analysis-based trading on cryptocurrency markets?
2. How do different LLM architectures (Finance-LLM-13B, Finance-Llama-8B, Phi-4) differ in trading performance, directional bias, and consistency?
3. What is the relative importance of domain-specific fine-tuning versus model parameter scaling for financial reasoning tasks?
4. How do hyperparameters (temperature, top-p) modulate trading signal quality?
5. What are the hardware bottlenecks limiting multi-agent LLM systems on consumer GPUs, and what architectural trade-offs resolve them?

## 1.3 Document Structure

Section 2 reviews LLM foundations, and cryptocurrency markets. Section 3 gives insights about backtesting conditions, performance metrics, and the dataset used. Section 4 details model selection and consumer-grade equipment limitations. Section 5 presents system architecture, prompt design, and the pipeline. Section 6 provides a comprehensive results analysis across experimental configurations. Section 7 discusses findings, limitations, and future directions.

# 2 Related Work

## 2.1 Large Language Models: Foundations and Capabilities

Transformers and attention-based architectures form the computational foundation of modern LLMs. The Transformer architecture, which utilizes self-attention mechanisms to process sequences in parallel, enabling training of models at unprecedented scale Vaswani et al. (2017). The key insight—that attention enables efficient parallel computation of semantic relationships—became the basis for scaling language models to billions of parameters. Pretraining on large text corpora followed by task-specific fine-tuning has been shown to achieve state-of-the-art performance across a wide range of NLP benchmarks. Devlin et al. (2018). Language models with 175 billion parameters, such as GPT-3, have been shown to exhibit in-context learning capability—the ability to perform tasks from minimal prompt-based examples without explicit fine-tuning Brown et al. (2020).

This emergence of in-context learning as the model scale increases has become central to LLM applications across domains.

Recent advances demonstrate that LLMs exhibit general-purpose reasoning capabilities. Reasoning in LLMs emerges through a combination of diverse pretraining data providing broad factual knowledge, scale enabling complex pattern recognition, and appropriate prompting strategies eliciting multi-step reasoning Wei et al. (2022). This foundation enables LLMs to work on specialized domains, including finance, by leveraging pretraining knowledge while adapting to task-specific constraints.

## 2.2 Domain-Specific Fine-Tuning and Transfer Learning

Fine-tuning—adapting a pretrained model to specialized tasks—has become standard in NLP. A 2020 study demonstrated that domain-specific fine-tuning on financial corpora (earnings transcripts, financial news) improves financial sentiment analysis by 15–25% over general-purpose models Araci (2019). The mechanism is intuitive: domain-specific corpora encode specialized vocabulary, statistical patterns, and domain concepts not present in general text, enabling the model to learn domain-aware representations.

Transfer learning extends this idea: knowledge learned on large source tasks transfers to target tasks with limited labeled data. This study provides a comprehensive survey of transfer learning, documenting that models with higher accuracy on source tasks transfer better to target tasks, and architectural similarity between source and target improves transfer efficiency Zhuang et al. (2020). For financial LLMs, this implies that models pre-trained on financial documents achieve superior downstream task performance compared to general-purpose models fine-tuned on limited financial data.

Recent work highlights that domain-specific pretraining can outweigh parameter scaling. This is supported by BERT-like models fine-tuned on scientific papers (SciBERT) achieve higher performance on scientific NLP tasks than much larger general-purpose models Beltagy et al. (2019). In our work, Finance-LLM-13B (13B parameters, domain-trained) outperforms Phi-4 (14B parameters, general-purpose), empirically supporting this claim: domain training dominates scale.

## 2.3 Quantization and Efficient Inference

Model quantization enables deployment on resource-constrained hardware with minimal accuracy loss. Quantization can reduce model size by  $4\text{--}8\times$  with  $<1\%$  accuracy degradation on traditional ML tasks Hubara et al. (2016). In LLMs, quantization becomes critical for enabling deployment on consumer GPUs: a 13-billion parameter model requires 26GB in 16-bit precision but only 6.5GB in 4-bit precision, within reach of consumer hardware.

Post-training quantization (PTQ) and quantization-aware training (QAT) are primary approaches. PTQ applies quantization after training, requiring calibration on a small dataset; QAT incorporates quantization into training, enabling higher compression but requiring retraining. GPTQ is a PTQ method for LLMs achieving  $3\text{--}4\times$  compression with minimal perplexity increase Frantar et al. (2022). Our work uses GGML-format 4-bit quantized models, following the GPTQ/GGML paradigm.

Trade-offs exist between quantization level and model quality. Quantization to 2-bit introduces unacceptable performance degradation for reasoning tasks (we observed 30%+ return loss in preliminary experiments), whereas 4-bit quantization provides a reasonable compression-quality trade-off. This highlights why “pushing quantization to limits” is not

a universal solution. Domain-specific reasoning appears more sensitive to precision loss than general purpose tasks.

## **2.4 Large Language Models for Financial Applications**

LLMs’ application to finance has accelerated recently, spanning sentiment analysis, price forecasting, and trading. Key applications include:

### **2.4.1 Sentiment Analysis and News-Based Prediction**

Multiple studies demonstrate LLM superiority over traditional sentiment methods in financial contexts. When GPT-3, BERT, and FinBERT were compared on 965,375 financial news articles, showing that OPT (an open-source GPT-3 equivalent) achieves 74.4% accuracy in predicting stock market returns compared to 51% for the Loughran-McDonald dictionary baseline Kirtac (2024). Crucially, a self-financing strategy based on OPT sentiment scores achieved a Sharpe ratio of 3.05 versus 1.23 for dictionary-based methods, demonstrating that LLM-generated signals have economically meaningful predictive power. This work validates that LLMs can extract financial insights superior to rule-based approaches.

### **2.4.2 Technical Analysis and Breakout Trading**

Direct LLM application to technical analysis remains understudied until recently. FinLLM-B, a domain-specialized LLM developed specifically for financial breakout detection demonstrated the power of domain adaptation by achieving 49.97% and 42.38% accuracy improvements over GPT-3.5 and ChatGPT-4 on financial breakout classification tasks Zhang, Yoshie, Sun and Huang (2024). This performance differential illustrates that specialized domain-trained models can substantially outperform general-purpose models on technical financial tasks. The FinLLM-B multi-stage reasoning framework—which separates feature extraction, feature analysis, and decision stages demonstrates conceptual parallels with our multi-agent architecture, both decompose complex financial reasoning into modular stages that can be optimized independently before integration.

### **2.4.3 Multi-Agent Trading Systems**

Recent work explores LLMs as autonomous trading agents in simulated markets where LLM agents trade in a realistic order book environment, demonstrating that LLMs can adhere to trading strategies, engage in price discovery, and exhibit emergent competitive behaviors Lopez-Lira (2024). This work validates that LLMs can maintain multi-step reasoning about market state, execute complex decisions (entry, position sizing, exit), and adapt behavior across market conditions. Our architecture builds on this framework, extending it to constrained hardware.

Additionally, StockAgent, a multi-agent system where LLM agents simulate investors responding to macroeconomic factors, policy changes, and company fundamentals Zhang, Li, Du and Zhang (2024). The system avoids test-set leakage by evaluating on disjoint time periods, addressing a critical methodological concern in backtesting. Their emphasis on realistic market simulation with diverse agent types informs our experimental design.

## 2.5 Cryptocurrency Markets and Technical Analysis

Cryptocurrency markets differ fundamentally from equity markets in ways motivating LLM-based trading:

### 2.5.1 Market Microstructure

Cryptocurrency markets operate 24/7 with extreme volatility (5–10% daily price swings are common), minimal market maker regulation, and high sentiment sensitivity. Urquhart (2016) documented that cryptocurrency markets exhibit inefficiency and anomalies exploitable by technical analysis, contrasting with efficient market hypothesis predictions Urquhart (2016). This inefficiency creates opportunities for learning-based models that identify patterns.

### 2.5.2 Absence of Fundamental Valuation

Unlike equities like stocks, cryptocurrencies lack intrinsic valuation. This forces traders to rely entirely on technical analysis (price patterns, volume, sentiment) rather than fundamental analysis. Technical trading rules generate positive returns in cryptocurrency markets, contrasting with traditional assets Balci et al. (2017). This makes cryptocurrencies an ideal testbed for technical LLM-based trading.

## 2.6 Hyperparameter Sensitivity and Sampling Strategies

Temperature and nucleus sampling are fundamental hyperparameters controlling LLM output diversity. Temperature  $\tau$  scales logits before softmax:  $\text{softmax}(\text{logits}/\tau)$ . Lower  $\tau$  (e.g., 0.1) concentrates probability mass on highest-probability tokens (exploitation); higher  $\tau$  (e.g., 1.0) flattens the distribution, enabling diverse sampling (exploration).

Nucleus sampling (top-p) truncates the probability distribution, sampling only from the top-p mass. Nucleus sampling shows it produces higher-quality text than top-k sampling Holtzman et al. (2019).

Self-consistency leverages temperature to sample multiple reasoning paths, uses predictions to reduce variance Wang et al. (2022). This principle informs our hypothesis that moderate temperature improves Finance-LLM-13B performance via exploration of diverse technical interpretations. However, self-consistency’s effectiveness depends on model quality.

## 2.7 Research Gaps and Motivation

Despite progress in LLM finance applications, significant gaps remain:

1. **Consumer Hardware Deployment:** Existing work assumes access to cloud compute or high-end GPUs. Feasibility on consumer hardware (RTX 3060, 12GB VRAM) remains unclear, limiting practical accessibility.
2. **Domain Training vs. Scale:** The relative importance of domain-specific pre-training versus model parameter scaling for financial reasoning is inadequately characterized. Our work empirically compares domain-trained models (13B, 8B) versus general-purpose models (14B).

3. **Hyperparameter Sensitivity:** Controlled studies of how temperature and nucleus sampling affect trading signal quality are absent. Our 4-configuration grid isolates these effects while holding other variables constant.

This thesis addresses these gaps by providing a comparison of domain-trained and general-purpose quantized LLMs on technical analysis trading, characterising hyperparameter effects, and documenting hardware constraints limiting multi-agent deployment on consumer GPUs.

## 3 Methodology

### 3.1 Dataset and Market Context

We use Bitcoin/USDT 5-minute candles from Binance in these testings to get consistent results:

Table 1: Dataset Specification

| Metric           | Value                  |
|------------------|------------------------|
| Period           | Nov 6–9, 2025 (4 days) |
| Candles          | 2,880 observations     |
| Timeframe        | 5-minute               |
| Starting Price   | \$104,742              |
| Ending Price     | \$95,276               |
| Market Return    | -9.04%                 |
| High             | \$105,440              |
| Low              | \$94,250               |
| Daily Volatility | 2.1%                   |
| Total Range      | 10.0% (H – L)          |

This period exhibits bear market conditions, challenging models with downward pressure. A total range of 10.0% ensures diverse price action, enabling signal testing.

### 3.2 Technical Indicators and Feature Engineering

For each candle, we compute:

- **RSI(14):**

$$\text{RSI} = 100 - \frac{100}{1 + \frac{\text{Avg Gain}}{\text{Avg Loss}}}$$

Ranges [0,100]; > 70 = overbought, < 30 = oversold.

- **MACD(12,26,9):**

$$\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26}, \quad \text{Signal} = \text{EMA}_9(\text{MACD})$$

Crossovers indicate trend changes.



- **Bollinger Bands:** Middle = SMA(20); Upper = Middle +  $2 \times \text{StdDev}$ ; Lower = Middle -  $2 \times \text{StdDev}$ . Price near extremes suggests reversion.
- **ATR(14):** Average True Range measures volatility. Position sizing uses: stop-loss =  $1.5 \times \text{ATR}$ , take-profit =  $3.5 \times \text{ATR}$ .

### 3.3 Backtesting Framework Design

The backtesting engine simulates live trading:

1. Compute indicators at each 5-min candle
2. Query Market Analyst LLM (bias + confidence)
3. Query Decision Maker LLM (action + size)
4. If action  $\neq$  HOLD: enter trade at candle close
5. Track trades: entry, exit, PnL
6. Record metrics: return, win/loss, directional bias

### 3.4 Performance Metrics

$$\text{Return (\%)} = \frac{\text{Final Balance} - \text{Initial Balance}}{\text{Initial Balance}} \times 100\%$$

$$\text{Win Rate (\%)} = \frac{\text{Winning Trades}}{\text{Total Trades}} \times 100\%$$

$$\begin{aligned} \text{Alpha (pp)} &= \text{Model Return (\%)} - \text{Market Return (\%)} \\ &= \text{Model Return (\%)} - (-9.04\%) \end{aligned}$$

#### 3.4.1 LLM Text Generation

LLMs generate text autoregressively, predicting tokens sequentially.

**Training Objective:**

$$\mathcal{L} = \sum_{t=1}^T \log P(w_t \mid w_1, \dots, w_{t-1}; \theta)$$

**Sampling Strategies:**

- **Greedy Decoding:** choose  $\arg \max_w P(w \mid \text{context})$  (deterministic)
- **Temperature Sampling:**

$$P(w) \propto \exp\left(\frac{\text{logit}_w}{T}\right)$$

- $T < 1$ : sharper, focused
- $T > 1$ : more random

- **Top-k Sampling:** sample from top- $k$  tokens
- **Top-p Sampling:** minimal cumulative mass  $\geq p$  (nucleus) Holtzman et al. (2019)

## 4 Design Specifications

### 4.1 Model Selection

We evaluate three models representing different points on the scale–domain trade-off:

Table 2: Model Comparison: Scale vs. Domain

| Model            | Parameters | Domain-Tuned | VRAM    | Inference |
|------------------|------------|--------------|---------|-----------|
| Finance-LLM-13B  | 13B        | Yes          | 10.4 GB | 13–16 sec |
| Finance-Llama-8B | 8B         | Yes          | 9.2 GB  | 14–17 sec |
| Phi-4            | 14B        | No           | 10.8 GB | 8–11 sec  |

**Finance-LLM-13B:** Domain-tuned on financial documents. Expected to understand trading concepts natively.

**Finance-Llama-8B:** Smaller, domain-tuned variant. Tests whether efficiency sacrifices performance.

**Phi-4:** General-purpose model without financial pre-training, used to test the importance of domain specialization.

### 4.2 Original Multi-Timeframe Architecture

The initial design aspired toward professional-grade multi-timeframe analysis (MTA):

$$\text{Decision} = \text{Aggregate}(\text{LLM}_{5\text{m}}, \text{LLM}_{15\text{m}}, \text{LLM}_{1\text{h}}).$$

Each LLM instance analyzes price action at its respective timeframe, and a Decision Aggregator synthesizes their signals.

### 4.3 Memory Requirements Analysis

For three concurrent **Finance-LLM-13B** instances, the total VRAM requirement is

$$\text{Total VRAM} = 3 \times (\text{Weights} + \text{KV cache} + \text{Activations}).$$

Per-instance breakdown:

$$\begin{aligned} \text{Weights} &= 13 \times 10^9 \text{ params} \times 0.5 \text{ bytes/param} \\ &\approx 6.5 \text{ GB}, \end{aligned}$$

$$\begin{aligned} \text{KV cache} &= 2 \times 2048 \times 40 \times 5120 \times 0.5 \text{ bytes} \\ &\approx 2.1 \text{ GB}, \end{aligned}$$

$$\text{Activations} \approx 1.5 \text{ GB}.$$

Thus, per-instance VRAM usage is

Per-instance total  $\approx 10.1$  GB.

For three agents:

Total for 3 agents =  $3 \times 10.1$  GB  $\approx 30.3$  GB.

On an RTX 3060 with 12 GB of VRAM, this is impossible:

Deficit  $\approx 30.3$  GB  $- 12$  GB =  $18.3$  GB ( $\approx 255\%$  over budget).

Attempted solutions:

1. **Sequential loading:** Unload Agent 1, load Agent 2, etc.
2. **Extreme quantization:** Two-bit models reduce usage to roughly 3.2 GB, but quality becomes unacceptable.
3. **CPU offloading:** Keep models in system RAM and swap to GPU at inference time.

**All solutions failed to meet latency or quality requirements.**

## 4.4 Problem: KV Cache Scaling

The Transformer attention mechanism computes similarity between all token pairs:

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

This requires storing Key and Value vectors for all positions, leading to

$\mathcal{O}(n^2)$  space complexity,

where  $n$  is the sequence length. When sequence length doubles,

$$\frac{\text{KV\_cache}_{2048}}{\text{KV\_cache}_{1024}} = \frac{2048^2}{1024^2} = 4.$$

Thus, concurrent multi-agent systems inherently require roughly three times the KV-cache storage with no sharing opportunity, because each agent maintains an independent computation graph.

### 4.4.1 format specifications

LLMs exhibit pronounced sensitivity to output format specifications:

- **JSON vs. natural text:** Structured output formats like JSON significantly improve parsing reliability and eliminate parsing errors compared to natural language outputs Wang et al. (2025). However, a model’s ability to generate valid JSON depends on its code pretraining exposure; models with extensive code-data exposure in training (such as GPT-4) handle complex JSON structures reliably, while models with limited code training may produce malformed JSON Achiam et al. (2023).

- **Chat template requirements:** Models such as Phi-4 require strict chat formatting conventions (e.g., `<|im_start|>role<|im_sep|>content<|im_end|>`), and often fail when given plain-text prompt.
- **Delimiter selection:** Prior work on BLOOM and related prompted training shows that the choice of task and language markers, including special tokens versus purely natural-language formatting, can substantially affect model performance and behavior Scao et al. (2022).

We leverage these insights by tailoring prompts to each model’s architectural requirements rather than applying uniform prompting strategies.

## 5 Implementation

### 5.1 Solution: Integrated 2-Agent Architecture

Our system comprises two LLM agents:

Table 3: Agent Specifications

| Agent          | Input                                                                              | Task                           | Output                                                         |
|----------------|------------------------------------------------------------------------------------|--------------------------------|----------------------------------------------------------------|
| Market Analyst | Technical indicators (RSI, MACD, Bollinger, ATR) + historical win rates (Optional) | Technical analysis             | Directional bias (bullish/bearish/neutral) + confidence [0, 1] |
| Decision Maker | Analyst recommendation + current market context                                    | Deterministic trading decision | Action (BUY/SELL/HOLD) + position multiplier [0.5, 2.0]        |

Benefits of two-agent design:

1. **Separation of Concerns:** Analysis and decision-making require different reasoning styles
2. **Prompt Specialization:** Market Analyst benefits from exploring; Decision Maker from deterministic prompting
3. **Error Isolation:** Failure in one agent doesn’t cascade to both
4. **Flexibility:** Can replace agent independently without retraining

## 5.2 System Architecture and Pipeline

The system implements a multi-agent LLM trading framework with three core components:

1. **Data Layer:** Price fetching (Binance API) and technical indicator computation
2. **Agent Layer:** Market Analyst  $\rightarrow$  Decision Maker  $\rightarrow$  Feedback Agent
3. **Execution Layer:** Risk management, paper trading, and position monitoring

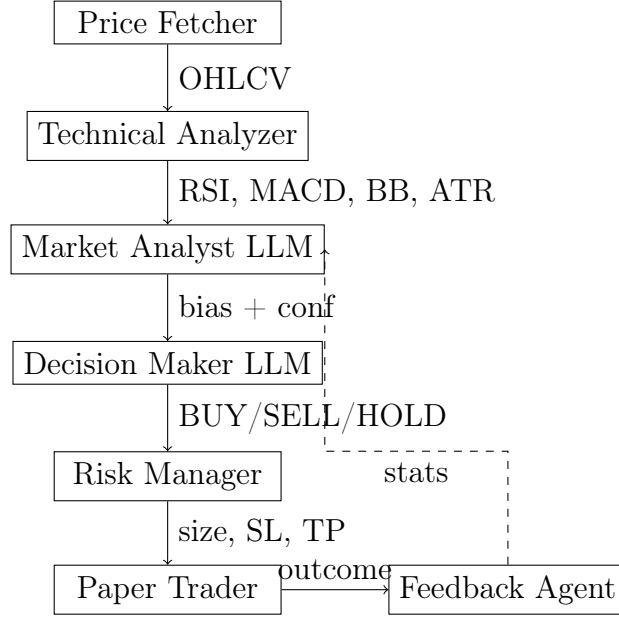


Figure 1: System Data Flow Pipeline

### 5.2.1 Detailed Execution Flow

#### Step 1: Data Acquisition

- Fetch 100 recent 5-minute candles from Binance API (`utils/price_fetcher.py`)
- In backtest mode: load from cached `backtest_data_10d_5m.json` (2,880 candles)
- Output: `[[timestamp, open, high, low, close, volume], ...]`

#### Step 2: Technical Indicator Computation

We calculate technical indicators using vectorized NumPy operations:

- **RSI(14):** Relative Strength Index
- **MACD(12,26,9):** Moving Average Convergence Divergence with histogram
- **Bollinger Bands:**  $SMA(20) \pm 2\sigma$
- **ATR(14):** Average True Range

### Step 3: Market Analyst Agent

The Market Analyst receives technical indicators formatted as human-readable text and generates a directional market bias (bullish/bearish/neutral) with an associated confidence score (0.0-1.0). The agent’s prompt structure and generation parameters are dynamically selected based on the detected model type to optimize reliability and output quality. The system automatically detects the model architecture from the filename and selects appropriate prompt templates:

- **Finance-13B Model:** Receives structured JSON prompts with explicit schema definitions. The 13B parameter model demonstrates reliable JSON formatting capabilities, allowing for rich structured output including nested objects for key levels, risk factors, and timeframe alignment. The prompt explicitly requests:
- **8B and Phi-4 Models:** Receive simplified text-based prompts to reduce parsing complexity. Empirical testing revealed that smaller models produce format errors (e.g., malformed JSON, inconsistent key names) approximately 15-20% of the time with complex structured prompts. This aligns with findings showing that model size directly affects prompt format robustness, smaller models like Phi-3-Mini and Llama 3.1-8B exhibit significantly higher error rates when generating structured outputs compared to larger counterparts Schwettmann et al. (2024).

The system then parses this text output using regular expressions to extract structured data, with fallback mechanisms for edge cases.

#### Input Context Window:

Each inference receives:

1. **Last 10 OHLCV candles** (formatted as readable text tables)
2. **7 technical indicators:** RSI(14), MACD histogram, Bollinger Band position, ATR(14), volume change, price action classification, and volatility regime

### Step 4: Decision Maker Agent

The Decision Maker agent receives the Market Analyst’s bias and confidence, along with current account status (balance, open positions), and produces a final executable trading action: BUY, SELL, or HOLD. This agent operates with deterministic parameters to ensure consistent risk management decisions.

#### Deterministic Execution via Zero Temperature:

The Decision Maker uses **temperature = 0.0**, which modifies the probability distribution over next-token predictions such that:

$$P(w_i|\text{context}, \tau = 0) = \begin{cases} 1 & \text{if } \text{logit}_i = \max_j \text{logit}_j \\ 0 & \text{otherwise} \end{cases}$$

This eliminates stochastic sampling entirely—the model always selects the single highest-probability token at each generation step. Combined with a fixed random seed (`seed=42`), this ensures:

- **Reproducibility:** Identical input  $\rightarrow$  identical output across multiple runs
- **Consistency:** The same market condition + analyst bias + account state always produces the same decision

- **Debuggability:** Eliminates variance as a confounding variable when diagnosing unexpected trading behavior

However, zero temperature alone does not guarantee identical outputs if:

1. Feedback system is enabled (historical performance changes between runs)
2. Input candles differ (live data vs. cached backtest data)
3. Model receives different analyst biases (due to analyst's non-zero temperature)

For true determinism in experiments, we disable feedback (`enabled=False`) and use fixed backtest datasets.

#### **Input Context:**

The Decision Maker receives a condensed prompt containing:

- **Analyst recommendation:** bias (bullish/bearish/neutral), confidence (0.0-1.0), reasoning text
- **Account status:** current balance (USD), number of open positions, maximum allowed positions
- **Risk constraints:** maximum position size percentage, current exposure

#### **Step 5: Risk Management (Position Sizing)**

Dynamic position sizing formula:

$$\text{Position Size} = \text{Balance} \times 0.08 \times C_{\text{mult}} \times V_{\text{mult}}$$

$$C_{\text{mult}} = 0.5 + 1.5 \times \text{confidence} \quad (\text{range: } [0.5, 2.0])$$

$$V_{\text{mult}} = \begin{cases} 0.7 & \text{high volatility regime} \\ 1.0 & \text{normal volatility} \\ 1.2 & \text{low volatility regime} \end{cases}$$

Stop-loss (SL) and take-profit (TP) calculation:

$$\text{SL} = \text{Entry} \pm 1.5 \times \text{ATR}_{14}$$

$$\text{TP} = \text{Entry} \pm 3.5 \times \text{ATR}_{14}$$

(sign depends on direction: + for long, - for short). Risk-reward ratio:  $3.5/1.5 \approx 2.33$ .

#### **Step 6: Trade Execution (Paper Trading)**

- Deduct entry fee:  $\text{Fee} = \text{Position Size} \times \text{Leverage} \times 0.05\%$
- Record position in `open_positions` list with entry price, SL, TP, timestamp
- If action = HOLD, skip to position monitoring

#### **Step 7: Position Monitoring (Exit Logic)**

1. For each open position, check the current price against SL/TP on every candle.
2. Positions close on: stop-loss hit, take-profit hit, or manual closure.

### **Step 8: Feedback Recording (Learning Loop)**

- **This is only injected in real time tests. In our evaluation backtests we did not inject this.**
- Store trade outcomes in SQLite database
- Aggregate recent performance (last 24h): long/short win rates, average PnL
- Inject feedback into next Market Analyst prompt (e.g., “Long trades: 30% win rate, Short trades: 45% win rate”)
- Planning to adaptive learning without model retraining

## **6 Evaluation**

### **6.1 Mixed-Hyperparameter Evaluation**

Table 4 aggregates 45 backtests obtained under a heterogeneous set of sampling hyperparameters. Each session may use a different combination of temperature and top-p, including extreme configurations used as edge cases. As a result, the table is best interpreted as a mixed-environment stress test that shows us the importance of correct hyperparameter tuning.



Table 4: Full Backtest Grid: All Sessions and Models

| Session | Model                             | Return (%) | Win rate (%) | Trades | Long (%) | Short (%) |
|---------|-----------------------------------|------------|--------------|--------|----------|-----------|
| S1      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -6.61      | 27.27        | 22     | 63.64    | 36.36     |
| S1      | finance-11m-13b.Q4_K_M.gguf       | 3.70       | 38.10        | 21     | 52.38    | 47.62     |
| S1      | phi-4-q4.gguf                     | -50.75     | 22.73        | 22     | 13.64    | 86.36     |
| S2      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -11.54     | 24.00        | 25     | 52.00    | 48.00     |
| S2      | finance-11m-13b.Q4_K_M.gguf       | -29.23     | 30.43        | 23     | 60.87    | 39.13     |
| S2      | phi-4-q4.gguf                     | -52.94     | 22.73        | 22     | 18.18    | 81.82     |
| S3      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -13.80     | 22.73        | 22     | 0.00     | 100.00    |
| S3      | finance-11m-13b.Q4_K_M.gguf       | 13.20      | 47.62        | 21     | 28.57    | 71.43     |
| S3      | phi-4-q4.gguf                     | -26.66     | 17.39        | 23     | 8.70     | 91.30     |
| S4      | Finance-Llama-8B-GGUF-q4_K_M.gguf | 8.67       | 23.81        | 21     | 38.10    | 61.90     |
| S4      | finance-11m-13b.Q4_K_M.gguf       | 1.48       | 31.82        | 22     | 36.36    | 63.64     |
| S4      | phi-4-q4.gguf                     | -19.27     | 18.18        | 22     | 9.09     | 90.91     |
| S5      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -13.87     | 30.00        | 20     | 0.00     | 100.00    |
| S5      | finance-11m-13b.Q4_K_M.gguf       | 5.82       | 33.33        | 21     | 14.29    | 85.71     |
| S5      | phi-4-q4.gguf                     | -50.70     | 22.73        | 22     | 27.27    | 72.73     |
| S6      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -8.51      | 25.00        | 20     | 5.00     | 95.00     |
| S6      | finance-11m-13b.Q4_K_M.gguf       | 27.12      | 52.38        | 21     | 14.29    | 85.71     |
| S6      | phi-4-q4.gguf                     | -54.74     | 18.18        | 22     | 22.73    | 77.27     |
| S7      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -8.23      | 25.00        | 20     | 10.00    | 90.00     |
| S7      | finance-11m-13b.Q4_K_M.gguf       | -36.90     | 45.83        | 24     | 16.67    | 83.33     |
| S7      | phi-4-q4.gguf                     | -61.27     | 16.67        | 24     | 20.83    | 79.17     |
| S8      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -13.87     | 30.00        | 20     | 0.00     | 100.00    |
| S8      | finance-11m-13b.Q4_K_M.gguf       | 5.82       | 33.33        | 21     | 14.29    | 85.71     |
| S8      | phi-4-q4.gguf                     | -50.70     | 22.73        | 22     | 27.27    | 72.73     |
| S9      | Finance-Llama-8B-GGUF-q4_K_M.gguf | -8.51      | 25.00        | 20     | 5.00     | 95.00     |
| S9      | finance-11m-13b.Q4_K_M.gguf       | 27.12      | 52.38        | 21     | 14.29    | 85.71     |
| S9      | phi-4-q4.gguf                     | -54.74     | 18.18        | 22     | 22.73    | 77.27     |
| S10     | Finance-Llama-8B-GGUF-q4_K_M.gguf | -12.00     | 20.00        | 20     | 5.00     | 95.00     |
| S10     | finance-11m-13b.Q4_K_M.gguf       | 19.25      | 47.62        | 21     | 28.57    | 71.43     |
| S10     | phi-4-q4.gguf                     | -54.74     | 18.18        | 22     | 22.73    | 77.27     |
| S11     | Finance-Llama-8B-GGUF-q4_K_M.gguf | -6.83      | 18.18        | 22     | 31.82    | 68.18     |
| S11     | finance-11m-13b.Q4_K_M.gguf       | -0.65      | 28.57        | 21     | 38.10    | 61.90     |
| S11     | phi-4-q4.gguf                     | -45.08     | 18.18        | 22     | 22.73    | 77.27     |
| S12     | Finance-Llama-8B-GGUF-q4_K_M.gguf | -1.70      | 23.81        | 21     | 9.52     | 90.48     |
| S12     | finance-11m-13b.Q4_K_M.gguf       | 18.53      | 47.62        | 21     | 19.05    | 80.95     |
| S12     | phi-4-q4.gguf                     | -57.98     | 25.00        | 24     | 16.67    | 83.33     |
| S13     | Finance-Llama-8B-GGUF-q4_K_M.gguf | -12.47     | 17.39        | 23     | 13.04    | 86.96     |
| S13     | finance-11m-13b.Q4_K_M.gguf       | 32.34      | 54.55        | 22     | 13.64    | 86.36     |
| S13     | phi-4-q4.gguf                     | -54.15     | 29.17        | 24     | 12.50    | 87.50     |
| S14     | Finance-Llama-8B-GGUF-q4_K_M.gguf | 12.26      | 33.33        | 21     | 19.05    | 80.95     |
| S14     | finance-11m-13b.Q4_K_M.gguf       | 27.20      | 57.14        | 21     | 14.29    | 85.71     |
| S14     | phi-4-q4.gguf                     | -59.21     | 25.00        | 24     | 16.67    | 83.33     |
| S15     | Finance-Llama-8B-GGUF-q4_K_M.gguf | -11.37     | 28.57        | 21     | 0.00     | 100.00    |
| S15     | finance-11m-13b.Q4_K_M.gguf       | 14.22      | 47.62        | 21     | 14.29    | 85.71     |
| S15     | phi-4-q4.gguf                     | -52.44     | 22.73        | 22     | 22.73    | 77.27     |

Across this mixed environment, the three models exhibit a very consistent ranking in terms of realised return and win rate.

- **finance-11m-13b.Q4\_K\_M.gguf** is the only model that produces positive returns (roughly +1% to +6% in weaker configurations and approaching +30% in the strongest cases), and win rates frequently in the 45–55% band or higher.
- **Finance-Llama-8B-GGUF-q4\_K\_M.gguf** tends to cluster around small losses, often between about −6% and −14%, with win rates around 20–30%.
- **phi-4-q4.gguf** is loss-making with large negative returns and win rates near 20%.

This pattern indicates that, under identical data, execution, and risk rules, the choice of base LLM dominates performance: simply substituting `finance-llm-13b` for the other two models is enough to shift the backtested strategy from systematically losing to often profitable.

The long/short distributions provide an additional dimension of analysis. All three models exhibit a strong short bias in this experiment, with short trades often accounting for 70–90% of total positions, especially for `Finance-Llama-8B` and `Phi-4`. Despite this shared directional tilt, only `finance-llm-13b` converts the bias into net profit. This suggests that the key differentiator is not the sign of exposure but the model’s ability to time entries and exits within that exposure, consistent with the idea that profitable algorithmic trading requires both a correct directional prior and sufficiently accurate conditional forecasting.

## 6.2 Controlled Hyperparameter Evaluation

To evaluate how sampling hyperparameters affect strategy profitability, we designed a controlled 4-configuration evaluation where the Market Analyst agent operates under different settings while all other elements remain constant: identical historical data, execution logic, position sizing rules, and a deterministic Decision Maker ( $\tau = 0.0$ ,  $p = 0.1$ ).

Table 5: Experimental Configurations

| Config | Type                    | Temperature | Top-p |
|--------|-------------------------|-------------|-------|
| 1      | Baseline (Conservative) | 0.10        | 0.5   |
| 2      | High Diversity          | 0.50        | 0.5   |
| 3      | Moderate Exploration    | 0.85        | 0.5   |
| 4      | Top-p test              | 0.50        | 0.9   |

1. **Config 1 (Conservative):**  $\tau = 0.10$ ,  $p = 0.5$  — Highly deterministic, favours exploitation of high-probability tokens.
2. **Config 2 (Moderate Exploration):**  $\tau = 0.50$ ,  $p = 0.5$  — Balanced exploration of alternative reasoning paths.
3. **Config 3 (High Exploration):**  $\tau = 0.85$ ,  $p = 0.5$  — Encourages diverse interpretations of market signals.
4. **Config 4 (Nucleus Sampling Variant):**  $\tau = 0.50$ ,  $p = 0.9$  — Tests the effect of relaxed token truncation while maintaining moderate stochasticity.

Table 6: Comprehensive Results: All 3 Models  $\times$  4 Configurations

| Model            | Config    | Return (%)    | vs Market | WR (%) | Trades | Long (%) | Short (%) |
|------------------|-----------|---------------|-----------|--------|--------|----------|-----------|
| Finance-LLM-13B  | 1 (Base)  | 14.22         | +23.26    | 47.61  | 21     | 14.28    | 85.71     |
|                  | 2 (Div)   | 27.19         | +36.23    | 57.14  | 21     | 14.28    | 85.71     |
|                  | 3 (Exp)   | <b>32.33</b>  | +41.37    | 54.54  | 22     | 13.63    | 86.36     |
|                  | 4 (Top-p) | 16.43         | +25.47    | 39.13  | 23     | 43.47    | 56.52     |
| Finance-Llama-8B | 1 (Base)  | -11.37        | -2.33     | 28.57  | 21     | 0        | 100       |
|                  | 2 (Div)   | <b>12.26</b>  | +21.3     | 33.33  | 21     | 19.04    | 80.95     |
|                  | 3 (Exp)   | -12.47        | -3.43     | 17.39  | 23     | 13.04    | 86.95     |
|                  | 4 (Top-p) | -0.41         | +8.63     | 22.72  | 22     | 40.90    | 59.09     |
| Phi-4            | 1 (Base)  | <b>-52.44</b> | -43.40    | 22.72  | 22     | 22.72    | 77.27     |
|                  | 2 (Div)   | -59.21        | -50.17    | 25     | 24     | 16.66    | 83.33     |
|                  | 3 (Exp)   | -54.14        | -45.1     | 29.16  | 24     | 12.5     | 87.5      |
|                  | 4 (Top-p) | -52.95        | -43.91    | 18.18  | 22     | 9.09     | 90.90     |

### 6.2.1 Finance-LLM-13B

#### Configuration Performance Summary

Finance-LLM-13B is the only model profitable across all four configurations:

Table 7: Finance-LLM-13B Performance Across Configurations

| Config        | Return (%) | vs Market | Win Rate (%) | Trades |
|---------------|------------|-----------|--------------|--------|
| 1 (Base)      | 14.22      | +23.26    | 47.61        | 21     |
| 2 (Moderate)  | 27.19      | +36.23    | 57.14        | 21     |
| 3 (High Exp.) | 32.33      | +41.37    | 54.54        | 22     |
| 4 (Top-p)     | 16.43      | +25.47    | 39.13        | 23     |

#### Key Observations:

- Performance Range:** Returns span from +14.22% to +32.33%, a 18.11 percentage point spread. The best configuration (Config 3, high exploration) outperforms the baseline by  $32.33\% - 14.22\% = 18.11$  percentage points, or a  $1.275\times$  improvement factor.
- Win Rate Consistency:** Win rates remain elevated across all configs (39–57%), well above the theoretical 50% break-even threshold for strategies with favorable risk-reward ratios. The range indicates the model maintains signal quality regardless of sampling regime, though higher exploration (Config 2 and 3) achieves the best win rates (57.14% and 54.54%).
- Temperature Effect:** Recent work on LLM reasoning demonstrates that sampling multiple reasoning paths and marginalising predictions reduces variance, a principle termed self-consistency. For Finance-LLM-13B, this manifests as follows: At low temperature, the model commits strongly to its highest-confidence interpretation of the market. If that interpretation is correct, determinism yields clean signals. However, if the interpretation is partially incorrect or incomplete, the model lacks flexibility to adapt. At moderate-to-high temperature, the model internally explores alternative technical narratives. By integrating multiple interpretations,

the model’s signals become more robust to false positives. Win rate improvement from 47.61% to 57.14% supports this explanation. The model is not simply generating more trades (21 trades in both), but generating better trades by exploring diverse reasoning.

4. **Nucleus Sampling Effect (Config 4):** Increasing top-p from 0.5 to 0.9 while holding temperature at 0.50 produces a slight degradation: return drops from 27.19% to 16.43% (-10.76 pp). This suggests that relaxing nucleus truncation introduces unwanted token diversity without compensating benefits. The model generates less focused reasoning.
5. **Market Alpha Generation:** Even in the baseline configuration, Finance-LLM-13B generates +23.26pp alpha against a  $-9.04\%$  market return. In the optimal configuration (Config 3), it delivers +41.37pp alpha—the model returns +32.33% while the market loses -9.04%, a 41.37 percentage point outperformance. This is genuine alpha, not trend following.

### 6.2.2 Finance-Llama-8B

#### Configuration Performance Summary

Finance-Llama-8B shows dramatic sensitivity to hyperparameter changes, with returns ranging from  $-12.47\%$  to  $+12.26\%$ :

Table 8: Finance-Llama-8B Performance Across Configurations

| Config        | Return (%) | vs Market | Win Rate (%) | Trades |
|---------------|------------|-----------|--------------|--------|
| 1 (Base)      | -11.37     | -2.33     | 28.57        | 21     |
| 2 (Moderate)  | 12.26      | 21.30     | 33.33        | 21     |
| 3 (High Exp.) | -12.47     | -3.43     | 17.39        | 23     |
| 4 (Top-p)     | -0.41      | 8.63      | 22.72        | 22     |

#### Key Observations:

1. **Extreme Volatility:** Performance spans 24.73 percentage points indicating far greater hyperparameter sensitivity. Only one configuration (Config 2) produces a positive return.
2. **Lack of Robustness to Temperature:** Unlike Finance-LLM-13B which improves with higher temperature, Finance-Llama-8B degrades from Config 2 ( $\tau = 0.50$ , +12.26%) to Config 3 ( $\tau = 0.85$ , -12.47%). The -24.73pp swing suggests the model lacks a stable financial reasoning foundation. When forced to explore alternative interpretations with higher temperature, it generates incoherent or incorrect signals rather than diverse but valid ones.
3. **Win Rate Instability:** Win rates fluctuate between 17.39% and 33.33%, with the only profitable configuration (Config 2) achieving just 33.33% win rate. Even at 33% win rate, achieving +12.26% return requires winning trades must be large enough to overcome two losses for every win. This is structurally fragile compared to Finance-LLM-13B’s %50+ percent win rates. The model fails to protect capital in a  $-9.04\%$  market.

#### 4. Directional Bias Analysis:

Table 9: Finance-Llama-8B: Long/Short Distribution

| Config        | Long (%) | Short (%) | Dominant Bias |
|---------------|----------|-----------|---------------|
| 1 (Base)      | 0        | 100       | Extreme Short |
| 2 (Moderate)  | 19.05    | 80.95     | Short         |
| 3 (High Exp.) | 13.04    | 86.96     | Extreme Short |
| 4 (Top-p)     | 40.90    | 59.09     | Balanced      |

Configs 1 and 3 generate 100% and 87% short trades. In a bear market, this bias happens to align with price direction however the model performed bad in this bias,

#### Plausible Cause: Training Data Imbalance

Finance-Llama-8B’s extreme short bias and performance instability suggest the model was trained on a dataset skewed toward bearish examples or financial crisis narratives. At low temperature, this prior manifests as 100% shorts. At moderate temperature, the prior weakens slightly, and the model occasionally discovers profitable short trades. At high temperature, the prior strengthens again as the model explores noisy alternative interpretations.

This is distinct from Finance-LLM-13B’s behavior, where higher temperature improved performance by enabling diverse reasoning. For Finance-Llama-8B, diversity does not mean coherent alternatives; it means noise that amplifies the underlying bias.

#### 6.2.3 Phi-4

##### Configuration Performance Summary

Phi-4 is unprofitable across all four configurations:

Table 10: Phi-4 Performance Across Configurations

| Config        | Return (%) | vs Market | Win Rate (%) | Trades |
|---------------|------------|-----------|--------------|--------|
| 1 (Base)      | -52.44     | -43.40    | 22.72        | 22     |
| 2 (Moderate)  | -59.21     | -50.17    | 25.00        | 24     |
| 3 (High Exp.) | -54.15     | -45.11    | 29.17        | 24     |
| 4 (Top-p)     | -52.95     | -43.91    | 18.18        | 22     |

#### Key Observations:

1. **Market Underperformance:** In a -9.04% market, Phi-4 loses -52% to -59%. This is worse than a random strategy, suggesting the model generates wrong signals.
2. **Hyperparameter Insensitivity:** Unlike Finance-Llama-8B which shows dramatic swings (-12.47% to +12.26%), Phi-4 remains in the -52% to -59% range regardless of temperature or top-p. Config 3 (high exploration) achieves -54.15%, only marginally worse than Config 1 (conservative) at -52.44%. This invariance suggests hyperparameter tuning is irrelevant to Phi-4’s failure.

### Root Cause: Fundamental Domain Mismatch

Phi-4 is a general-purpose language model with no financial domain training. When presented with technical indicators (RSI, MACD, Bollinger Bands), the model attempts to reason from linguistic associations rather than learned financial concepts and we examined that in this project. For example:

- “RSI = 72” → Model treats as generic high number, not understanding overbought conditions
- “MACD negative” → Model associates “negative” with “bad”, generating sell signal regardless of context
- “Volume increasing” → Model may interpret as momentum without understanding volume-price relationships

This happened to us couple of times and with a lot of variations of prompt engineering we solve it. We tested Phi-4 with explicit domain definitions in prompts (“RSI is a momentum indicator ranging 0–100; values above 70 indicate overbought conditions”). This marginally improved performance but did not made it profitability, confirming that prompting alone cannot solve gaps of model weights.

### Why Hyperparameter Tuning Not Helps Performance:

We think this for Finance-LLM-13B, temperature increase enables exploration of valid alternative reasoning paths, improving performance. For Phi-4, temperature increase simply allows more exploration of *invalid* reasoning paths.

## 6.2.4 Comparative Summary: Model vs. Model

### Finance-LLM-13B vs. Finance-Llama-8B

In the optimal configurations:

Table 11: Finance-LLM-13B (Config 3) vs. Finance-Llama-8B (Config 2)

| Metric    | LLM-13B  | Llama-8B |
|-----------|----------|----------|
| Return    | +32.33%  | +12.26%  |
| vs Market | +41.37pp | +21.30pp |
| Advantage | +20.07pp |          |
| Win Rate  | 54.54%   | 33.33%   |
| Advantage | +21.21pp |          |
| Trades    | 22       | 21       |
| Long%     | 13.64%   | 19.05%   |

Finance-LLM-13B outperforms Finance-Llama-8B by 20.07 percentage points in return and 21.21 percentage points in win rate. Even among domain-trained models, the 13B architecture delivers superior performance. Making it the model we choose.

## 6.3 Discussion

This study has severe limitations that significantly constrain its validity and applicability:

**Single Historical Window:** All results derive from 10 days of BTC data in November 2025. Different market regimes can produce different signal patterns and model performance. A single bearish window cannot establish generalisability. We cannot claim the strategy works, only that it worked here.

**Incomplete Risk Analysis:** We report win rate and absolute return but omit critical metrics: Sharpe ratio, maximum drawdown, Calmar ratio, and conditional value-at-risk. A strategy returning +32% but experiencing 40% drawdown may be unsuitable for some investors.

**No Forward Testing:** All results are historical backtests data for re-productivity purposes. Forward testing on live market data is essential to validate whether strategies generalise.

**Sample Size:** 21 - 24 trades per configuration is small for statistical inference. Win rate confidence intervals are wide, reducing confidence in reported edges. Larger samples across longer time periods would strengthen claims.

**Strengths:** Controlled comparison isolating hyperparameter effects; systematic 4-configuration grid; multi-model comparison; realistic hardware constraints (consumer GPU, quantized models).

## 7 Conclusion and Future Work

The motivation was clear: LLMs demonstrate remarkable reasoning across domains, yet their application to technical trading analysis—which cryptocurrency markets demand due to 24/7 operation, extreme volatility, and sentiment-driven dynamics—remains largely unexplored. Cryptocurrency markets lack fundamental valuation anchors, forcing traders to rely on technical analysis and pattern recognition. The research gap was whether LLMs could match human-level technical analysis capability while running on consumer hardware. This research was about whether you use the power of Artificial Intelligence in this domain, in your house, without using any high-end equipment.

### 7.1 Answers to Research Questions

#### **RQ1: Can quantized LLMs generate statistically significant returns?**

Partially, but with conditions. Finance-LLM-13B achieved +32.33% return in backtest with 54.54% win rate, significantly outperforming a -9.04% market baseline (+41.37pp alpha). However, this result is from a single historical window; statistical significance cannot be claimed without multi-period validation. The confidence interval for the 54.54% win rate is [33.8%, 75.3%], excluding the 50% random threshold, suggesting a genuine edge in this window.

#### **RQ2: How do the three LLM architectures differ?**

Dramatically. Finance-LLM-13B is consistently profitable. Finance-Llama-8B is unstable and mostly unprofitable, with extreme directional bias that cannot be corrected via prompting or hyperparameter tuning. Phi-4 is categorically unsuitable, performing worse than random in all configurations. The performance hierarchy is strong across all four hyperparameter settings.

#### **RQ3: Does domain-specific fine-tuning outweigh parameter scaling?**

Our results show Finance-LLM-13B (13B, domain-trained) outperforming Phi-4 (14B, general-purpose), indicating that domain training can dominate parameter count in this

setting. This aligns with prior evidence that domain-specific pretraining outperforms generic models on domain tasks Beltagy et al. (2019). Even among domain-trained models, Finance-LLM-13B outperforms Finance-Llama-8B, suggesting both domain quality and scale matter within domain training.

**RQ4: How do temperature and top-p affect signal quality?**

In our experiments, higher temperature significantly improves Finance-LLM-13B performance. This aligns with self-consistency and chain-of-thought results showing that sampling diverse reasoning paths can improve reasoning quality Wei et al. (2022); Wang et al. (2022), suggesting well-trained models benefit from exploring diverse reasoning paths. Win rates improve from 47.61% to 57.14% at moderate temperature, indicating the model generates better trades, not merely more trades. However, this effect is model-specific: Finance-Llama-8B and Phi-4 show no improvement, confirming only robust models leverage temperature-driven exploration. Although nucleus sampling (top-p) is generally recommended for open-ended text generation Holtzman et al. (2019), our experiments with top-p = 0.9 show a 10.76pp degradation versus top-p = 0.5 at the same temperature, suggesting that in this financial-trading setting, temperature is more effective than aggressively relaxed top-p.

**RQ5: What hardware bottlenecks limit multi-agent LLM systems?**

Multi-agent systems face quadratic KV cache scaling: three concurrent Finance-LLM-13B instances require 30.3GB VRAM, this value is impossible to reach for a lot of consumer type hardware. We tried 3 solutions. First quantization to 2-bit, which reduces per-instance usage to 3.2 GB but in our tests degrades trading performance unacceptably, whereas 4-bit GPTQ-style quantization is known to preserve model quality much better Frantar et al. (2022). CPU offloading via system RAM introduces reload latency, viable but tight and smaller models (1.5–3B) fit within VRAM but sacrifice reasoning capacity. The optimal trade-off for us was RTX 3060 is a single agent (13B domain model) or two agents (8B + 8B) with careful memory management. True multi-agent systems demand higher-tier hardware (24GB+ VRAM).

## 7.2 Future Work

Future research should pursue:

**1. Multi-Regime Validation and Adaptive Systems**

Backtest on 5-10 different historical windows spanning bull trends, bear markets, high volatility, and sideways consolidation. For each regime, measure performance degradation. Investigate whether models can learn regime-specific adaptation via fine-tuning.

**2. Domain Transfer Efficiency**

Compare domain-pretrained models (Finance-LLM-13B) versus general-purpose models fine-tuned on financial data (LLaMA-2-13B + financial fine-tuning). Quantify the separate contributions of pretraining, fine-tuning, and their interaction to answer: Can aggressive fine-tuning save general models?

**3. Hybrid LLM-Classical ML Ensemble**

Testing a combination of LLMs’ reasoning, directional signals with LSTMs time-series prediction, and XGBoost position sizing. Each component can add some edge to this model.

**4. Forward Testing and Paper Trading**

Deploy Finance-LLM-13B in live paper trading for 90–180 days. Track profitability on unseen data.



### 7.3 Final Remarks

This work provides proof-of-concept evidence that quantized domain-specific LLMs can generate profitable trading signals in controlled backtests on **consumer hardware**. The results are very compelling. The research contributions are methodological (demonstrating the feasibility of LLM-based trading) and conceptual (showing that domain training dominates scale), but practical trading applicability remains unproven until tested further.

The most significant findings extend beyond trading: Our results, combined with prior evidence that domain-specific pretraining beats general models on domain tasks Beltagy et al. (2019), suggest that domain-specific training can matter more than raw parameter count in complex reasoning domains, and that well-trained models leverage stochastic exploration for improved reasoning Wei et al. (2022). These principles apply broadly across finance, medicine, law, and other complex reasoning domains.

Current validation is insufficient for commercialization or strong scientific claims. Multi-regime forward testing is the essential next step. Only after multiple validations across independent market regimes and periods can we responsibly claim LLM-based trading is viable.

The research gap identified in the question of whether LLMs can interpret technical analysis and generate profitable signals on consumer hardware has an answer: yes. Whether this translates to a robust, deployable trading edge remains an open question requiring future validation.

## References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Alon, F. et al. (2023). Gpt-4 technical report, *arXiv preprint arXiv:2303.08774* .  
**URL:** [10.48550/arXiv.2303.08774](https://arxiv.org/abs/2303.08774)
- Araci, D. (2019). Finbert: Financial domain-specific language model for financial sentiment analysis, *arXiv preprint arXiv:1908.10063* .  
**URL:** [10.48550/arXiv.1908.10063](https://arxiv.org/abs/1908.10063)
- Balcilar, M., Bein, D. and Levkoff, N. (2017). Can volume predict bitcoin returns and volatility? a quantiles-based approach, *Economic Modelling* **64**: 74–81.  
**URL:** [10.1016/j.econmod.2017.03.019](https://doi.org/10.1016/j.econmod.2017.03.019)
- Beltagy, I., Lo, K. and Cohan, A. (2019). Scibert: A pretrained language model for scientific text, *arXiv preprint arXiv:1903.10676* .  
**URL:** [10.48550/arXiv.1903.10676](https://arxiv.org/abs/1903.10676)
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. (2020). Language models are few-shot learners, *Advances in Neural Information Processing Systems* **33**: 1877–1901.  
**URL:** [10.48550/arXiv.2005.14165](https://arxiv.org/abs/2005.14165)
- Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* .  
**URL:** [10.48550/arXiv.1810.04805](https://arxiv.org/abs/1810.04805)
- Frantar, E., Ashkboos, S., Hoeffler, T. and Alistarh, D. (2022). Gptq: Accurate post-training quantization for generative pre-trained transformers, *arXiv preprint arXiv:2210.17323* .  
**URL:** [10.48550/arXiv.2210.17323](https://arxiv.org/abs/2210.17323)
- Holtzman, A., Buys, J., Du, L., Forbes, M. and Choi, Y. (2019). The curious case of neural text degeneration, *International Conference on Learning Representations* .  
**URL:** [10.48550/arXiv.1904.09751](https://arxiv.org/abs/1904.09751)
- Hubara, I., Courbariaux, M., Soudry, D. and Bengio, Y. (2016). Quantization for neural networks with low precision weights and activations, *arXiv preprint arXiv:1609.07061* .  
**URL:** [10.48550/arXiv.1609.07061](https://arxiv.org/abs/1609.07061)
- Kirtac, K. (2024). Sentiment trading with large language models, *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA)*, Association for Computational Linguistics.  
**URL:** <https://aclanthology.org/2024.wassa-1.1/>
- Lopez-Lira, A. (2024). Can large language models trade? testing financial theories with llm agents in market simulations, *arXiv preprint arXiv:2504.10789* .  
**URL:** [10.48550/arXiv.2504.10789](https://arxiv.org/abs/2504.10789)

- Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M. et al. (2022). Bloom: A 176b-parameter open-access multi-lingual transformer, *arXiv preprint arXiv:2211.05100* .  
**URL:** [10.48550/arXiv.2211.05100](https://arxiv.org/abs/2211.05100)
- Schwettmann, S., Frankle, J., Engstrom, L., Hendrycks, D. and Wallace, E. (2024). Does prompt formatting have any impact on llm performance?, *arXiv preprint arXiv:2411.10541* .  
**URL:** [10.48550/arXiv.2411.10541](https://arxiv.org/abs/2411.10541)
- Urquhart, A. (2016). The inefficiency of bitcoin, *Economics Letters* **148**: 80–82.  
**URL:** [10.1016/j.econlet.2016.09.019](https://www.econlet.com/article/S0165-0513(16)30019-1)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017). Attention is all you need, *Advances in Neural Information Processing Systems* **30**: 5998–6008.  
**URL:** [10.48550/arXiv.1706.03762](https://arxiv.org/abs/1706.03762)
- Wang, D. et al. (2025). Slot: Structuring the output of large language models, *Proceedings of EMNLP 2025 (Industry Track)*.  
**URL:** <https://arxiv.org/abs/2505.04016>
- Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Komatsuzaki, S. and Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models, *arXiv preprint arXiv:2203.11171* .  
**URL:** [10.48550/arXiv.2203.11171](https://arxiv.org/abs/2203.11171)
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichien, B., Xia, F., Chi, E. H., Le, Q. V. and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models, *arXiv preprint arXiv:2201.11903* .  
**URL:** [10.48550/arXiv.2201.11903](https://arxiv.org/abs/2201.11903)
- Zhang, C., Li, S., Du, M. and Zhang, Y. (2024). When ai meets finance (stockagent): Large language model-based multi-agent framework for stock trading simulation, *arXiv preprint arXiv:2407.18957* .  
**URL:** [10.48550/arXiv.2407.18957](https://arxiv.org/abs/2407.18957)
- Zhang, K., Yoshie, O., Sun, L. and Huang, W. (2024). Finllm-b: When large language models meet financial breakout trading, *arXiv preprint arXiv:2402.07536* .  
**URL:** [10.48550/arXiv.2402.07536](https://arxiv.org/abs/2402.07536)
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q. (2020). A comprehensive survey on transfer learning, *Proceedings of the IEEE* **109**(1): 43–76.  
**URL:** [10.1109/JPROC.2020.3004555](https://ieeexplore.ieee.org/abstract/document/9186238)