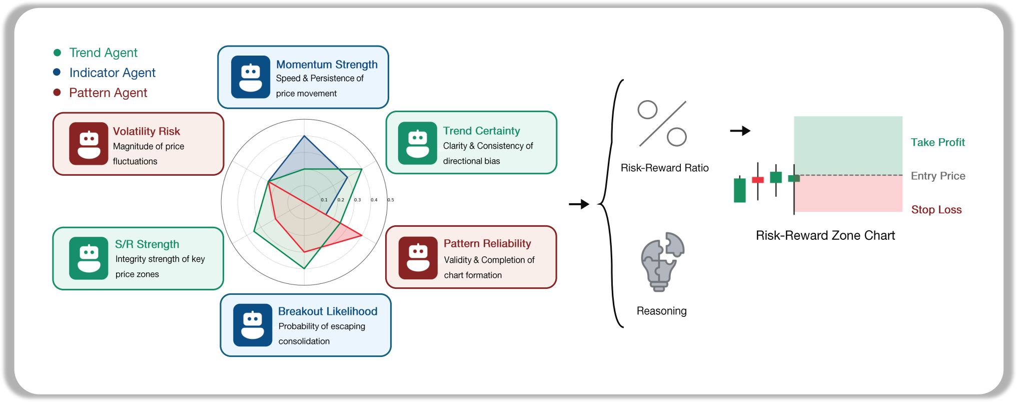

Trend Agent
Indicator Agent
Pattern Agent
Momentum Strength
Speed & Persistence of price movement
Volatility Risk
Magnitude of price fluctuations
Trend Certainty
Clarity & Consistency of directional bias
S/R Strength
Integrity strength of key price zones
Pattern Reliability
Validity & Completion of chart formation
Breakout Likelihood
Probability of escaping consolidation
Risk-Reward Ratio
Reasoning
Take Profit
Entry Price
Stop Loss
Risk-Reward Zone Chart

Figure 2 Workflow of RiskAgent. Signals from Trend, Indicator, and Pattern are aggregated into a radar chart. DecisionAgent predicts with stop-loss and take-profit.

3.1 IndicatorAgent

IndicatorAgent constitutes the initial analytical module of our framework, responsible for transforming raw OHLC sequences into structured quantitative signals (Figure 1). In high-frequency trading (HFT), where decisions must be executed under strict latency constraints, technical indicators provide compact representations that highlight shifts in market momentum and sentiment. Formally, this process can be viewed as a mapping from price tuples to an interpretable signal space, $(O, H, L, C) \mapsto \mathcal{S}$, where $\mathcal{S}$ denotes a set of derived features summarizing short-horizon market dynamics (Lo et al., 2000). By abstracting low-level price data into high-level features, IndicatorAgent facilitates fast and interpretable downstream reasoning.

To achieve this, it converts raw OHLC values into a compact set of informative technical indicators. Specifically, IndicatorAgent uses five widely used technical indicators to extract signals from market data. RSI captures momentum and flags overbought or oversold zones (Wilder, 1978), while MACD tracks convergence or divergence between short and long term price trends (Appel, 2005). RoC measures the speed of price changes (Murphy, 1999), STOCH identifies turning points based on recent highs and lows (Investopedia, n.d.), and WILLR detects price drops from recent peaks to signal possible reversals (Williams, 2011).

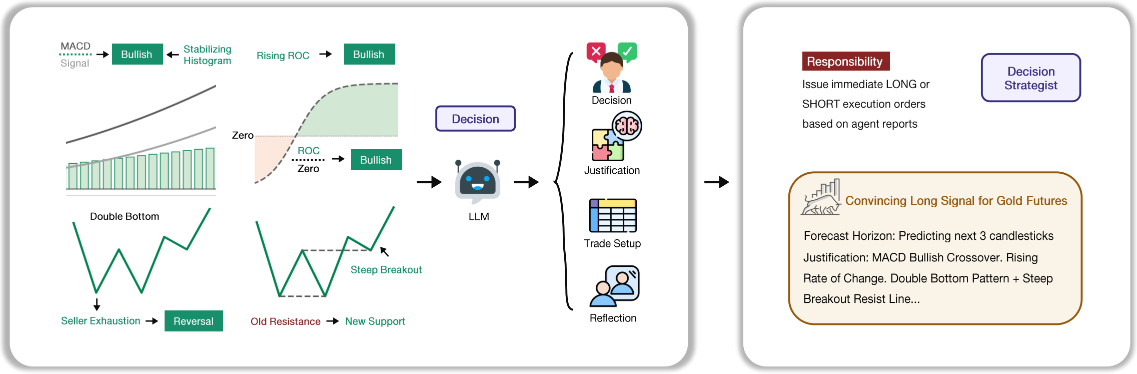
Together, these indicators capture both short-term volatility and longer-term momentum. IndicatorAgent integrates them into a structured summary that reflects current market conditions. Rather than applying fixed rules, it interprets signals in context such as highlighting dynamics like momentum shifts, overbought or oversold zones, and sudden reversals (Murphy, 1999). This contextual reasoning enables DecisionAgent to act on timely, relevant insights, improving responsiveness in fast-moving trading environments (Lo et al., 2000).

3.1.1 PatternAgent Design

While IndicatorAgent offers useful signals, these numerical indicators can become unclear—especially when price movement stalls or enters a new regime (Murphy, 1999; Lo et al., 2000). To address these limitations, our PatternAgent introduces a more visual and structural multi-modal reasoning perspective (Figure 1).

Upon receiving a request to analyze market patterns, PatternAgent first utilizes LLM-binded tools to generate clear, simplified candlestick charts directly from raw price data. These visualizations present recent market behavior without explicitly highlighting specific geometric shapes or details. Our agent, instead, automatically detects essential visual features from price movements, such as significant highs and lows, structural symmetry, and potential reversal formations,effectively capturing key visual patterns used in technical analysis (Wang et al., 2023).

Using this information, PatternAgent compares the current market structure to an extensive library of well-known patterns described in clear language. Each pattern in this library includes concise yet detailed descriptions of its visual form and the market behaviors it typically signals. Through this comparison, PatternAgent identifies the most plausible match and evaluates its relevance to the current context. This process blends visual



Issue immediate LONG or SHORT execution orders based on agent reports

Forecast Horizon: Predicting next 3 candlesticks
Justification: MACD Bullish Crossover. Rising Rate of Change. Double Bottom Pattern + Steep Breakout Resist Line...

Figure 3 Workflow of DecisionAgent. LLM summarizes upstream signals and consolidates them into structured outputs: direction, reasoning, trade setup, and post-trade reflection. It then formulates an executable order with rationale, ready for market submission.

understanding with language-based reasoning to recognize patterns and assess whether their context—such as preceding trends or volatility—makes them likely to signal a meaningful price move.

By translating complex visual signals into concise and interpretable summaries, PatternAgent plays a key role in bridging raw chart data and high-level reasoning, allowing the system to integrate visual structure into its overall market understanding (Lo et al., 2000).

3.1.2 TrendAgent Design

Canonical chart patterns, such as double bottoms or flags, can be reliably interpreted only when evaluated within the context of a well-defined trend (Pring, 1991). By tracking both the direction and steepness of price movements over time, TrendAgent provides a structural representation of trend dynamics, clarifying whether a detected pattern is consistent with the prevailing trend, signals a potential reversal, or indicates a phase of non-directional price congestion (Elder, 2002).

As shown in Figure 1, TrendAgent generates an annotated K-line chart $\mathcal{K}_t$, which includes a trend channel $\mathcal{C}_t$ that captures the price trajectory through two fitted lines: the upper resistance line $\mathcal{R}_t(x) = m_r x + b_r$, drawn through recent local highs, and the lower support line $\mathcal{S}_t(x) = m_s x + b_s$, drawn through recent local lows (Lo et al., 2000). The trendlines, computed using ordinary least squares regression as outlined in Algorithm 1, serve to characterize price direction, strength, and potential breakout zones. The average slope $\kappa_t = \frac{m_r + m_s}{2}$ provides a basic estimate of directional drift. However, reliable trend classification requires more than just the slope sign, as market noise can obscure short-term movements.

To address that, TrendAgent examines the geometric relationship between the lines—such as parallel upward slopes indicating strong trends, or converging lines forming a wedge that suggests indecision or accumulation. These structural cues allow the agent to reason about not just direction but also the confidence and stability of the trend. They work in conjunction with shape-based patterns and momentum signals identified by other agents, improving decision-making and reinforcing signal coherence across the system (Kirkpatrick and Dahlquist, 2015).

3.1.3 RiskAgent and DecisionAgent

RiskAgent, shown in Figure 2, translates technical insights into risk-aware trade boundaries, reflecting the central role of risk control in real-world trading. Instead of signal generation, RiskAgent integrates other agents' output into a unified risk-reward framework. It sets a fixed stop-loss value $\rho = 0.0005$ to account for short-term volatility and computes a take-profit level $\mathcal{R} = r \cdot \rho$, where $r \in [1.2, 1.8]$ is predicted by the LLM. This forms a structured decision zone bounded by stop-loss, entry, and take-profit levels. Within this zone, the agent reasons over signal quality and predefined risk exposure to ensure consistent and informed actions. By aligning domain knowledge with real-time constraints, RiskAgent grounds high-level analysis in practical execution under uncertain and fast-moving market environment.

The final stage of the framework is DecisionAgent, which functions as the reasoning and execution layer. As illustrated in Figure 3, it integrates the outputs of upstream agents to decide between a LONG or SHORT position. Since holding is not permitted, the agent forecasts short-term market direction over the next three candlesticks and generates actionable decisions aligned with the aggregated signals from the overall system (Kissell, 2013).

Specifically, DecisionAgent takes in an aggregated signal state from IndicatorAgent, PatternAgent, and TrendAgent, and outputs a structured trading decision comprising the predicted direction (LONG or SHORT), a concise justification, and a risk–reward ratio conditioned on market context (Kissell, 2013). The agent integrates heterogeneous evidence and evaluates the consistency across upstream signals, proceeding only when the majority align and are reinforced by confirmations such as indicator crossovers, completed breakout formations, or price interactions with major trend boundaries. This layered reasoning filters out noisy or conflicting inputs and yields confident, high-quality decisions. Consequently, the outputs are not only optimized for execution in high-frequency settings but also more robust and interpretable than those produced by traditional rule-based systems.

4 Experiments

We evaluate our QuantAgent framework in a fair and comprehensive manner, with trading decisions generated autonomously without prior demonstrations or supervised fine-tuning. Building on the structured reasoning provided by upstream agents, our system uses only recent candlestick data and basic context (e.g., asset type and time interval) to predict short-term market direction. It then generates clear trade suggestions with human-readable explanations, allowing us to evaluate its performance in realistic settings. The experiment is designed to test the framework's effectiveness in realistic, data-limited environments where fast, adaptive decision-making is required.

Benchmark Construction and Evaluation Protocol. To support evaluation, we build benchmarks of 4-hour and 1-hour OHLC data across key asset classes such as cryptocurrencies, equity indices, and commodities. For each asset, 5000 historical bars are collected via a public TradingView data extraction tool API. From this, 100 evaluation segments per asset are sampled, each with 100 consecutive candlesticks—the last three withheld to prevent test-time leakage. Details of benchmark are illustrated in Appendix E.

The system processes agents' analysis to generate a structured trade report containing a directional prediction (LONG or SHORT), a brief textual rationale, and an estimated risk–reward ratio. Among these outputs, the directional decision and risk–reward estimate are used for quantitative evaluation.

Baselines. We evaluate four categories of baselines: **(i) Random Methods**, which performs random selection of market trend between LONG and SHORT and risk-reward ratio $\in [1.2, 1.8]$. **(ii) Linear Regression**, which serves as a rule-based baseline. It fits a linear model to a 40-bar window of recent closing prices and use the slope of the fitted line to classify future trend. If the slope exceeds 0, the system predicts LONG. Otherwise, it predicts SHORT. **(iii) XGBoost**, which serves as a tree-based supervised learning model. It uses technical indicators extracted via a public API TA-Lib (TA-Lib Development Team, 2025) including RSI, MACD, and SMA. This model is trained on hundreds of sliding-window sample across 50 randomly selected csv files. The trained model is tested on the rest 50 csv files with the same metrics as other methods. A majority-vote rule is applied across predictions to produce a final LONG/SHORT/HOLD decision where HOLD decision is disregarded during final average calculation. **(iv) QuantAgent**, our LLM-based approach, perform short-term prediction using multi-modal multi-agent cooperation.

Evaluation Metrics. To evaluate prediction accuracy, we compare the LLM's directional forecast to the next three candlesticks. For a LONG decision, each candle that closes above the last close counts as a correct hit (max 3); for SHORT, each close below the current close counts. Accuracy is defined as $\alpha = \frac{\mathbb{C}}{\mathcal{T}}$, where $\mathbb{C}$ is the number of correct predictions and $\mathcal{T}$ is the total evaluated. Each test yields a score from 0 to 3, and the average is computed over all samples. This aligns with the Mean Directional Accuracy metric used in forecasting (Pesaran and Timmermann, 2004).

In addition, we evaluate trade outcomes based on multiple Rate of Return (RoR) estimators (Fama and MacBeth, 1973) commonly used in HFT, each is to quantify the profitability of a trade by measuring the relative gain or loss between the entry price and exit. All rate-of-return metrics in our framework, including