©Copyright 2026
Dazhi Li

# Towards Smarter Trading: An AI Trading Framework Combining Reinforcement Learning and Large Language Model

Dazhi Li

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science In Computer Science & Software Engineering

University of Washington

2026

Reading Committee:
Dr. Min Chen, Chair
Dr. Munehiro Fukuda, member
Dr. Dong Si, member

Program Authorized to Offer Degree:
Department of Computing and Software Systems

University of Washington

**Abstract**

Towards Smarter Trading: An AI Trading Framework Combining Reinforcement Learning and Large Language Model

Dazhi Li

Chair of the Supervisory Committee:
Dr. Min Chen
Department of Computing and Software Systems

The rapid evolution of financial markets demands intelligent trading systems capable of synthesizing heterogeneous information and making adaptive decisions under conditions of uncertainty. In this paper, we propose a trading framework that leverages reinforcement learning (RL) to fine-tune a Large Language Model (LLM) for autonomous trade decision-making. Unlike prior approaches that depend on supervised pre-training with expert-annotated analyses or domain-specific corpora for cold-start guidance, our method applies Group Relative Policy Optimization (GRPO) directly to a general-purpose instruction-tuned LLM, enabling the model to develop trading competence purely through reward-driven exploration, without curated professional signals.

The model ingests multi-source market observations — encompassing technical indicators, financial news, and corporate financial statements — within a rolling temporal window and outputs structured trading strategies specifying action type, share quantity, take-profit price, and stop-loss price. This formulation enforces strategy completeness through explicit exit conditions while also supporting flexible position sizing; bridging the gap between simplified academic models and practical trade execution.

To guide learning, we design a multi-dimensional reward function grounded in profitability and trading discipline. Each strategy is evaluated on path-dependent profit-and-loss, risk

exposure relative to stop-loss levels, position sizing appropriateness, and regulatory adherence; overall, this provides fine-grained feedback that cultivates the model's awareness of both return potential and downside risk.

We conduct comprehensive experiments along three dimensions: (1) model comparisons, contrasting the RL-trained LLM against its base model, alternative LLM architectures, and a DQN-based traditional RL trading agent to quantify improvements from RL fine-tuning and LLM-based reasoning respectively; (2) training budget analyses, investigating how training steps influence the model; and (3) strategy ablations, examining the contributions of quantity-based position sizing. Our results demonstrate that the proposed framework produces coherent, risk-aware trading strategies without supervised warm-up; the ablation analyses further yield insights into the respective roles of model capacity, training sufficiency, and strategy design.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# GLOSSARY

A2C: Advantage Actor–Critic, an on-policy deep reinforcement-learning algorithm with paired policy and value networks.

ACTION SPACE: The set of decisions the agent may emit. In this thesis the action space is a structured JSON object specifying ticker, action (buy/sell/hold), quantity, current price, take-profit, and stop-loss.

BUY & HOLD: A passive benchmark that allocates the initial capital evenly across the ticker universe at the start of the evaluation window and holds without further trading.

DECISION DAY: The last trading day of an observation window; the day on which the model is asked to choose an action.

DQN: Deep Q-Network, a value-based deep RL algorithm that approximates an action-value function with a neural network. Used as the non-LLM baseline in RQ1.

DUST TRADE: An economically negligible trade — a buy whose notional is below 2% of account value, or a sell smaller than 10% of the held position. Dust trades pass schema validation but are penalised by the regulation and correctness rewards.

EMA-$N$: Exponential moving average of the closing price over an $n$-day window. EMA-7 and EMA-14 are used here as trend indicators.

FMP: Financial Modeling Prep, the public financial-data API used here for news, balance sheets, and income statements.

GRPO: Group Relative Policy Optimisation, the PPO variant used here for fine-tuning. Replaces the learned value network with an on-the-fly group baseline computed from $K$ parallel rollouts of the same prompt.

LORA: Low-Rank Adaptation. Freezes the base weights and inserts trainable low-rank matrices into selected linear layers, reducing the trainable parameter count to under 1% of the full model.

MACD: Moving Average Convergence Divergence; a momentum indicator equal to the difference of two exponential moving averages of the price.

OBSERVATION WINDOW: A rolling $T = 7$ trading-day slice of OHLCV and indicator history that, combined with the agent's portfolio and account state, forms one decision-making context.

OHLCV: Open, high, low, close, volume; the daily price and volume record retrieved from the data provider.

PPO: Proximal Policy Optimisation; an on-policy deep RL algorithm that maximises a clipped surrogate objective. GRPO is its group-baseline variant.

RSI: Relative Strength Index; a 0–100 momentum indicator that flags overbought and oversold conditions.

SFT: Supervised fine-tuning. Teaching a base model to imitate human-curated outputs. The framework here deliberately omits the SFT step used by prior LLM-trading systems.

STOP-LOSS: A self-specified price level below the entry; if the market crosses it, the position is closed to cap downside.

SWEET SPOT: The buy notional range 5%–15% of the account (or sell range 25%–50% of the held position) at which the regulation reward and the correctness conviction multiplier both peak.

SWING TRADING: A short-to-medium-term trading style holding positions for several days; the horizon implicit throughout this thesis.

TAKE-PROFIT: A self-specified price level above the entry; if the market reaches it, the position is closed to lock in gains.

VLLM: A high-throughput inference engine for LLMs; disabled on the 24 GB development GPU and enabled on the cloud A100-80 GB used for the final reported runs.

# ACKNOWLEDGMENTS

I am deeply grateful to my advisor, Professor Min Chen, whose guidance shaped every part of this thesis. From the earliest framing of the research questions, through the design and implementation of the framework, to the empirical study that closes the work, her insight, patience, and concrete suggestions on experimental direction were the steady hand behind every meaningful turn this project took.

I also wish to thank Craig Rainey, whose work on reinforcement-learning-based algorithmic trading was my entry point into RL and remained a recurring reference throughout this thesis.

Finally, I am grateful to my thesis committee for their feedback and for helping me plan the workflow and milestone checks that kept this project on track. The structure they helped impose turned what might have been an open-ended exploration into a thesis I am proud to submit.

# Chapter 1

# INTRODUCTION

## *1.1 Problem Statement*

Financial markets are complex dynamic systems driven by macroeconomic indicators, corporate fundamentals, investor sentiment, and geopolitical events. Profitable trading requires the rapid synthesis of heterogeneous information, from numerical price data and technical indicators to unstructured text such as financial news and earnings reports, at a volume that has outpaced what manual analysis can sustain, motivating decades of research into automated trading systems.

Classical rule-based and statistical-arbitrage strategies have succeeded in production but rely on hand-crafted features and fixed decision rules that adapt poorly to shifting market regimes. Reinforcement learning (RL) addresses this by letting an agent learn policies directly from market interaction through trial-and-error optimisation of cumulative reward [8], and deep RL architectures such as DQN [4], PPO, and A2C have been widely applied to portfolio management and trade execution, with FinRL [2] standardising the paradigm. Yet recent empirical work continues to find that standalone deep RL trails well-tuned algorithmic baselines in realistic settings: Rainey [6], for example, reports that a carefully tuned DQN agent fails to consistently outperform conventional baselines, pointing to the representational and reasoning limits of traditional RL architectures.

In parallel, Large Language Models (LLMs) have shown strong capabilities in understanding, reasoning, and structured output generation. Furthermore, their ability to ingest unstructured text e.g., financial news, analyst reports, and regulatory filings, makes them a natural foundation for systems that must integrate quantitative and qualitative signals. Combining LLM reasoning with reward-driven RL, typically through algorithms such

as Group Relative Policy Optimization (GRPO) [7], is a particularly promising direction. Trading-R1 [10] is the most direct recent instantiation, but its pipeline depends on a supervised fine-tuning (SFT) warm-up over a curated corpus of professional trading analyses; on enterprise-grade financial data; and on a simplified action space of discrete sentiment-level recommendations (e.g. "buy," "strong buy"), rather than executable trade parameters. These requirements make it difficult to attribute the reported performance to the RL component itself rather than to the expert-curated supervised signal that precedes it.

A further limitation cuts across both lines of work: existing frameworks typically output only a directional signal (buy, sell, hold), with no position size, take-profit, or stop-loss. In practice a trading decision without defined exit conditions is incomplete, since risk management is not optional [1]. The open gap, therefore, is an accessible end-to-end RL framework that can fine-tune a general-purpose LLM for trading without supervised warm-up or expert-curated corpora, and that produces complete, executable strategies including sizing and explicit risk controls. This work targets independent researchers and retail-scale traders who need a reproducible, risk-aware LLM trading framework without proprietary data or curated analyst corpora.

### *1.2 Goals*

The overarching goal of this project is to design, implement, and empirically evaluate an end-to-end reinforcement learning framework that fine-tunes a general-purpose LLM into a competent trading agent, producing complete, risk-aware strategies directly from multi-source market observations — without supervised pre-training, expert-curated corpora, or proprietary data infrastructure. The project pursues five specific objectives:

- Objective 1 — RL fine-tuning pipeline. Build a GRPO-based fine-tuning pipeline that adapts an off-the-shelf instruction-tuned LLM into a trading agent under purely reward-driven signals, with no supervised warm-up.

- Objective 2 — Heterogeneous market observation construction. Aggregate technical

indicators, financial news, and corporate financial statements into rolling observation windows from publicly accessible sources only.

- Objective 3 — Complete, risk-aware action formulation. Define the action space as a structured trading strategy that specifies direction, position size, take-profit, and stop-loss, making risk management a first-class component of every decision.

- Objective 4 — Multi-dimensional reward function. Score each generated strategy along path-dependent profit-and-loss, stop-loss-relative risk, sizing appropriateness, and trading-rule compliance, jointly optimising for return and downside control.

- Objective 5 — Comprehensive empirical evaluation. Evaluate the framework along three independent dimensions — model, training, and strategy — to quantify the contribution of each design choice and to position it relative to existing baselines.

The evaluation component (Objective 5) is guided by three research questions:

- RQ1 (Model): Does RL fine-tuning measurably improve an LLM's trading performance over its own base model, over an alternative LLM architecture and over a traditional DQN trading agent?

- RQ2 (Training budget): On a fixed training corpus, does doubling the GRPO step budget from roughly half-epoch to full-epoch coverage improve out-of-sample performance, or have the shaping rewards already saturated?

- RQ3 (Strategy): How does explicit per-trade quantity in the action space shape the overall performance of the trained agent?

### *1.3 Report Structure*

The remainder of this thesis is organised into four chapters. Chapter 2 reviews RL for trading, LLMs in finance, and the recent fusion of the two, and positions this work against the

limitations identified there. Chapter 3 presents the proposed framework: the observation pipeline, the structured action space, the multi-dimensional reward function, the two-stage GRPO fine-tuning procedure, and the evaluation methodology. Chapter 4 reports experiments along the model, training-budget, and strategy dimensions defined in Section 1.2, answering each research question in turn. Chapter 5 summarises the findings, discusses their implications and limitations, and outlines directions for future work.

# Chapter 2

# RELATED WORK

## *2.1 Reinforcement Learning for Financial Decision-Making*

Reinforcement learning has been extensively studied as a framework for automated trading, due to its natural correspondence with the sequential, reward-driven nature of market decision-making. Deep RL architectures, most notably DQN [4], PPO, and A2C, have been applied to portfolio management, trade execution, and signal generation. A representative example is the ensemble strategy of Yang et al. [12], which dynamically selects among PPO, A2C, and DDPG agents by Sharpe ratio and outperforms both any individual agent and the Dow Jones benchmark on risk-adjusted return. Such work illustrates a broader pattern: improvements are typically pursued through algorithmic ensembling or architectural refinement, rather than through richer decision representations. Open-source frameworks such as FinRL [2] have standardised this paradigm and made deep RL for trading widely reproducible.

More recent empirical work questions the practical ceiling of the neural network. Rainey [6] reports that, despite careful tuning and feature engineering, a DQN-based trader cannot consistently outperform conventional algorithmic baselines and attributes the gap to the limited representational capacity of the network and the inability of traditional RL agents to reason about broader financial context. This is a recurring limitation: deep RL traders emit actions as opaque policy decisions, with no accompanying rationale and no awareness of the qualitative information (e.g. news, fundamentals, regulatory context) that human traders routinely use, which constrains both trustworthiness and cross-regime generalisation.

### *2.2 Large Language Models in Finance and the Rise of RL-Enhanced Reasoning*

LLMs open a complementary direction for financial AI, grounded in natural-language understanding and reasoning rather than purely numerical policy optimisation. BloombergGPT [9], a 50B-parameter model trained on a large proprietary financial corpus, has demonstrated that domain-specialised LLMs substantially outperform general-purpose ones on financial NLP tasks such as sentiment analysis, NER, and QA. It remains, however, a text-understanding system, not a trading decision-maker.

FinGPT [11] pursues the same goal from a different design point. Rather than training a single 50B-parameter model from scratch on a closed 363B-token corpus, FinGPT releases an open-source, instruction-tuned stack that adapts publicly available LLM backbones to publicly available financial datasets, and reports improved sentiment-task performance over BloombergGPT under a far smaller compute budget. The openness and the performance gain are both significant for reproducibility, but FinGPT — like BloombergGPT — targets financial language *understanding*: it produces sentiment scores and analyst-style commentary, not executable trading decisions.

A second turning point is the rise of reward-driven reasoning enhancement. Group Relative Policy Optimization (GRPO) [7], introduced with the DeepSeekMath line of work, lets an LLM improve its reasoning through RL without a separately trained value model; it has proven effective across mathematical, logical, and structured-reasoning tasks, naturally motivating its application to financial decision-making, where multi-factor reasoning over heterogeneous inputs is equally essential. Trading-R1 [10] is the most direct application to trading. This three-stage pipeline of SFT on professional trading analyses, reasoning-oriented fine-tuning, and GRPO produces an LLM that emits trading recommendations with explicit reasoning. The reported results show that combining LLM reasoning with RL meaningfully improves over non-reasoning baselines.

A closer parallel to the present framework is the recent work of Long et al. [3], who pair

LLM-derived sentiment signals with technical indicators in a reinforcement-learning trading framework over 44 S&P 500 stocks. Both frameworks share one design ingredient — an external LLM is used to compress raw financial news before downstream consumption — but the role of the LLM diverges sharply thereafter. In their pipeline, FinGPT sits outside the trading agent as a fixed sentiment-feature extractor, and a separately trained Twin Delayed DDPG (TD3) network consumes the resulting sentiment-plus-technical vector and emits the trading action; the LLM itself never participates in the decision, which places the approach closer to a traditional neural-network RL pipeline equipped with LLM-derived input features. In ours, the LLM *is* the trading agent — the Llama model that reads the observation also emits the decision, and it is the parameters of that LLM that the RL phase updates.

This architectural difference propagates to two further distinctions. The training procedures operate on different objects: our GRPO update modifies the parameters of the LLM directly, whereas TD3 updates the weights of a standalone MLP actor–critic pair sitting on top of fixed LLM features. And the research questions differ accordingly: their study focuses on whether LLM-derived sentiment signals add value when combined with traditional technical indicators in an RL setup, whereas ours focuses on whether RL fine-tuning can equip an LLM to make trading decisions in the first place — with news, indicators, and financial statements all serving as input modalities to the policy rather than as the subject of the investigation.

Three structural limitations nevertheless persist across this group. First, both BloombergGPT and Trading-R1 depend on prerequisites that are neither public nor easily reproducible (a 363B-token proprietary corpus in the former, a curated set of professional analyses for SFT/RFT warm-up in the latter); this makes it impossible to isolate the contribution of the RL component from that of the expert-curated supervised signal that precedes it. Second, the action space these systems consider is highly simplified: Trading-R1, for example, emits only discrete sentiment-level labels such as "buy" or "strong buy," with no position size, take-profit, or stop-loss — which means the policy never has to reason about how much to

deploy or under what condition to exit. Third, the training reward in Trading-R1 is a single classification-style signal that grades agreement with an analyst-derived target label, and so cannot register failure modes orthogonal to directional correctness (malformed JSON, hallucinated price references, dust-sized orders, over-budget buys, late stop-loss triggers). Our multi-component reward is designed precisely to expose these orthogonal failure modes during training, with format, length, factual grounding, and regulation each contributing a bounded scalar alongside the path-dependent correctness signal.

### *2.3 Research Gap and Positioning of This Thesis*

Two research gaps therefore remain open. First, it is unclear whether an LLM can become a competent trading agent through pure RL, with no supervised warm-up, no expert-curated reasoning corpus, and only publicly available data, because existing LLM-based trading systems entangle SFT and RL and cannot separate the contribution of each. Second, no prior LLM-based trading framework produces complete, execution-ready strategies that include size and exit conditions [1].

This thesis addresses both of these gaps. It proposes a two-stage, pure-RL fine-tuning framework over a general-purpose instruction-tuned LLM, using only publicly accessible technical indicators, financial news, and financial statements, and trains the model to emit structured strategies specifying action, size, take-profit, and stop-loss. Both stages are driven entirely by reward signals: Stage 1 shapes output format, length, factual grounding, and regulatory compliance; Stage 2 then targets profit-and-risk correctness on top of a stable output distribution. Within this framework, the thesis further studies how base-model choice, training budget, and action-space design each shape the resulting trading performance.

# Chapter 3

# MULTI-REWARD TWO-STAGE GRPO FINE-TUNING FOR CLOSED-LOOP LLM TRADING

## *3.1 Framework Overview*

The proposed framework is an end-to-end pipeline that turns publicly accessible market data into a fine-tuned LLM that generates complete, risk-aware trading strategies. It is organised into four stages — observation construction, task formulation and reward design, two-stage GRPO fine-tuning, and sequential evaluation — and is shown in Figure 3.1.

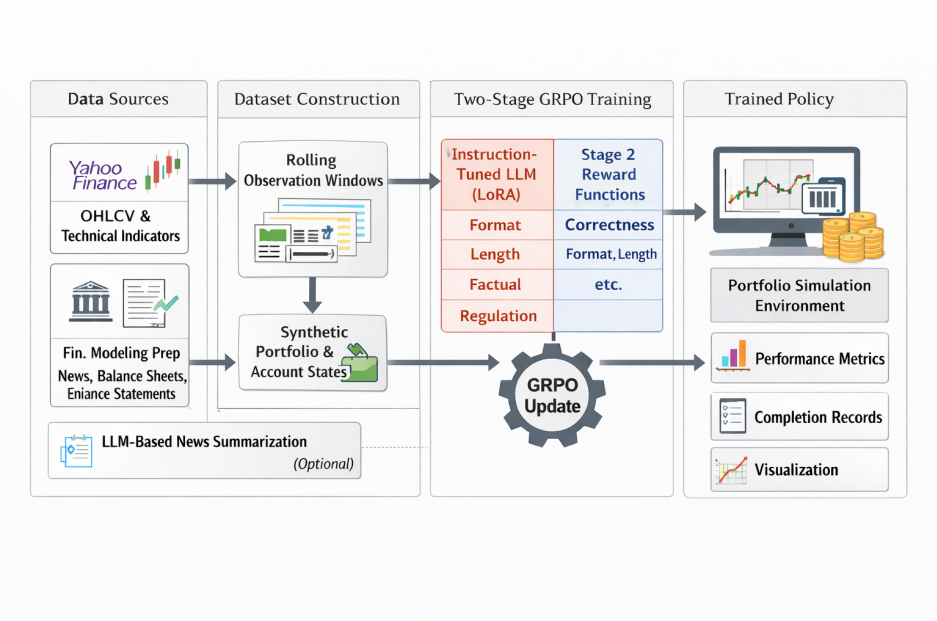


Figure 3.1: End-to-end architecture of the proposed reinforcement-learning-based LLM trading framework.

**Stage 1 — Input assembly.** Numerical price series and technical indicators (MACD,

RSI, EMA-7, EMA-14) are collected from Yahoo Finance; financial news, balance sheets, and income statements from the Financial Modeling Prep API. An optional news-summarisation step shortens the input and stabilises its length.

**Stage 2 — Task and reward.** Market signals are organised into rolling observation windows over a configurable look-back horizon and paired with a synthetically sampled portfolio and account state (Section 3.2.2). A fixed system prompt declares the agent's role, the JSON output schema, and the trading rules; together with the schema it forms the contract between the model and the environment (Section 3.3). Each generated strategy is then scored by a multi-dimensional reward function combining profitability and trading discipline (Section 3.4).

**Stage 3 — Two-stage GRPO fine-tuning.** Rather than applying a single monolithic reward, training is split into two curriculum stages. Stage 1 drives the model with four equally weighted output-shaping rewards (format, completion length, factual grounding, regulation), establishing a structurally valid output distribution. Stage 2 then promotes the path-dependent correctness reward to the dominant signal while keeping the four shaping rewards as low-weight regularisers. Both stages are purely reward-driven, with no supervised imitation step, distinguishing the framework from prior LLM-trading systems that rely on SFT warm-up. The base model is Llama-3.2-3B-Instruct with a LoRA adapter, keeping the pipeline reproducible on commodity hardware (Section 3.5).

**Stage 4 — Sequential evaluation.** Whereas training scores each prompt independently against a synthetic portfolio context, evaluation chains states sequentially: the cash and portfolio at step $t{+}1$ are the exact outcome of the policy's action at step $t$, producing a single realised PnL trajectory. The evaluation pipeline logs every completion (including malformed ones) and visualises reward dynamics across both stages; the rationale for this asymmetry and the design of the model–data–strategy comparison are developed in Section 3.6.

## *3.2 Dataset Design*

### *3.2.1 Multi-Source Real Market Data*

The framework draws on three publicly accessible data categories — a quantitative view from price and indicator series, a qualitative view from financial news, and a fundamental view from corporate financial statements — none of which requires proprietary access.

**Price series and technical indicators.** Daily OHLCV records are retrieved from Yahoo Finance and augmented with four indicators chosen as compact representatives of the main technical-signal categories: trend (EMA-7, EMA-14), momentum (MACD), and overbought/oversold conditions (RSI). Indicators follow the standard definitions in [5], with full formulas in Appendix B. Because each indicator requires a look-back history, data acquisition begins from a configurable warm-up date that precedes the first usable observation date by a margin sufficient to fully populate every indicator's window.

**Financial news.** Daily news items per ticker are obtained from the Financial Modeling Prep API; the per-day items, with headline and body text, are retained. The summarisation step that processes them is detailed in Section 3.2.3 and Appendix A.

**Corporate financial statements.** Quarterly balance-sheet and income-statement filings come from the same API. To prevent look-ahead leakage, each observation is paired with the most recent filing whose `filingDate` is on or before the last trading day of the observation window; later filings are excluded even if they cover an earlier reporting period. Selected fields and several pre-computed ratios (current ratio, debt-to-equity, gross/operating/net margin) are extracted; the rationale is part of the prompt token economy in Section 3.2.3.

To form a complete training sample, this real-data component is paired with the agent's portfolio and account context, which — for reasons developed next — is sampled synthetically rather than drawn from a single historical trajectory. The full windowing and train/test partitioning are described in Section 3.2.4.

### *3.2.2 Synthetic Portfolio and Account State Generation*

Every observation pairs the real market data of Section 3.2.1 with the agent's portfolio and account context: available cash, any open positions, and — per position — the entry price, current quantity, take-profit, and stop-loss. Without this context the model cannot reason about how much to invest, whether to add to or close a position, or whether a trade is economically meaningful given the cash on hand. A natural alternative — rolling a single hypothetical account through the historical price series and letting each day's state be the outcome of prior decisions — is structurally incompatible with GRPO's group-relative sampling (Section 3.6.1), so the context is sampled synthetically from the joint distribution described next.

**Cash sampling: a weighted-bucket distribution covering extreme regimes.** The agent operates with a fixed initial balance of \$10,000, an order of magnitude representative of a retail account; all dollar thresholds below should be read in that context. Uniform sampling over \$0–\$10,000 would under-represent the constrained regimes real traders frequently face, so available cash is drawn from a five-bucket distribution: *extreme-low* (\$0–\$50, 5%), *low* (\$50–\$300, 10%), *small* (\$300–\$1,500, 20%), *medium* (\$1,500–\$6,000, 30%), and *high* (\$6,000–\$10,000, 35%). The two low-cash buckets (15% jointly) deliberately expose the model to the *dust-buy* problem: with fractional-share trading a buy is technically valid for any positive cash balance, so a model never trained on near-empty accounts will tend to issue economically meaningless dust buys (e.g. \$5 into a \$200 stock).

The weight of the tight-cash buckets must be set with care. An earlier iteration placed 30% of samples in the combined sub-\$300 region, which proved counter-productive: when equities like AAPL trade at or above that threshold, forcing dust-only feasibility on roughly one third of training samples taught the model that dust-scale trading was the norm, driving the policy into a dust-trade local optimum from which correctness could not easily recover. The current 15% / 65% split between tight and comfortable-cash regimes preserves the lessons of low-cash regimes while leaving most of the training mass in regimes where meaningful

sizing can actually be practiced. The synthetic state distribution is thus itself a design surface that must be tuned with the downstream reward in mind.

**Joint sampling of cash and portfolio.** Available cash and holdings are not independent: a trader with \$50 in cash and no open positions is a "fresh broke account" that essentially never occurs, since a near-empty balance realistically means most capital has already been converted into positions. The framework therefore samples the two jointly, parameterising the probability of an open position as a function of cash:

$$P(\text{portfolio} \mid \text{cash}) = \begin{cases} 0.95, & \text{cash} < \$100 \\ 0.80, & \$100 \leq \text{cash} < \$500 \\ 0.60, & \$500 \leq \text{cash} < \$2{,}000 \\ 0.45, & \text{cash} \geq \$2{,}000. \end{cases} \tag{3.1}$$

This monotonically decreasing relationship enforces an implicit conservation-of-capital constraint without an explicit accounting identity, and prevents the dataset from being polluted by implausible (cash, portfolio) combinations.

**Position state generation.** When the joint sampler decides that a position is held, the position itself is characterised plausibly. The entry date is drawn uniformly from trading days 5–60 days before the observation date — a window matching the swing-trading horizon used elsewhere in the framework, excluding both intraday holds and multi-year buy-and-hold positions whose unrealised PnL would be unrealistic. Quantity is sampled as a uniform fraction in [5%, 80%] of the maximum shares the initial balance could have afforded at the entry price, floored at 0.1 share. Take-profit is drawn uniformly from [+3%, +15%] above entry and stop-loss from [−3%, −10%] below: bounds tight enough that, given the seven-day observation-window price range, the TP/SL levels are actually reachable by the price trajectory; placing them too far would leave the look-ahead-based correctness reward (Section 3.4) without positive trigger signal. Empirically, about 50% of generated TP/SL pairs trigger within the correctness-reward horizon, mirroring real swing-trading practice and giving roughly balanced training signal for "TP/SL respected" and "TP/SL not reached"

outcomes.

**Resulting state distribution.** Cash bucketing, cash–portfolio joint sampling, and the calibrated entry/quantity/TP/SL ranges together cover a far broader region of the (cash × position × risk-exposure) space than any single historical trajectory could, in roughly balanced proportion, while eliminating the temporal contamination that would otherwise let early-training mistakes propagate into later states. The resulting (real market data, synthetic portfolio context) pair forms the complete observation, which is then token-economised according to the principles in Section 3.2.3.

### *3.2.3 Prompt Token Economy*

Naively serialised, the observations above produce prompts whose length is both excessive and highly variable, with three practical consequences for GRPO: long samples are silently truncated by the tokeniser and lose their tail; per-sample variance destabilises batch behaviour and inflates the group-relative advantage estimate; and every redundant input token shrinks the completion budget available for reasoning. Four classes of token-economising transformation are therefore applied.

#### *Character-level shortening of numerical fields*

Technical indicators, prices, quantities, unrealised PnL, and TP/SL prices are rounded to two decimals, which suffices for any decision the model is asked to make. Large monetary values dominating financial statements are rewritten in scale form: any absolute value $\geq 1000$ is converted to a number followed by `thousand`, `million`, `billion`, or `trillion`, kept to one decimal with trailing zeros stripped, so `383285000000` becomes `383 billion`. Table 3.1 illustrates the effect.

Table 3.1: Character-level shortening of representative numerical fields.

| Field type | Raw form | Shortened form | Raw tokens | Shortened tokens | Reduction |
|---|---|---|---|---|---|
| Technical indicator (RSI) | 64.83729104857143 | 64.84 | 7 | 3 | 57% |
| Price (close) | 192.5301971435547 | 192.53 | 7 | 3 | 57% |
| Monetary (revenue) | 383285000000 | 383 billion | 4 | 2 | 50% |
| Monetary (cash) | 67150000000 | 67 billion | 4 | 2 | 50% |

*Selective extraction and ratio derivation for financial statements*

Rather than serialising the full balance-sheet and income-statement objects returned by the provider, the pipeline keeps only a fixed subset of raw fields most relevant to a swing-trading decision and pre-computes several ratios that analysts routinely reason in. From the balance sheet, the retained fields cover *liquidity* and *short-term capital structure* (cash and equivalents, short-term investments, net receivables, inventory, total current assets/liabilities, short-term debt), with derived ratios for solvency and leverage (current, quick, net debt, debt-to-equity). From the income statement, the retained fields trace the *profitability cascade* (revenue, gross profit, operating income, net income), normalised by margins (gross, operating, net). This selective representation reduces tokens, lifts the semantic level of the input from raw quantities to interpretable indicators, and spares the model from re-deriving the ratios within its reasoning trace. The full list and rationale are in Appendix C.

*News summarisation as a token-stability mechanism*

News is the most length-volatile component of the input. Table 3.2 compares the per-day token count of raw vs. summarised AAPL news across the full date range: raw spans 15 to 16054 tokens (mean 831, std 1058 over 1812 days), whereas the summarised stream ranges 6 to 167 tokens (mean 113, std 16 over 1096 days). The day-count discrepancy (1812 raw vs. 1096 summarised) reflects that raw items include weekends and market holidays, whereas the summarised file follows the trading calendar used in dataset construction; calendar alignment

Table 3.2: Per-day token statistics of raw vs. summarised AAPL news.

| Stream | N (days) | Min | Max | Mean | Std |
|---|---|---|---|---|---|
| Raw news (per day) | 1812 | 15 | 16054 | 831 | 1058 |
| Summarised news (per day) | 1096 | 6 | 167 | 113 | 16 |

is deferred to observation assembly (Section 3.2.4) so the raw file remains an unaltered record of news flow. The roughly 96× peak-length reduction and 65× variance reduction simultaneously keep long samples inside the context budget, stabilise batch composition and group-relative advantage estimation, and reduce padding overhead so GPU memory occupancy becomes predictable. A representative input–output example is reproduced in Appendix A.

*Truncation-safe ordering of observation fields*

Even after the above, residual overflow on news-heavy days remains possible. To ensure any forced truncation drops the least valuable information first, fields are serialised from most to least essential and from lowest to highest length variance: the technical-indicator window, portfolio context, and account status (small, near-constant footprint) sit at the head; financial-statement fields (moderate, predictable size) follow; news (longest, most variable) sits at the tail. Whenever a sample exceeds the context budget, the discarded suffix is news rather than core market state.

Together the four transformations produce prompts that are substantially shorter and substantially less variable than the unprocessed alternative; the aggregate effect on the average and maximum input token counts is reported in Chapter 4 as part of the experimental setup.

### *3.2.4 Observation Window and Dataset Splitting*

The components introduced in the preceding subsections are brought together into the JSONL files consumed by the GRPO training loop. For each ticker, the technical indicator series is traversed with a sliding window of $T = 7$ trading days and a stride of one day, yielding one observation per available window. The last trading day of each window is treated as the decision day: the model is asked, given the seven days of price and indicator history up to that day, to decide what action to take at the close of that day. The arrays of price and indicator values inside each observation are ordered from oldest to newest, with the final element representing the decision day; an explicit instruction inside the serialised observation makes this temporal convention unambiguous to the model. Each observation is then combined with the freshly sampled portfolio and account state, the most recent balance-sheet and income-statement snapshot whose filing date does not exceed the window end date, and the day's news component (raw or summarised according to the configuration) into a single JSON object whose top-level structure is shown in Figure 3.2. The internal field order of the object follows the truncation-safe arrangement motivated in Section 3.2.3: portfolio, status, and technical indicators at the head; financial statements in the middle; news at the tail.

The assembled entries are partitioned into a training split and a test split per ticker and chronologically: within each ticker, the entries are sorted by date, and the earliest 80% are assigned to the training split whereas the latest 20% are assigned to the test split. Two design considerations motivate this scheme. First, the chronological boundary prevents look-ahead leakage: no test sample's decision day precedes any training sample's decision day on the same instrument. Second, applying the split per ticker rather than across the pooled stream avoids any single ticker dominating either split when the tickers have unequal date ranges or sample counts. The exact tickers used, the date range of the underlying market data, and the resulting per-split sample counts are reported in Chapter 4 as part of the experimental setup.

```
{
  "date":   "<window-end date>",
  "ticker": "<symbol>",
  "observation": {
    "current portfolio": [ ... synthetic open positions ... ],
    "current status":    { available cash, total return, account value },
    "stock information": {
      "ticker": "<symbol>",
      "technique indicators": { 7-day arrays of OHLCV, MACD,
                                RSI, EMA-7, EMA-14 },
      "last quarter balance sheet": { selected fields + ratios },
      "last quarter income sheet":  { selected fields + ratios },
      "news" or "7-day-news":       [ raw items or daily summaries ]
    }
  }
}
```

Figure 3.2: Top-level structure of a single observation.

## *3.3 Structured Action Space and Task Formulation*

### *3.3.1 Trading Strategy Schema*

The action space of the agent is defined not as a low-dimensional categorical variable, but as a structured JSON object that encodes a complete trading decision. For each observation, the model is required to emit a top-level object with three keys:

- `reasoning thoughts` — a short natural-language rationale that references the observation (indicators, news, portfolio, status);

- `evidence for decision` — the key signals from the observation that the model relies on;

- `strategy` — the structured decision object that is consumed by the environment.

Only the `strategy` field is executable; the two preceding fields are the model's externalised reasoning trace. Requiring the model to produce the rationale and the evidence *before* the decision is a chain-of-thought style formulation: it forces the policy to commit to an explanation of the observation prior to acting, and it exposes a textual artefact that can be audited qualitatively and scored quantitatively during training (the factual reward function, defined in Section 3.4, checks that the evidence is actually drawn from the observation rather than hallucinated). The `strategy` object contains six fields, summarised in Table 3.3. Each field plays a deliberate role in realising the design principles stated in Chapter 1. Two schema-level choices deserve emphasis. First, every action — not just buys — must carry a `target price` and a `stop loss price`. This makes an exit plan a mandatory part of every decision rather than an optional afterthought, directly operationalising the closed-loop trading claim of this thesis: the model is not allowed to declare a buy without declaring the conditions under which the position will be closed. Second, `quantity` is denominated in shares rather than in a percentage of cash or a categorical size bucket (e.g. "small / medium / large"). A numeric share count is directly executable against the account state, and it lets the reward function evaluate the *notional* of the trade relative to the account value (Section 3.4) — a signal that a bucketed encoding would flatten away. Figure 3.3 shows a minimal but complete example of the three-part output for a single decision. The strict JSON schema is a deliberate design choice. It gives the reward function unambiguous access to each decision variable (Section 3.4), turns parser failures into a first-class training signal, and makes the agent–environment interface fully declarative: the same schema is used in training, sequential evaluation, and any downstream deployment.

Table 3.3: Fields of the `strategy` object.

| **Field** | **Type** | **Role** |
|---|---|---|
| `ticker` | string | Must equal the ticker in the observation. Prevents the model from issuing an action on a symbol it was not asked about. |
| `action` | {buy, sell, hold} | Categorical decision direction. |
| `number of shares` | number | Number of shares to buy or sell; fractional shares allowed. Enables *position sizing as a first-class decision*, rather than collapsing trading to a binary buy/sell switch. |
| `current price` | number | Must equal the last closing price in the observation's technical-indicator window. Anchors the decision to the observed market state and prevents price hallucination. |
| `target price` | number $> 0$ | Take-profit level. |
| `stop loss price` | number $> 0$ | Stop-loss level. |

```
{
  "reasoning thoughts": "Price has broken above the 14-day EMA
      with MACD turning positive; available cash covers a ~7%
      notional position in AAPL, which aligns with a high-
      conviction entry tier.",
  "evidence for decision": "MACD histogram flipped positive on
      the decision day; RSI=58 (not yet overbought); latest
      news summary emphasises strong iPhone demand.",
  "strategy": {
    "ticker": "AAPL",
    "action": "buy",
    "number of shares": 12,
    "current price": 192.53,
    "target price": 205.00,
    "stop loss price": 184.00
  }
}
```


Figure 3.3: Example of the three-part model output for a single observation.

### *3.3.2 System Prompt and Task Instruction Design*

The schema introduced in Section 3.3.1 specifies what the model must output. The system prompt specifies how the model is instructed to produce it. The same system prompt is reused across every training and evaluation sample, so that task framing is held constant and all variation between samples is carried by the observation rather than by the instruction itself. The prompt is organised into four blocks, each serving a role that belongs to prompt engineering rather than to the reward function.

First, the prompt assigns role and scope. It opens by declaring the model to be a professional stock trader and by stating that each invocation concerns exactly one ticker. Instruction-tuned base models respond strongly to role priming, and the single-ticker restriction removes the ambiguity that would otherwise arise when the "current portfolio" section of the observation contains positions in other symbols: the model is not asked to rebalance a basket, only to decide on the one ticker presented under "stock information".

Second, the prompt describes the input schema. It enumerates the structure of the observation that the model will receive — the ticker-level stock information object with its technical-indicator window, financial statements, and either raw or summarised news; the current portfolio list; and the current status block — and, for each field, states its semantics. The prompt specifies, for example, that the technical-indicator arrays are ordered oldest-to-newest and that the final element corresponds to the decision day, and that available cash and current account value are distinct quantities that must both be used for sizing decisions. This block is needed because the observation, once serialised as described in Section 3.2.4, is just a nested JSON blob: without an explicit reading guide, the model would have to infer the meaning of each field from its name alone.

Third, the prompt fixes two grounding constraints that anchor the model's output to the observation it was given rather than to prior beliefs. The first requires `strategy.ticker` to equal the ticker in the observation; the second requires the reported current price to equal the last closing price of the technical-indicator window. Both constraints are specific

to LLM-based agents — a classical RL policy cannot produce such mismatches in the first place — and both are verified at training time by the factual reward of Section 3.4. Stating them in the prompt, in addition to scoring them in the reward, narrows the space of outputs the model is likely to attempt and reduces the fraction of exploration cost spent on symbol or price hallucinations.

Fourth, the prompt fixes the output format contract. The output must be a single JSON object matching the schema of Section 3.3.1 exactly, with precise key names, no markdown fences, no prose outside the JSON, and termination as soon as the JSON is closed. Format-level compliance is enforced independently by the format reward and the completion-length reward of Section 3.4, but the contract itself — the declarative statement of what a well-formed output looks like — is a prompt-layer artefact. The prompt closes with a summary of the trading rules that the reward function will subsequently formalise. They are listed here by name only; their reward-side formalisation, including thresholds, multipliers, and weighting, is developed in Section 3.4.

1. **Budget feasibility.** A buy may not exceed available cash.

2. **Buy sizing.** Buy notionals are evaluated against account value in tiers, with dust-sized buys discouraged.

3. **Sell sizing.** Sell sizes are evaluated against the held position in tiers, with dust-sized sells discouraged.

4. **Hold semantics.** Holding requires a zero quantity.

5. **Take-profit and stop-loss trigger.** A held position that has reached its target price or breached its stop-loss price should be sold rather than continued.

6. **When to sell.** Sells are appropriate when the thesis deteriorates or cash must be freed, not only when the take-profit or stop-loss level is reached.

The rules above are declared in natural language inside the prompt and, in parallel, formalised as numerical reward components in Section 3.4. The prompt teaches the rules in a form the model can read; the reward grades compliance in a form the optimiser can use. The prompt does not reveal reward weights or specific penalty magnitudes, so that the model learns the task rather than a cheat sheet for the reward. A reward-only setup would force the model to discover every rule through exploration; a prompt-only setup would leave it without a gradient signal to internalise them.

### *3.3.3 Strategy Validation and Normalization*

Between the raw text emitted by the language model and its downstream consumers — the reward function during training, the portfolio simulator during evaluation — sits a thin two-layer interface. Validation is the primary layer: it enforces the schema and semantic constraints that every strategy must satisfy before it is scored or executed. Normalization is an auxiliary layer placed in front of it, whose only role is to absorb the small formatting variance that is intrinsic to language-model output. Validation takes a strategy object and returns only a boolean. It checks that all required keys are present, that the action label is one of the three allowed, that the ticker belongs to the configured universe of supported symbols, and that the numeric constraints on quantity, current price, take-profit, and stop-loss hold — with the quantity rule specialised per action, positive for buy and sell and zero for hold. Validation rewrites nothing; it only asserts that the object is a legal member of the schema of Section 3.3.1. Normalization canonicalises fields without altering the model's intent: numeric values emitted as strings are cast to float, the ticker is uppercased and stripped, the action label is lowercased, and a "hold" action has its quantity zeroed. One case deserves to be called out. Because a language model cannot be expected to emit a byte-identical schema on every sample, a small fraction of outputs arrives with a meaningful action and quantity but an omitted or non-positive take-profit or stop-loss value. In that situation the model has still committed to an actionable decision on the observation, and discarding it over a cosmetic omission would destroy a useful signal; normalization therefore

fills the missing field with the current price as a neutral placeholder, deferring the actual semantic judgement to the validation layer that follows. Table 3.4 summarises the two layers. The interface itself is identical in training and in evaluation. What differs is how

Table 3.4: The two layers of the output interface.

| | **Normalization (auxiliary)** | **Validation (primary)** |
|---|---|---|
| Purpose | Canonicalise formatting | Enforce schema and semantics |
| Output | A repaired copy of the strategy | Boolean |
| Modifies input? | Yes, via type coercion, casing, and fallback fills | No |
| Typical handling | "0.5" becomes 0.5; "aapl" becomes "AAPL"; missing TP/SL is filled | Rejects unknown ticker, wrong action label, non-positive buy quantity, missing required key |

each phase responds to a strategy that still fails validation after normalization, and that difference reflects what the phase is trying to measure. In training, the failure is kept inside the reward computation: the reward components of Section 3.4 treat an invalid strategy as a bounded negative outcome, so that the group-relative comparison used by GRPO still has a well-scaled signal to assign to malformed samples. In evaluation, the failure is kept outside the simulator: the strategy is logged with a categorical reason code, the step is executed as a no-operation, and the rate at which strategies are rejected becomes a reliability statistic reported alongside return-based metrics in Section 3.6.

## *3.4 Reward Function Design*

### *3.4.1 Design Principles*

The training signal of the agent is produced by five reward components that operate in parallel on every generated completion: the format reward, the completion length reward, the factual grounding reward, the regulation reward, and the correctness reward. Four of them are active in both stages of the GRPO fine-tuning described in Section 3.5, whereas the correctness reward is activated only in the second stage. Table 3.5 summarises their scope, activation, and range.

Table 3.5: The five reward components.

| **Reward** | **Scope** | **Stage** | **Range** |
|---|---|---|---|
| Format | Structural legality of the completion | 1, 2 | $[-1, +1]$ |
| Completion length | Token count relative to an ideal window | 1, 2 | $[-1, +1]$ |
| Factual grounding | Citation of observation values and entities | 1, 2 | $[-1, +1]$ |
| Regulation | Compliance with trading rules | 1, 2 | $[-1, +1]$ |
| Correctness | Profit relative to a do-nothing baseline | 2 only | $[-1, +1]$ |

Three principles guide the design of these components so that the weighted combination GRPO optimises remains well-defined under the stage-specific weights of Section 3.5.

First, each reward has a primary area of responsibility while deliberately allowing overlap on fundamental failures. The primary assignments are listed in Table 3.6.

The primary assignment is not exclusive. On catastrophic failures — unparseable JSON,

Table 3.6: Primary responsibility of each reward for the failure or signal categories.

| **Category** | **Primary reward** |
|---|---|
| Malformed JSON or missing top-level keys | Format |
| Schema-legal but illegal action, ticker, or quantity | Format |
| Token count far from the ideal window | Completion length |
| Hallucinated ticker, price, or indicator reference | Factual grounding |
| Over-budget buy | Regulation |
| Sell of an unheld ticker, or oversell | Regulation |
| Dust-sized buy or sell | Regulation |
| Action inconsistent with a triggered take-profit or stop-loss level | Regulation |
| Strategy profit against the do-nothing baseline | Correctness |

an illegal ticker or action label — every non-length reward registers the failure independently rather than deferring to the primary owner: malformed JSON drives format, factual, regulation, and correctness all to their floor of $-1$; an unsupported ticker is penalised both by factual (as a ticker mismatch against the observation) and by regulation (as a basic-sanity violation). An earlier iteration that enforced strict non-overlap empirically produced a weaker signal, because the GRPO advantage on a fundamentally broken output was dominated by a single bounded term rather than compounded across redundant channels. The current design preserves primary focus for nuanced signals — sizing, TP/SL triggers, factual citation — while letting the fundamental guardrails fire on every reward at once, so the gradient pressure on the most basic competences is amplified by redundancy rather than attenuated

by division of labour.

Second, every reward is bounded and normalised to the interval $[-1, +1]$. This common scale gives the Section 3.5 weights a consistent interpretation as a convex combination of equally-ranged signals, and it caps the influence that any single reward can exert on the policy gradient regardless of how its internal formula is tuned.

Third, wherever a reward encodes a continuous phenomenon, its shape is continuous rather than a step cliff. Linear ramps, piecewise-linear transitions, and a tanh squash are used in place of hard thresholds. GRPO operates on group-relative advantages, and a cliff shape collapses all samples on the same side of the threshold into an indistinguishable score; a continuous shape lets the algorithm discriminate between mild and severe violations within the same batch, producing a directional gradient that drives the policy towards compliance rather than merely away from violation.

### *3.4.2 Format Reward*

The format reward scores whether the model's completion conforms to the output schema of Section 3.3.1. A completion that cannot be parsed as JSON or that lacks the top-level "strategy" key is assigned the floor value $-1$ and skipped. Otherwise, three additive sub-checks are applied: whether a "reasoning thoughts" field is present ($\pm 0.3$), whether an "evidence for decision" field is present ($\pm 0.3$), and whether the "strategy" object passes the validation gate of Section 3.3.3 after normalization ($\pm 0.4$).

Two choices shape this design. The sub-checks are awarded partial credit rather than fused into a single pass-or-fail test, so that the reward distinguishes a completion that is partially correct — for example, a strategy that is schema-legal but lacks a reasoning trace — from one that is entirely malformed, giving GRPO a separable gradient along each axis. The validation sub-check carries more weight than the other two because a semantically invalid strategy object is the only failure that prevents every downstream reward from being computed at all, making it the most expensive structural failure in the pipeline.

### *3.4.3 Completion Length Reward*

The completion length reward is included for training stability. GRPO's group-relative advantage is meaningful only when every completion in the group can be parsed and scored by the downstream rewards, and two length regimes break this assumption. A completion truncated mid-JSON cannot be parsed, so the factual, regulation, and correctness rewards all collapse to their $-1$ floor regardless of the decision the completion contained, turning that sample into noise within its own group. A completion far below the typical length usually carries only a bare strategy object with no reasoning or evidence, which starves the factual grounding reward of the text it needs to score and deprives the model of the chain of thought that should precede the decision.

Let $L$ denote the token count of the completion and $L^*$ denote the target length. The length reward is the symmetric triangular ramp

$$r_{\text{length}} \;=\; \text{clip}\!\left(1 - \frac{|L - L^*|}{L^*},\; -1,\; +1\right), \tag{3.2}$$

which peaks at $+1$ when $L = L^*$ and decays linearly to zero on either side at $L = 0$ and $L = 2L^*$.

Two constants must be chosen: the target length $L^*$ and the training-time hard limit $L_{\max}$ beyond which a completion is truncated. The target length is set to $L^* = 800$ tokens, large enough to accommodate a full three-part output — reasoning, evidence, and the strategy JSON — and small enough to fit inside the VRAM and wall-clock budget of the training hardware. Because the attention memory grows quadratically with the sequence length and the per-token forward time grows linearly, pushing $L^*$ higher would have traded a modest quality gain for a substantial reduction in training throughput.

The truncation limit is set to $L_{\max} = 1600$ tokens, twice the target. An earlier configuration used $L_{\max} = 1200$ and left roughly 40% of early-training completions truncated, because the base model had not yet learned to fit both a full reasoning trace and a closed JSON inside that budget. GRPO is configured to mask truncated completions out of the gradient computation at the loss level, so under the 1200-token setting nearly half of every batch

contributed no learning signal at all, and the advantage values computed on the remaining completions were dominated by sampling variance. Raising the limit to 1600 reduced the early-training truncation rate to a small minority and restored the usable batch size that the advantage computation relies on.

Hard truncation at $L_{\max}$ is therefore handled outside the length reward by the masking mechanism described above. The length reward's role is the softer one of keeping the typical completion length clustered around $L^*$ rather than drifting toward either edge of the admissible window.

### *3.4.4 Factual Grounding Reward*

The factual grounding reward is designed to combat hallucination. Without a signal that ties the completion's reasoning to the observation it was given, a language model will often produce a rationale that reads plausibly but is disconnected from the specific ticker, price, indicators, and news the observation actually contains. The decision is then not supported by evidence the grader can verify, and the credibility of the agent — its claim to be acting on the market state rather than on prior beliefs memorised during pre-training — collapses. The factual grounding reward therefore scores how closely the completion is anchored in the observation along three levels: hard alignment of the decision variables against observed facts, textual reference to the named signals the observation carries, and numerical citation of the observation's actual values.

A completion that fails to parse or lacks a strategy object receives the terminal floor value $r_{\text{factual}} = -1$ and is not scored further. Otherwise, the reward is the sum of five additive terms clipped to $[-1, +1]$,

$$r_{\text{factual}} \;=\; \text{clip}(r_T + r_P + r_I + r_N + r_V,\; -1,\; +1)\,, \tag{3.3}$$

where the terms are defined below. The first level, hard alignment, contains two terms. The ticker term $r_T$ compares the ticker reported in the strategy against the ticker supplied by

the observation:

$$r_T = \begin{cases} +0.22 & \text{ticker matches the observation,} \\ -0.40 & \text{otherwise.} \end{cases} \tag{3.4}$$

The price term $r_P$ compares the strategy's reported current price $p_{\text{pred}}$ against the final closing price in the observation's technical-indicator window, $p_{\text{obs}}$. Let $e = |p_{\text{pred}} - p_{\text{obs}}|/p_{\text{obs}}$ be the relative error. Then

$$r_P = \begin{cases} +0.24 & e \leq 0.01, \\ +0.12 & 0.01 < e \leq 0.03, \\ -0.25 & e > 0.03. \end{cases} \tag{3.5}$$

Both terms are asymmetric, with mismatches carrying larger penalties than matches carry rewards. Ticker and current price are unambiguous, verifiable facts that the observation contains explicitly; reporting a different ticker or a clearly wrong price is a straightforward hallucination rather than a stylistic choice, and warrants a strong corrective signal.

The second level, textual reference, inspects the free-text fields of the completion — reasoning thoughts and evidence for decision. The indicator term $r_I$ fires when any of the keywords "rsi", "macd", "ema", "volume", "close", or "price" appears in these fields. The news term $r_N$ fires when the observation actually contains a news field and any of "news", "sentiment", "headline", or "article" appears in the text:

$$r_I = \begin{cases} +0.08 & \text{indicator keyword present,} \\ 0 & \text{otherwise,} \end{cases} \qquad r_N = \begin{cases} +0.08 & \text{news is present in the observation} \\ & \text{and a news keyword is cited,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

The asymmetry here runs in the opposite direction from the hard alignment terms: these two terms are positive-only. Keyword presence is a low-precision signal — it shows that the model engaged with the observation in words, but its absence does not prove disengagement, because legitimate reasoning styles may paraphrase a trend without naming RSI or MACD

by their acronyms. A symmetric $\pm 0.08$ penalty would punish stylistic variation rather than grounding failure, whereas a positive-only bonus rewards explicit citation without penalising equally valid paraphrase.

The third level, numeric citation, rewards approximate matches between numbers appearing in the free-text fields and specific observation values. Seven observation values are considered: the final closing price, available cash, current account value, and the final values of RSI, MACD, EMA-7, and EMA-14. A number in the text is considered a match for one of these values when their relative error is within 3%, with a small absolute tolerance for near-zero quantities. Let $N$ be the number of distinct observation values that are cited in this way. Then

$$r_V \;=\; \min\bigl(0.18,\; 0.03\,N\bigr). \tag{3.7}$$

The per-match granularity of 0.03 gives GRPO a continuous signal that distinguishes, for example, a completion citing one observed value from one completion citing four, whereas the cap at $+0.18$ prevents numeric citation alone from dominating the factual score: grounding is valued across all three levels, not concentrated in one.

### *3.4.5 Regulation Reward*

The regulation reward enforces trading rules that are observable at the moment of decision, i.e., rules whose violation can be detected from the current observation alone without running the simulator. It is included for three reasons. First, several rule violations — dust trades, hold with non-zero quantity, selling what you do not own — pass the JSON schema and therefore earn a neutral format reward, but describe strategies no reasonable trader would place; regulation provides the penalty signal these failures need. Second, a few violations (e.g., buying more than the available cash) are also intercepted by the simulator inside the correctness reward, but only as a single terminal $-1$; the regulation reward exposes a smooth, severity-scaled gradient so the policy learns how much it overshot, not merely that it overshot. Third, Stage 1 runs without the correctness reward, so regulation is the only

signal shaping action selection and sizing during cold-start; if this signal is weak, the policy converges to shortcut behaviors (trivial holds, dust buys) that survive the rest of Stage 1 unchallenged and bottleneck Stage 2.

The reward is a sum of small additive terms grouped into five rule families. Each term is designed to be interpretable on its own and bounded in $[-1, +1]$ after clipping.

*Rule family 1: basic sanity.*

Three observation-agnostic checks catch parseable-but-nonsense outputs: the ticker must belong to the traded universe ($+0.08/-0.16$), the action must be one of `buy`, `sell`, or `hold` ($+0.08/-0.16$), and the current price must be positive ($+0.06/-0.12$). Magnitudes are deliberately small so these act as shaping rather than dominating the reward; the asymmetric scaling (penalties twice the reward) reflects that a violation here is a categorical failure whereas a pass is only the baseline expectation.

*Rule family 2: buy-side rules.*

Three terms govern buy actions. The motivation and numerical shape of each is as follows.

**(i) Budget legality.** A buy is legal when $q \cdot p_{\text{curr}} \leq C_{\text{avail}}$, where $q$ is the ordered quantity, $p_{\text{curr}}$ the current price from the strategy, and $C_{\text{avail}}$ the available cash in the observation. A legal buy receives $+0.12$; an illegal one receives $-0.30$. This is the binary pass/fail signal.

**(ii) Budget-violation severity.** On top of the $-0.30$ flat penalty, illegal buys receive a graded term

$$\Delta_{\text{budget}} = -0.5 - 0.5 \cdot \min\left(1,\ \frac{q \cdot p_{\text{curr}} - C_{\text{avail}}}{\max(C_{\text{avail}},\ 1)}\right), \tag{3.8}$$

bounded in $[-1, 0]$. Small overshoots (few percent above available cash) collect a penalty near $-0.5$; doubling the budget or more saturates at $-1$. Motivation: the simulator already rejects unaffordable buys as a hard $-1$ in the correctness reward, which provides no gradient about

how over-budget the order was. This smooth term restores that gradient inside regulation, letting the policy learn proportional restraint rather than merely "illegal or not".

**(iii) Buy sizing (dust-trade shaping).** A legal buy still requires a meaningful position size. Early runs showed that without a sizing term, the policy collapses all its buys to trivial quantities: a dust buy clears every other check in this reward (legal ticker, valid action, positive price, within-budget) and so earns a small positive regulation score, turning the smallest quantity into the safest way to pass Stage 1. The sizing reward eliminates this shortcut by making dust strictly negative while rewarding buys that move a material fraction of the available budget. One notional fractions govern the piecewise shape: $f_{\text{acct}} = q \cdot p_{\text{curr}}/V_{\text{acct}}$ measures the buy as a share of the account and is used for dust detection. The term is applied only when the buy is legal, since rewarding an unaffordable buy for being "well-sized" against a notional it cannot pay for would be misleading. Table 3.7 gives the payoffs.

Table 3.7: Buy sizing reward.

| Range | Reward | Interpretation |
|---|---|---|
| $f_{\text{acct}} < 2\%$ | $-0.80 + 0.30 \cdot \left(f_{\text{acct}}/2\%\right)$ | dust ramp, gradient preserved |
| $f_{\text{acct}} < 5\%$ | $-0.20$ | undersized buy |
| $5\% \leq f_{\text{acct}} < 15\%$ | $+0.30$ | sweet spot for single-ticker exposure |
| $f_{\text{acct}} \geq 15\%$ | $+0.20$ | large bet, slightly de-emphasised |

*Rule family 3: sell-side rules.*

Three terms govern sell actions. Let $q_{\text{held}}$ be the quantity of the sold ticker currently held in the portfolio.

**(i) Ownership.** A sell of a ticker with $q_{\text{held}} = 0$ is meaningless (short selling is not allowed in this framework) and receives $-0.45$; a sell of a held ticker receives $+0.12$.

**(ii) No over-sell and positive quantity.** Conditional on ownership, the ordered quantity must satisfy $0 < q \leq q_{\text{held}}$. Being within the held quantity adds $+0.10$; exceeding it adds $-0.35$. A strictly positive sell quantity adds $+0.06$; zero or negative adds $-0.14$. These two terms together ensure that a valid sell is a real trim of the actual position.

**(iii) Sell sizing (dust-trim shaping).** Sell sizing mirrors the buy side on the fraction of the held position being liquidated, $f_{\text{sell}} = q/q_{\text{held}}$, and is applied only when the sell is feasible (held and within position). The motivation matches dust buys: a 1%–5% trim passes every gate but does not materially de-risk the position, and without this term the policy has an incentive to emit tiny sells whenever it wants to "look active" without committing to a real rebalance. As on the buy side, the dust region is shaped as a linear ramp rather than a flat floor to preserve within-group gradient under GRPO. Table 3.8 gives the payoffs.

Table 3.8: Sell sizing reward.

| Range | Reward | Interpretation |
|---|---|---|
| $f_{\text{sell}} < 10\%$ | $-0.70 + 0.25 \cdot \left(f_{\text{sell}}/10\%\right)$ | dust ramp, gradient preserved |
| $10\% \leq f_{\text{sell}} < 25\%$ | $-0.20$ | light trim, weak signal |
| $25\% \leq f_{\text{sell}} < 50\%$ | $+0.25$ | meaningful rebalance |
| $f_{\text{sell}} \geq 50\%$ | $+0.20$ | strong trim or full exit |

*Rule family 4: hold semantics.*

Hold means no trade. A hold with $q = 0$ receives $+0.10$; any non-zero quantity under a hold action receives $-0.25$. This is a small but important consistency check, since the schema

lets the model emit any quantity regardless of action, and non-zero-qty holds are a common early-training failure mode that simulators reject but format does not catch.

*Rule family 5: TP/SL trigger-based action consistency.*

A held position carries a stored take-profit price $p_{\text{TP}}$ and stop-loss price $p_{\text{SL}}$ which the model itself set on a prior buy. The current price in the observation is enough to determine whether either threshold has been crossed:

$$\text{TP triggered} \iff p_{\text{curr}} \geq p_{\text{TP}}, \qquad \text{SL triggered} \iff p_{\text{curr}} \leq p_{\text{SL}}. \tag{3.9}$$

When a trigger is observed, the model is expected to sell the position. The reward structure is asymmetric: stop-loss is treated as more urgent than take-profit because delaying a stop-loss materially exposes the portfolio to further downside, whereas delaying a take-profit forfeits unrealized profit but does not increase risk. Table 3.9 gives the full payoff matrix.

Table 3.9: TP/SL trigger-based action-consistency reward.

| Trigger | sell | hold | buy |
|---|---|---|---|
| Stop-loss triggered | +0.45 | −0.30 | −0.40 |
| Take-profit triggered | +0.35 | −0.20 | −0.30 |

This term is what makes TP/SL a functional trading primitive rather than a cosmetic field the model writes and forgets. Because the trigger is computed from the current observation only, no future lookahead is required, and the signal remains deterministic and well-defined on any day of the evaluation horizon.

### *3.4.6 Correctness Reward*

The correctness reward is the primary profit signal of the framework and the only reward component that directly answers the question "was this a good trade?". It is active only in

Stage 2, once the policy is already emitting well-formed and rule-compliant strategies thanks to the four Stage 1 rewards, so that every completion reaching the correctness reward is a syntactically and structurally legitimate candidate trade whose economic quality is worth grading. The core idea is simple: simulate what the agent's strategy does to the account over a short look-ahead horizon, compare that to what would happen if the agent did nothing over the same horizon, and translate the relative difference into a bounded scalar through a tanh squash. This baseline-relative design is what puts buy, sell, and hold on the same scale and lets GRPO compare them as apples to apples.

**Baseline-vs-action core.** Every evaluation of the correctness reward runs two parallel forward simulations starting from the decision day $t$ and ending $H$ trading days later. The horizon $H$ is a configurable parameter of the framework, and we set $H = 5$ to match the five active trading days of a standard week — long enough to give a strategy's take-profit and stop-loss a realistic chance to trigger within the price trajectory, short enough that exogenous events accumulating beyond a single week cannot dominate the credit assigned to any one decision. The baseline simulation represents the do-nothing counterfactual: cash is left untouched and every position currently held walks forward under its own stored take-profit and stop-loss prices. Let $C_t$ denote the available cash at decision time, $\mathcal{P}_t$ the current portfolio, and $p_{i,\tau}$ the closing price of ticker $i$ on trading day $\tau$. For each held position $i \in \mathcal{P}_t$ with take-profit $p_i^{\mathrm{TP}}$ and stop-loss $p_i^{\mathrm{SL}}$, define the horizon exit price

$$e_i \;=\; \begin{cases} p_i^{\mathrm{TP}} & \text{if } \exists \tau \in [t{+}1, t{+}H] :\; p_{i,\tau} \geq p_i^{\mathrm{TP}} \text{ first,} \\ p_i^{\mathrm{SL}} & \text{else if } \exists \tau \in [t{+}1, t{+}H] :\; p_{i,\tau} \leq p_i^{\mathrm{SL}} \text{ first,} \\ p_{i,\,t+H} & \text{otherwise (mark-to-market at horizon end).} \end{cases} \tag{3.10}$$

The baseline account value at horizon end is then

$$V_{t+H}^{\mathrm{base}} \;=\; C_t + \sum_{i \in \mathcal{P}_t} q_i \cdot e_i, \tag{3.11}$$

where $q_i$ is the quantity of ticker $i$ held at time $t$. The action simulation executes the strategy first — debiting cash for a buy, crediting cash for a sell, or leaving cash unchanged for a hold — and then walks the resulting portfolio forward under the same TP/SL rule. Writing $\mathcal{P}'_t$ and $C'_t$ for the portfolio and cash immediately after executing the strategy, the action value is

$$V^{\text{act}}_{t+H} \;=\; C'_t + \sum_{i \in \mathcal{P}'_t} q'_i \cdot e'_i, \tag{3.12}$$

with $e'_i$ computed exactly as in Equation (3.10) but using the updated positions and the strategy's own TP/SL for the freshly opened or topped-up position. The correctness signal is the normalised excess return

$$\text{excess}_t \;=\; \frac{V^{\text{act}}_{t+H} - V^{\text{base}}_{t+H}}{V^{\text{acct}}_t}, \tag{3.13}$$

where $V^{\text{acct}}_t$ is the total account value (cash plus current mark-to-market of all held positions) at decision time. Dividing by $V^{\text{acct}}_t$ makes the signal account-size-invariant so that a \$200 excess on a \$10,000 account is rewarded the same as a \$20 excess on a \$1,000 account. The raw excess is then squashed through a scaled tanh to obtain the base correctness reward

$$r^{\text{base}}_t \;=\; \tanh\!\left(\frac{\text{excess}_t}{\sigma}\right), \qquad \sigma = 0.02. \tag{3.14}$$

The scale $\sigma = 0.02$ means a 2% horizon-level excess return saturates to roughly +0.76, a 5% excess to +0.99. The choice of 2% as the saturation reference is itself calibrated to the structure of weekly trading. A 2% excess sustained over the five-day horizon, if achieved on every one of the $\sim 50$ weekly horizons in a calendar year, would compound into an annualised excess of roughly 171% ($1.02^{50.4} \approx 2.71$) over the do-nothing baseline — a return profile far beyond what any documented systematic strategy sustains on out-of-sample equity data, where the long-run S&P 500 return sits near 10% annually and well-known quantitative funds report annualised excess returns roughly in the 15–40% range. We therefore treat a per-decision excess of 2% over five trading days as the upper edge of an *informative* signal: outcomes in the 0–2% band fall inside the near-linear region of tanh where the gradient is

most discriminative and the policy sees the strongest pull, while outcomes beyond that band saturate so that a single unusually lucky (or unlucky) week cannot dominate a training batch and the policy is steered toward producing consistently solid trades rather than chasing tail outcomes. Figure 3.4 illustrates the two parallel simulations and the quantities that feed into Equation (3.13).

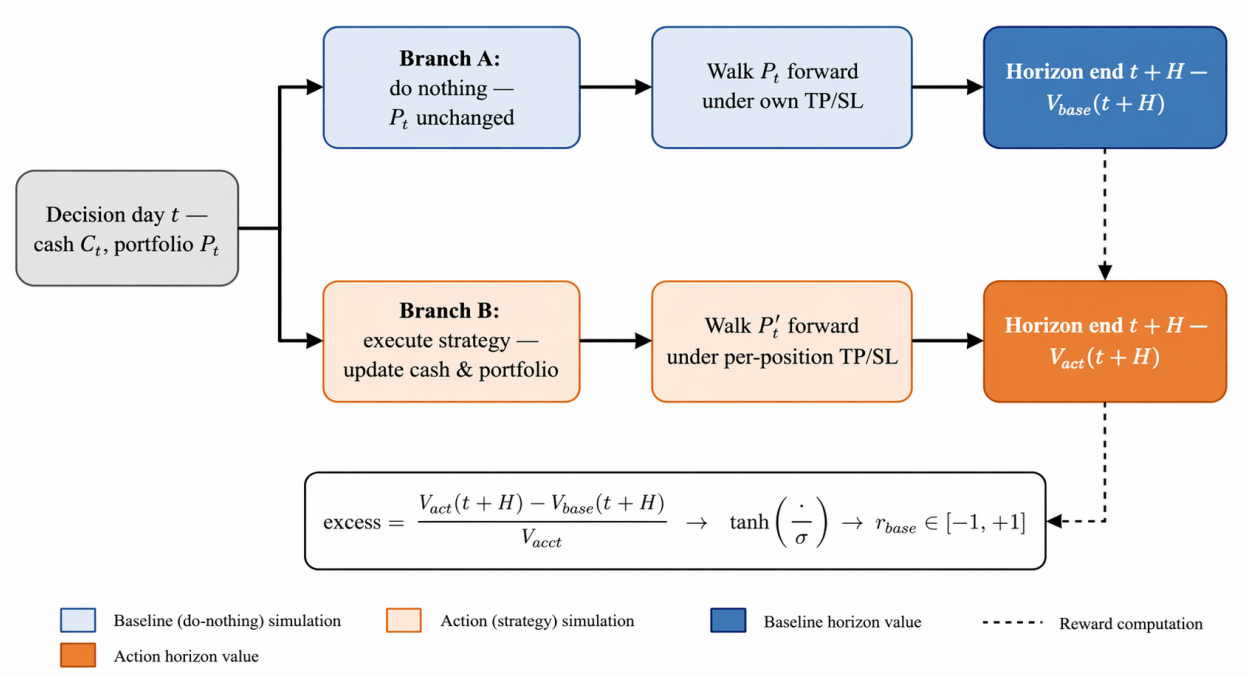


Figure 3.4: The two parallel simulations that define the correctness reward. Starting from the shared state on decision day $t$, Branch A represents the do-nothing baseline and Branch B represents executing the strategy. Both branches walk their respective portfolios forward for $H = 5$ trading days under each position's take-profit / stop-loss rule. The difference of the two horizon-end account values, normalised by the decision-day account value, feeds into a scaled tanh squash that produces the base correctness reward $r_t^{\text{base}}$. Hold is the case where Branch B is identical to Branch A, which naturally yields $r_t^{\text{base}} = 0$.

Hold is a first-class case of this formulation, not a special one. By construction, executing

a hold does not change cash or portfolio, so $\mathcal{P}'_t = \mathcal{P}_t$ and $C'_t = C_t$; the two simulations then produce the same exit price for every position, $V^{\text{act}}_{t+H} = V^{\text{base}}_{t+H}$, and the base reward is exactly $\tanh(0) = 0$. This is intentional. Hold sits at the neutral point of the reward scale and becomes positive or negative only through GRPO's group comparison: when a sibling completion on the same prompt proposes a buy that loses money over the horizon, the buy gets a negative base reward and hold's 0 wins the relative advantage; when a sibling proposes a buy that gains, hold loses the relative advantage. The same mechanism applies when the agent starts with an empty portfolio. With no positions to walk forward, the baseline value is just $V^{\text{base}}_{t+H} = C_t$; a hold trivially matches it and scores 0, whereas a buy opens a new position whose horizon exit price $e'_i$ is compared against the unchanged cash. There is no degenerate case where the reward is undefined or asymmetric: the baseline-vs-action mechanism reduces cleanly to a "bought a new position vs. stayed in cash" comparison when the portfolio is empty, and to a "did something vs. rode out existing positions" comparison when it is not.

**Dust and sizing shape.** Two multiplicative/additive modifiers attach to $r^{\text{base}}_t$ for non-hold actions. A dust short-circuit returns a linear penalty ramp in $[-0.95, -0.70]$ for any buy whose notional is below 2% of the account or any sell whose quantity is below 10% of the held position, so that trades which are technically legal but economically null cannot free-ride on a favourable market; the ramp preserves a monotonic gradient that GRPO can exploit even when every rollout in a group falls inside the dust zone. Non-dust trades then receive a conviction multiplier $\alpha \in [0.5, 1.4]$ for buys and $\alpha \in [0.5, 1.3]$ for sells, amplifying the base reward when sizing falls in the sweet spot of the regulation tables already introduced. The payoffs are summarised in Tables 3.10 and 3.11.

**Final form.** The assembled correctness reward, for any completion that survives the hard gates and the dust short-circuit, is

$$r^{\text{corr}}_t \;=\; \text{clip}\big(\alpha \cdot r^{\text{base}}_t,\; -1,\; +1\big)\,, \tag{3.15}$$

Table 3.10: Buy-side conviction multiplier applied to $r_t^{\text{base}}$.

| $f_{\text{buy}}$ range | Multiplier $\alpha$ | Interpretation |
|---|---|---|
| $2\% \leq f_{\text{buy}} < 5\%$ | linear $0.5 \to 1.0$ | undersized, signal dampened |
| $5\% \leq f_{\text{buy}} < 15\%$ | linear $1.0 \to 1.4$ | sweet spot, signal amplified |
| $f_{\text{buy}} \geq 15\%$ | 1.4 | large bet, capped at sweet-spot peak |

Table 3.11: Sell-side conviction multiplier applied to $r_t^{\text{base}}$.

| $f_{\text{sell}}$ range | Multiplier $\alpha$ | Interpretation |
|---|---|---|
| $10\% \leq f_{\text{sell}} < 25\%$ | linear $0.5 \to 1.0$ | light trim, signal dampened |
| $25\% \leq f_{\text{sell}} < 50\%$ | linear $1.0 \to 1.3$ | meaningful rebalance, signal amplified |
| $f_{\text{sell}} \geq 50\%$ | 1.3 | strong trim or full exit, capped |

with $\alpha$, trade sizing multiplier, $= 1$ for hold. The formal output range is therefore $[-1, +1]$, the same as the other four reward components, which is what allows the Stage 2 weighted combination to be interpreted as a convex mixture of comparable signals.

## *3.5 Two-Stage GRPO Fine-Tuning*

### *3.5.1 GRPO and LoRA Setup*

The training algorithm is Group Relative Policy Optimisation (GRPO) [7], the PPO variant introduced by DeepSeek for reasoning-heavy LLM fine-tuning. Compared to standard PPO, GRPO replaces the learned value network with an on-the-fly group baseline computed from $K$ parallel rollouts of the same prompt, which removes the critic's training cost and avoids the value-function instability that is particularly pronounced when the reward signal is a heterogeneous mixture of structural and economic terms as in Section 3.4. For each prompt $x$, GRPO samples $K$ completions $\{y_i\}_{i=1}^{K}$ from the current policy $\pi_\theta$, evaluates each through

the weighted reward $r_i = \sum_k w_k \cdot r_i^{(k)}$, and converts the raw rewards into group-relative advantages

$$A_i \;=\; \frac{r_i - \mathrm{mean}(r_1, \ldots, r_K)}{\mathrm{std}(r_1, \ldots, r_K) + \varepsilon}, \tag{3.16}$$

so that what matters is each completion's rank within its sibling group rather than its absolute reward. The policy is then updated by maximising the clipped surrogate objective with a Kullback–Leibler penalty against a frozen reference policy $\pi_{\mathrm{ref}}$,

$$\mathcal{J}(\theta) \;=\; \mathbb{E}_{x,\,\{y_i\}}\left[\frac{1}{K}\sum_{i=1}^{K}\min\Big(\rho_i A_i,\;\mathrm{clip}(\rho_i, 1{-}\epsilon, 1{+}\epsilon)\,A_i\Big)\right] \;-\; \beta \cdot D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\,, \tag{3.17}$$

where $\rho_i = \pi_\theta(y_i|x)/\pi_{\theta_{\mathrm{old}}}(y_i|x)$ is the importance ratio against the policy snapshot used for sampling, $\epsilon$ is the PPO clip width, and $\beta$ controls how strongly the trained policy is anchored to the reference.

Intuitively, the objective in Equation (3.17) drives the policy toward higher-reward completions through a sign-based gradient mechanism. For each sibling $y_i$, the advantage $A_i$ inherits the sign of $r_i - \mathrm{mean}(r)$: completions whose weighted reward sits above the group mean have $A_i > 0$, so maximising $\rho_i A_i$ increases $\pi_\theta(y_i \mid x)$ at every token of $y_i$ and makes the policy more likely to regenerate that completion on the same prompt in the future; completions below the group mean have $A_i < 0$, and the same expression decreases their per-token likelihoods. The two guard rails sit on top of this signal: the clip $[1{-}\epsilon, 1{+}\epsilon]$ truncates the importance ratio whenever a single sample tries to push the policy too far away from the sampling snapshot, and the $\beta\, D_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})$ term pulls the policy back toward a frozen reference so that batch-level gains cannot collapse the broader output distribution. Over many GRPO steps the policy mass therefore migrates toward regions of completion space that consistently rank highest within their sibling group, which is precisely what the multi-component reward of Section 3.4 is designed to define.

The group-relative formulation in Equation (3.16) is also what makes the synthetic portfolio design of Section 3.2 a structural requirement rather than a stylistic choice. The $K$ rollouts on a single prompt represent $K$ mutually exclusive hypothetical actions on the same

decision day; only one of them could be "the next observation" under any sequential simulator, which is incompatible with the parallel sibling structure GRPO needs. Decoupling each training observation from any specific prior trajectory, by sampling cash and positions synthetically, is the cleanest way to make every prompt a self-contained optimisation target that GRPO can score $K$ times without contradiction.

Full fine-tuning of the 3B backbone model is infeasible on the 24 GB GPUs available for this work once gradient buffers, optimiser state, and KV cache for $K$ rollouts per prompt are accounted for. We therefore train through Low-Rank Adaptation (LoRA), which freezes the base weights and inserts a pair of trainable low-rank matrices into selected linear layers, reducing the trainable parameter count to well under 1% of the original model. The adapter configuration used throughout this work is $r = 16$, $\alpha = 16$, dropout $= 0$, no bias adaptation, and target modules covering all attention projections ($q$, $k$, $v$, $o$) and all MLP projections (gate, up, down) of every transformer block. A uniform application across attention and MLP is consistent with current practice for instruction-tuning of small LLMs and was retained here without ablation because the training bottleneck in this work is the reward design and the two-stage schedule, not the adapter capacity.

### *3.5.2 Two-Stage Training Schedule*

The five reward components of Section 3.4 are not turned on simultaneously; they are split across two sequential stages. Empirically, a single-stage run that activates all five rewards at once leaves the correctness signal gradient-starved: at cold start almost every completion fails the schema gate, the tanh path of correctness is rarely reached, and the group-relative advantages introduced in Section 3.5.1 are dominated by structural-error variance. Splitting the schedule into a Stage 1 that stabilises output structure and a Stage 2 that pursues profit converts this single noisy objective into two sharper ones, which is what makes the full reward stack tractable in practice.

**Stage 1: output shaping.** Stage 1 trains under the four rewards introduced in Sections 3.4.2 through 3.4.5 (format, length, factual grounding, regulation), each contributing with unit weight. Correctness is intentionally not enabled, so every gradient step in Stage 1 pushes the policy purely toward producing well-formed, length-appropriate, observation-grounded, and rule-compliant strategies. The exit point of Stage 1 is determined empirically rather than by a fixed step budget. We run a long exploratory training run, monitor the per-reward running means on the validation prompts, and record the step at which the slowest of the four rewards has plateaued and just begins to regress (a typical sign that the policy is starting to over-fit the structural objectives at the expense of generation quality). The selected step count is then used as the Stage 1 budget for production runs at that dataset size, and is re-tuned whenever the dataset is regenerated.

**Stage 2: profit optimisation.** Stage 2 introduces correctness as the dominant signal while retaining all four Stage 1 rewards as low-weight regularisers to prevent structural decay. Correctness takes the majority of the reward budget, with regulation occupying the largest share of the regulariser budget so the rule-compliance gradient stays active under correctness-dominated updates, and length, format, and factual grounding sharing the remainder so the schema, completion length, and observation grounding established in Stage 1 do not drift. The production weights are listed in Table 3.12.

Table 3.12: Stage 2 reward weights.

| Reward component | Weight | Role in Stage 2 |
|---|---|---|
| Correctness | 0.60 | Primary profit signal |
| Regulation | 0.16 | Preserve rule compliance |
| Length | 0.08 | Preserve format-friendly completion length |
| Format | 0.08 | Preserve JSON schema and reasoning fields |
| Factual | 0.08 | Preserve evidence grounding |

The Stage 2 budget is set by the desired epoch coverage of the training set rather than by a fixed step count. Targeting a coverage of roughly 0.5 to 1.0 epoch keeps the policy long enough on the data to escape the dust local optimum that correctness is designed to penalise, but stops short of mode collapse onto specific training-set ticker–date combinations that would erode validation performance.

**Stage-specific hyperparameter choices.** Beyond the reward weights, the two stages are configured with deliberately different sampling and optimisation hyperparameters, listed in Table 3.13. The differences are not cosmetic: each one reflects a property of the corresponding training objective.

Table 3.13: Two-stage hyperparameter contrast

| Hyperparameter | Stage 1 | Stage 2 | Reason for the difference |
|---|---|---|---|
| Learning rate | $1.5\times10^{-5}$ | $3\times10^{-6}$ | Stage 2 must protect Stage 1 gains; lower LR slows decay |
| Sampling temperature | 0.80 | 0.90 | Stage 2 widens the quantity distribution to escape dust |
| Top-$p$ | 0.85 | 0.92 | Same rationale as temperature; broader exploration |
| KL penalty $\beta$ | 0.15 | 0.15 | Same for stable training |
| Rollouts per prompt $K$ | 6 | 6 | Both large siblings for stable group advantages |

The asymmetry is concentrated in two themes. First, Stage 2 lowers the learning rate by roughly 4× because the cost of breaking the structural behaviour acquired in Stage 1 is high and the correctness gradient is noisier than the Stage-1 rewards. Second, Stage 2 widens the sampling distribution ($T = 0.90$, top-$p = 0.92$) so that sibling rollouts differ materially in quantity; without that diversity, the group-relative advantage of Section 3.5.1 collapses toward zero in the dust local optimum that Section 3.4.6 was designed to penalise, and the policy cannot move.

## *3.6 Evaluation Methodology*

### *3.6.1 Sequential Simulation Environment*

Evaluation uses sequential simulation rather than the synthetic portfolio states of Section 3.2.2. The reason is a change of question: training asked the policy to score a prompt in isolation through GRPO's parallel rollouts, whereas evaluation asks whether the policy can steer a real account through a trading horizon, and only a single realised trajectory can answer the latter. Each evaluation step therefore feeds the policy an observation whose cash, portfolio, and account value are the exact output of the previous step's decision, rather than independently sampled. Figure 3.5 contrasts the two regimes side by side.

Evaluation is run one ticker at a time. At the start of each ticker the account is reset to the configured initial balance with an empty portfolio, and the policy acts on every trading day until the last date in that ticker's test file. This per-ticker isolation attributes realised PnL cleanly to the (policy, ticker, evaluation period) triple, prevents one ticker's blow-up from starving the account into silent no-ops on the others, and makes the aggregate a simple average over comparable per-ticker outcomes rather than a path-dependent mix.

Strategy rejection at evaluation reuses the rule set of training: the schema validator of Section 3.3.3 and the hard gates on buy budget, unheld sell, and oversell from the correctness reward (Section 3.4.6). A rejected strategy is logged with its reason and treated as a no-op for that day, so cash and portfolio are unchanged. Keeping the rule set identical between training and evaluation guarantees that no behaviour internalised by the policy is silently waived at test time.

### *3.6.2 Metrics*

Evaluation reports three trajectory-level metrics for each ticker run: the cumulative return rate (CR), the annualised Sharpe ratio (SR), and the maximum drawdown (MDD). All three are computed from the daily total-asset series $V_0, V_1, \ldots, V_T$, where $V_0$ equals the configured initial balance and $V_T$ is the mark-to-market account value on the last trading day of the

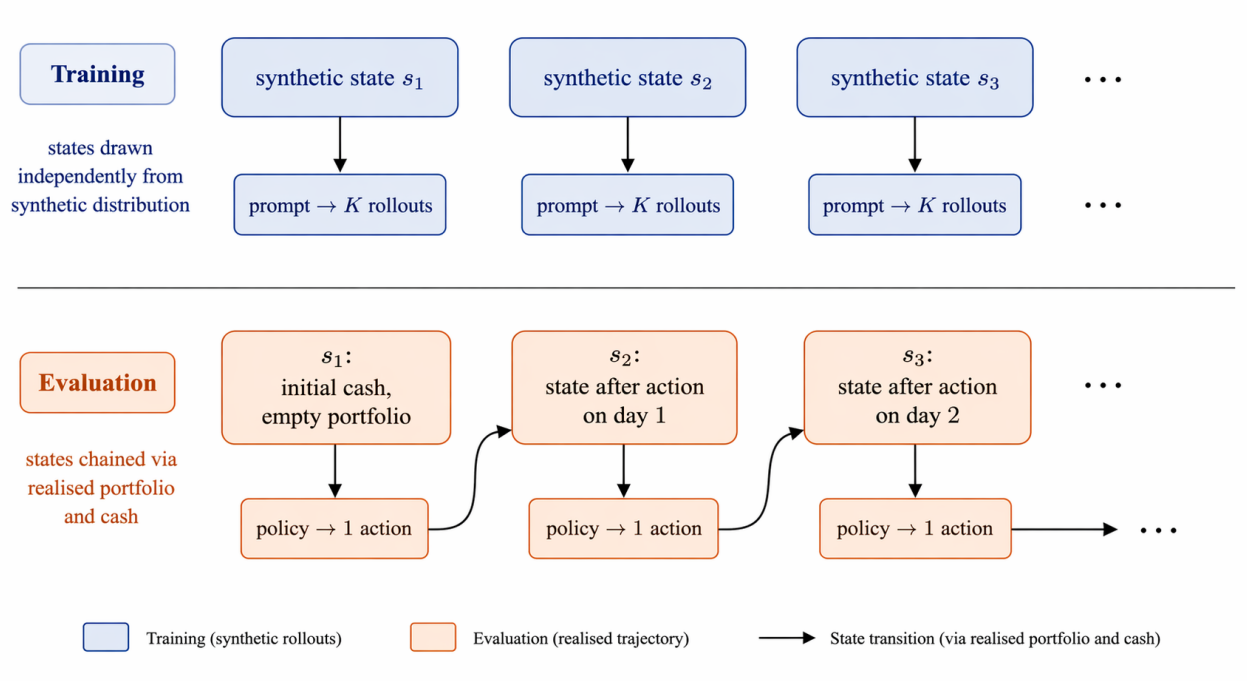


Figure 3.5: Training vs. evaluation regime. Training prompts each carry a synthetic state sampled independently of any prior trajectory, which is what enables GRPO's $K$-rollout group-relative advantages (Section 3.5.1). Evaluation chains states sequentially: the portfolio and cash of step $t{+}1$ are the exact outcome of the policy's action on step $t$, so the full run produces a single realised PnL trajectory against which the metrics of Section 3.6.2 are computed.

evaluation period.

CR is the net change in account value as a fraction of starting capital,

$$\mathrm{CR} = \frac{V_T - V_0}{V_0}, \tag{3.18}$$

reporting the end-to-end profit or loss of the run.

SR is computed from the daily returns $r_t = V_t/V_{t-1} - 1$ using the sample standard deviation (unbiased estimator with ddof $= 1$) and annualised by the 252 trading-day convention,

$$\mathrm{SR} = \frac{\bar{r} - r_f}{\sigma_r}\sqrt{252}, \tag{3.19}$$

where $\bar{r}$ and $\sigma_r$ are the mean and standard deviation of $\{r_t\}_{t=1}^{T}$ and $r_f = 0.00011773$ is the daily risk-free rate (roughly 3% annualised). A small floor on $\sigma_r$ guards against division blow-up when the portfolio is effectively static across the run.

MDD is the largest peak-to-trough relative fall observed along the asset series,

$$\text{MDD} = \max_{1 \le t \le T} \frac{\max_{s \le t} V_s - V_t}{\max_{s \le t} V_s}. \tag{3.20}$$

MDD is orthogonal to CR: two runs with the same end-to-end return can differ substantially in their worst intermediate drawdown, and this metric captures the risk profile that CR cannot.

A win rate is not reported. A hit is well defined only when each trade has an unambiguous open–close pair, as in all-in / all-out DQN setups; under the quantity-aware action space here a single buy can be closed through several partial sells at different prices (and vice versa), so there is no canonical way to pair fills into trades. The three trajectory-level metrics above describe the realised PnL path directly without requiring per-trade accounting.

### *3.6.3 Research Questions and Comparison Design*

The evaluation is organised around three research questions that together cover the main variables of this work: the choice of policy, the composition of the training dataset, and the shape of the action space. Each research question is decomposed into paired ablation comparisons, listed in Table 3.14. All pairs are evaluated under the same sequential simulation of Section 3.6.1 and scored on the same three metrics of Section 3.6.2, so that any difference observed between two rows of a pair is attributable to the single variable being ablated.

Table 3.14: Research question matrix.

| RQ | Comparison A | Comparison B | Question answered |
|---|---|---|---|
| *RQ1: policy choice* | | | |
| 1-a | Buy-and-hold | Llama-3.2-3B + RL | Does the policy add value over a passive benchmark? |
| 1-b | Llama-3.2-3B (no train) | Llama-3.2-3B + RL | What does our RL training contribute over the base LLM? |
| 1-c | Qwen3-4B (no train) | Qwen3-4B + RL | How does our framework apply to an alternative LLM base? |
| 1-d | Craig's DQN (re-run) | Llama-3.2-3B + RL | Does LLM + RL improve over a traditional RL trader? |
| *RQ2: Training-budget dimension* | | | |
| 2-a | 1000 steps ($\approx 0.5$ epoch) | 2000 steps ($\approx 1.0$ epoch) | Does extending training improve the trading performance? |
| *RQ3: action-space design* | | | |
| 3-a | Buy / sell only | Quantity-aware | Does quantity flexibility improve performance? |

# Chapter 4

# EXPERIMENTS

## *4.1 Experiment Setup*

Most experiments were run on a single workstation with one NVIDIA RTX A4500 (24 GB VRAM). vLLM is disabled because the 24 GB budget cannot host both the policy-with-gradient path and a second inference replica at the context length this task requires. Other experiments run on a NVIDIA A100 (80 GB VRAM) to accelerate the training speed with vLLM on. The base model is Llama-3.2-3B-Instruct, fine-tuned with the LoRA adapter described in Section 3.5.1, and trained with the two-stage GRPO schedule of Section 3.5.2.

The ticker universe consists of five U.S.-listed equities, one per market archetype, chosen so that the evaluation covers qualitatively different return and volatility regimes rather than five variations of the same behaviour. Table 4.1 lists the selection and the rationale for each slot.

Table 4.1: Evaluation ticker universe.

| Category | Ticker | Rationale |
|---|---|---|
| Market leader | AAPL | Mega-cap technology benchmark with deep news coverage |
| Cyclical | CAT | Growth-cyclical equity with pronounced momentum swings |
| Defensive (stable) | KO | Consumer-staples proxy with low volatility and steady drift |
| Financial | JPM | Large-bank exposure tied to rate-regime and macro news |
| High volatility | GME | High volatility symbol in price trend with small price per share |

The dataset covers the window defined in Section 3.2 and is partitioned 80%/20% along the time axis, with the later 20% reserved as the test split; random shuffling is applied only to

train dataset, so no future observations can leak into training. Training runs for 500 Stage-1 steps, terminated when the four shaping rewards plateau on the held-out split, followed by 1000 Stage-2 steps, sized to cover roughly 0.5 epoch(s) of the training set at the Stage-2 batch configuration.

## *4.2 Training Dynamics*

This section reports the empirical learning behaviour of the two GRPO stages. The goal is twofold: 1) to show that Stage 1 converges on the four shaping rewards, which justifies the plateau-based exit criterion described in Section 3.5.2, and 2) to show that Stage 2 lifts the correctness reward into a positive regime without destabilising the behaviors acquired in Stage 1.

### *4.2.1 Stage 1: Shaping Reward Convergence*

Stage 1 optimises format, completion length, factual grounding, and regulation with equal weights and the correctness reward disabled. Figure 4.1 plots the four reward components alongside training loss and the KL penalty against the frozen reference policy. The four rewards rise from their initial, largely negative values to a clear plateau. The KL trace increases smoothly and settles at a bounded level, which indicates that the policy moves away from the reference only as far as the shaping rewards require, rather than drifting freely.

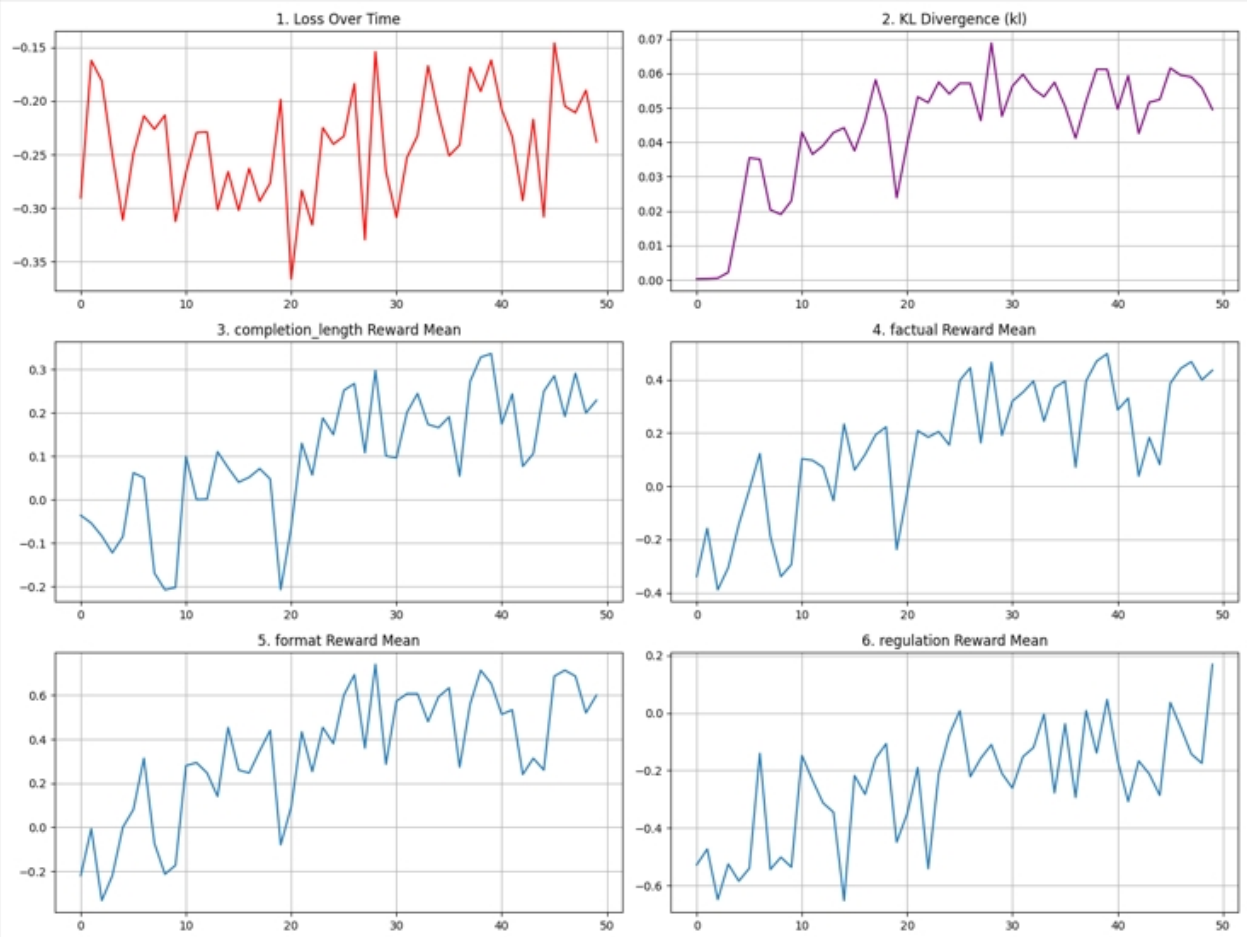


Figure 4.1: Stage 1 training dynamics over 500 optimisation steps. Panels show the four shaping rewards (format, completion length, factual grounding, regulation), training loss, and the KL penalty against the reference policy. All four rewards reach a stable plateau and the KL settles at a bounded level; together these curves define the Stage 1 exit point.

### *4.2.2 Stage 2: correctness-driven optimisation*

Stage 2 re-enables the correctness reward and demotes the four shaping rewards to low regularising weights (Section 3.5.2). The question this stage has to answer is whether the profit signal can be driven upward without the policy losing the output discipline it acquired in Stage 1. Figure 4.2 shows training loss, the KL penalty, the correctness reward, and

regulation reward as a representative shaping component sampled over Stage 2. Correctness rises from approximately zero into a sustained positive band, indicating that the policy is producing strategies that beat the do-nothing baseline more often than not. The regulation panel remains at its Stage-1 plateau, confirming that trading-rule compliance is preserved while profit is being optimized. The KL trace is monitored rather than targeted: it is expected to reach a bounded plateau as the $\beta$ penalty balances reward gain against drift from the reference, and the observed plateau is evidence that this balance is holding rather than as stalled learning on the reward side.

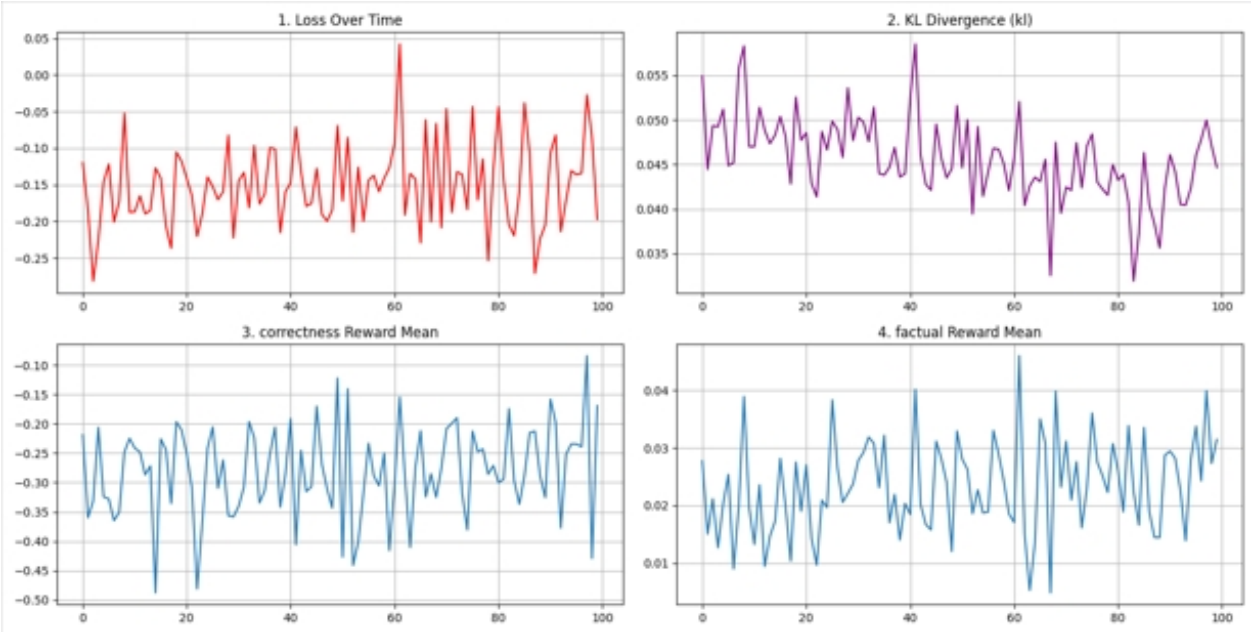


Figure 4.2: Stage 2 training dynamics over 1000 optimisation steps. Panels show training loss, the KL penalty against the reference policy, the correctness reward (dominant signal at weight 0.60), and factual reward as a truth grounding component. Correctness climbs slowly whereas regulation stays at its Stage-1 plateau, indicating profit optimisation

### *4.3 RQ1: Model Comparison*

This section compares the proposed RL-trained Llama against five reference points: its own pre-RL base, an independently selected Qwen base, the same Qwen base after the same two-stage GRPO schedule, the DQN baseline of Craig[6] re-run on our test window, and a passive Buy & Hold portfolio that splits the initial capital evenly across the five tickers. The pairing of each LLM with its own base isolates the contribution of GRPO training, whereas DQN and Buy & Hold anchor the absolute scale of performance against a learned non-LLM agent and a no-skill baseline respectively.

The evaluation window covers 101 trading days from 2025-11-03 to 2026-03-30; the constituent tickers do not share a single market regime over this period. AAPL closes ($-8.2\%$) under its starting price, JPM ($-7.9\%$) below, and GME is essentially flat ($+0.7\%$); only KO ($+13.8\%$) and CAT ($+17.2\%$) trend upward. With three of five names net-negative or flat over the test window, every active strategy is forced to generate alpha against a dominantly bearish-to-sideways tape.

We report four metrics per model: 1) average cumulative return (CR), 2) annualised Sharpe ratio (SR), 3) average maximum drawdown (MDD), and 4) clean rate. Hit rate, used in earlier work that restricts the action space to fixed-size buy/sell, is omitted because our agents emit a continuous quantity, which makes the per-trade win/loss attribution that hit rate relies on ill-defined. The clean rate denotes the fraction of model completions that pass every validation check of Section 3.3.3 and are therefore eligible for execution in the simulator; non-clean completions are rejected and treated as a hold by the environment, so a low clean rate directly caps how often the model can act on its own decisions. Each LLM result is averaged over five independent decoding runs ($k = 5$), and the DQN baseline is averaged over five independent training seeds; Buy & Hold is deterministic. Per-ticker breakdowns are given in Appendix D.

GRPO training improves both base models. Llama-3.2-3B gains 0.74 percentage points of cumulative return ($+1.22\% \rightarrow +1.96\%$) and a one-point reduction in maximum drawdown

Table 4.2: RQ1 model comparison on the five-ticker test window (2025-11-03 to 2026-03-30).

| Model | CR (%) | SR | MDD (%) | Clean (%) |
|---|---|---|---|---|
| Llama-3.2-3B base | +1.22 | −0.28 | 10.22 | 30.6 |
| **Llama-3.2-3B RL (ours)** | **+1.96** | −0.31 | 9.23 | 31.4 |
| Qwen3-4B base | +0.52 | −0.69 | 5.66 | 70.6 |
| Qwen3-4B RL | +0.90 | −0.70 | **2.59** | **98.7** |
| DQN [6] | −7.69 | −1.56 | 13.61 | — |
| Buy & Hold | +3.14 | +0.28 | 13.09 | — |

(10.22% → 9.23%), confirming that the proposed reward system can transfer a small base model from format-only competence to a profit-positive policy. Qwen3-4B benefits in a different profile: cumulative return rises by 0.38 points (+0.52% → +0.90%), maximum drawdown is more than halved (5.66% → 2.59%), and clean rate climbs from 70.6% to 98.7%. This indicates that the regulation and format components of the reward push Qwen toward an almost fully executable policy while the correctness component re-prioritises trades for risk control. The Sharpe ratio is essentially unchanged for both models, which is consistent with the bearish test regime described above: in a market where two of five names lose ground, even a profit-positive policy cannot achieve positive volatility-adjusted return without taking sizing decisions far more aggressive than those allowed by the regulation reward.

The clean-rate gap between the Llama and Qwen families is structural rather than training-induced. Inspection of the rejected Llama completions shows two recurring failure modes that survive GRPO: the model frequently issues buy quantities whose notional exceeds available cash even after the prompt explicitly states the budget, and it rarely chooses hold when the portfolio is already fully invested in the requested ticker. Both behaviours stem from weak arithmetic competence in the 3B base, and although the regulation reward penalises them at training time, fine-tuning alone is unable to teach reliable budget-aware sizing to a model that does not handle the underlying numerical reasoning. The Qwen 4B

base does not exhibit either failure mode at significant rates, which is why its clean rate starts high and saturates to nearly 100% after RL training. This is a known limitation of the proposed framework on small Llama-class backbones rather than a shortcoming of the reward design.

Both LLM families dominate the DQN baseline by a wide margin on every metric. The DQN agent loses 7.69% on average over the test window, has the worst Sharpe ratio of any agent considered ($-1.56$), and the largest drawdown (13.61%). One thing to note is that we did not have the sentiment data that Craig used to improve their model's performance, owing to data availability; the comparison is, however, still enough to deduce that value-based RL on raw price features struggles to generalise to a five-month out-of-sample window of mixed-regime equities.

The Buy & Hold portfolio still outperforms every LLM agent on cumulative return (+3.14% versus the best LLM result of +1.96%). The gap is consistent with the structural constraint of our action space: each trade is bounded by a regulation-shaped sizing reward (Section 3.4.5) that strongly discourages single-ticker concentration, so the LLM portfolio can never take on the full positional exposure that a passive equal-weight buyer takes on day one. In a market where the two upward-trending tickers (KO, CAT) gain more than the two weak tickers lose, this exposure cap directly translates into a return cap. The flip side is risk: maximum drawdown is uniformly lower for the LLM agents (2.59%–10.22%) than for Buy & Hold (13.09%), so the LLMs trade absolute return for downside containment. RQ3 in Section 4.5 isolates the quantity restriction directly by ablating the per-trade sizing constraint, and quantifies how much of the gap to Buy & Hold can be recovered by allowing larger positions.

### *4.4 RQ2: Training-Budget Scaling*

We stop Stage 2 by empirical observation of the reward plateau rather than by a fixed step target. This section tests that decision directly by varying the Stage 2 step budget across three levels while holding every other component of the pipeline fixed. Arm A is trained

for 500 Stage-2 steps ($\approx$ 0.25 epoch over the training split); Arm B for 1000 steps ($\approx$ 0.5 epoch), which corresponds to the policy of Section 4.3; and Arm C for 2000 steps ($\approx$ 1.0 epoch, covering the training split approximately once). All three arms start from the same Stage 1 checkpoint, use the same training corpus and hyperparameters, and are evaluated on the same test window (2025-11-03 to 2026-03-30) with the same decoding settings and five-run averaging protocol.

Table 4.3: RQ2 training-budget comparison on the five-ticker test window.

| Model | CR (%) | SR | MDD (%) | Clean (%) |
|---|---|---|---|---|
| Llama-3.2-3B RL, 500 steps ($\approx$ 0.25 epoch) | 1.33 | $\mathbf{-0.14}$ | 10.17 | 34.3 |
| Llama-3.2-3B RL, 1000 steps ($\approx$ 0.5 epoch) | $\mathbf{+1.96}$ | $-0.31$ | 9.23 | 31.4 |
| Llama-3.2-3B RL, 2000 steps ($\approx$ 1.0 epoch) | $+1.31$ | $-0.16$ | **9.13** | **51.3** |

**Cumulative return.** CR follows an inverted-U trajectory across the three arms: +1.33% (Arm A) $\rightarrow$ +1.96% (Arm B) $\rightarrow$ +1.31% (Arm C). Arm B's peak is only $\sim$ 0.63 percentage points above Arm C and of the same order as the cross-run standard deviation, so no arm is decisively superior on the profit axis. This pattern is consistent with early saturation of the correctness reward: once the policy's excess return reliably sits in the flat region of the tanh normalisation used in Section 3.4.6, the within-group advantage that GRPO relies on collapses, and further steps produce negligible gradient signal on the profit component.

**Output discipline and clean rate.** The clean-rate trajectory is non-monotone, tracing a pronounced V-shape: 34.3% (Arm A) $\rightarrow$ 31.4% (Arm B) $\rightarrow$ 51.3% (Arm C). The initial dip from Arm A to Arm B warrants careful interpretation. As the correctness reward drives CR upward in the first 1000 steps, the policy simultaneously becomes more aggressive: it issues larger position sizes and trades more frequently, which in turn raises the incidence of out-of-budget buy orders and sell orders on positions not currently held. These constraint violations

are penalised by the shaping rewards but their residual error at the Arm B checkpoint is still substantial, so the net effect is a lower clean rate despite higher returns. Beyond 1000 steps, the correctness reward has largely saturated and the gradient is channelled predominantly into the shaping rewards — format compliance, length calibration, factual grounding, and regulation adherence. These objectives have finer residual error at the Arm B termination point, so their cross-rollout differentiation survives into the second half of the run. The result is the dramatic 19.9-point clean-rate recovery observed in Arm C, making the policy substantially more executable on a per-trade basis.

**Risk-adjusted return and drawdown.** The Sharpe ratio and MDD broadly corroborate the clean-rate story. Arm B's Sharpe of $-0.31$ is the worst of the three arms, reflecting the elevated volatility introduced by aggressive constraint-violating trades. Arm A's Sharpe of $-0.14$ is numerically the best, though its clean rate of 34.3% means a substantial fraction of those trades would be rejected in production, limiting the practical value of that figure. Arm C achieves a middle-ground Sharpe of $-0.16$ while also posting the lowest MDD (9.13%), suggesting that the output-discipline gains from the extended budget translate into more stable equity curves even if they do not raise headline returns.

**Summary.** Taken together, the three arms paint a consistent picture of *differential saturation*: the correctness reward saturates within the first $\sim$1000 steps, yielding the peak CR at Arm B, while the shaping rewards require the full 2000-step budget to converge, yielding the peak clean rate and lowest drawdown at Arm C. This validates the early-stop philosophy of Section 3.5.2 for the correctness-driven component of Stage 2: additional compute beyond the reward plateau does not improve profit. It also surfaces a concrete direction for future work — splitting the Stage 2 budget asymmetrically across reward families so that the correctness component can exit as soon as its variance collapses while the shaping rewards continue to tighten the policy's output discipline.

### *4.5 RQ3: Quantity-Aware vs Full-Position Action Space*

Section 4.3 attributed the cumulative-return gap between the quantity-aware policy and the Buy & Hold baseline to the regulation reward's sizing constraint, which prevents any single trade from approaching the full-position exposure that a passive buyer takes on day one. This section tests that attribution directly, by removing quantity from the action space and retraining the policy end-to-end. A pure evaluation-time override was rejected because the quantity-aware model was trained with sizing rewards active, so interpreting its output under a full-position execution rule would mix policy behaviour with a reward-action mismatch. We therefore retrain a counterpart model with the sizing dimension eliminated from every upstream component and compare the two on an otherwise identical protocol.

The retrain changes exactly the pieces that quantity touches and leaves everything else alone. The synthetic portfolio generator of Section 3.2.2 is collapsed to a bimodal state: each training observation either holds a full position in a single ticker with available cash near zero, or holds no position with the full account value in cash, so that the policy always faces an all-in or all-out decision. The system prompt of Section 3.3.2 is stripped of the quantity field and its associated trading rules, leaving the schema with action and take-profit / stop-loss references only. On the reward side, only the sizing terms of Section 3.4.5 are removed; format, completion length, factual grounding, correctness, and the remaining regulation checks (budget legality, ownership, TP/SL trigger consistency, hold semantics) are retained without modification. The trading simulator translates an emitted buy into a spend of all available cash at the current price and an emitted sell into a liquidation of the full held position. Hyperparameters, LoRA configuration, Stage 1/Stage 2 structure, and evaluation window are identical to those in Section 4.3.

Removing the quantity dimension closes the profit gap to Buy & Hold, and does so conclusively. Cumulative return rises from +1.96% to +3.17%, a gain of 1.21 percentage points that more than fills the 1.18-point deficit against Buy & Hold reported in Section 4.3; the retrained policy is now essentially at parity with the passive benchmark on the

Table 4.4: RQ3 action-space comparison on the five-ticker test window.

| Model | CR (%) | SR | MDD (%) | Clean (%) |
|---|---|---|---|---|
| Llama-3.2-3B RL, quantity-aware (4.3) | +1.96 | −0.31 | **9.23** | 31.4 |
| **Llama-3.2-3B RL, no-quantity (ours)** | **+3.17** | **+0.18** | 12.19 | **73.3** |
| Buy & Hold | +3.14 | +0.28 | 13.09 | — |

same five tickers. The Sharpe ratio crosses zero for the first time across all LLM agents considered in this thesis, moving from −0.31 to +0.18, and indicates that the additional return is not merely bought with symmetric volatility. Maximum drawdown does increase from 9.23% to 12.19%, consistent with the larger per-trade exposure and approaching the 13.09% drawdown of Buy & Hold itself. These three movements together reproduce almost exactly the tradeoff that the regulation-reward sizing terms were designed to enforce i.e., capping trade size trades nominal return for downside containment and removing the cap recovers the return at the cost of the containment.

The clean-rate jump from 31.4% to 73.3% is consistent with the failure-mode analysis of Section 4.3, where the dominant cause of rejected completions on the Llama-3.2-3B backbone was the model's difficulty reasoning about budget-aware share counts. The no-quantity schema removes that numerical reasoning requirement by construction: the validator no longer checks a sizing field that the policy no longer emits, so a large class of failures disappears at the schema level rather than through better training. This improvement is therefore best read as structural rather than as evidence that the retrained model has learned more robust output formatting; it still shares a base model with the quantity-aware policy.

Taken together, RQ3 validates the structural interpretation of Section 4.3 and characterises the quantity-aware design as an explicit risk-return tradeoff: the quantity-aware policy delivers roughly one-third of the raw cumulative return in exchange for roughly three percentage points of reduced maximum drawdown and a substantially less aggressive risk

profile, whereas the no-quantity variant recovers full-benchmark return at the cost of larger drawdowns.

## *4.6 Qualitative Analysis*

The aggregate metrics of Sections 4.3–4.5 capture average behaviour but not the interpretability that motivates the LLM approach in the first place. This section reports three representative completions from the Llama-3.2-3B RL policy on the same test window, chosen to illustrate how the model's natural-language reasoning, its structured strategy output, and the closed-loop TP/SL mechanism interact in individual decisions. One completion is selected per outcome type: a profitable buy driven by news-conditioned reasoning (Case A); a stop-loss trigger on a falling held position (Case B); and a coherent but ultimately incorrect buy that is contained by the stop-loss the model itself specified (Case C). Forward five-day returns are reported using the same simulator horizon as the correctness reward of Section 3.4.6.

### *4.6.1 Case A: news-conditioned buy on GME, 2026-01-16*

Listing 4.1: Model output for GME on 2026-01-16 (run $k$=4). Action is buy with an asymmetric target-price / stop-loss envelope around the current price of $21.42. Forward five-day return on the underlying is +13.79%.

```
{
  "reasoning thoughts": "The stock is experiencing a shift in risk profile
      as CEO Ryan Cohen's compensation plan ties his earnings to long-term
      value creation, signaling a focus on execution over passive decline.
      The company benefits from a strong cash position and positive cash
     flow, limiting downside risk. However, the retail sector is facing
     ongoing challenges, and the company's financial health is being
     questioned. The stock has surged over sevenfold under CEO Ryan Cohen,
      raising questions about the sustainability of this growth.",
```

```
  "evidence for decision": "observation.stock information.technique
     indicators.RSI[6]",
  "strategy": {
    "ticker": "GME",
    "action": "buy",
    "number of shares": 5.00,
    "current price": 21.42,
    "target price": 24.0,
    "stop loss price": 21.0
  }
}
```

The reasoning paragraph is structured as a two-sided analysis: a bullish mechanism grounded in a specific corporate event (the CEO compensation plan tying pay to long-term value creation, paired with the company's liquidity position) is weighed against a bearish backdrop (retail-sector headwinds and sustainability concerns after the multi-year run-up). The model arrives at a buy with an asymmetric TP/SL envelope, placing the target roughly twelve percent above the entry and the stop-loss only 2 percent below, which encodes the conviction that the upside mechanism outweighs the downside. The underlying appreciated by 13.79% over the following five trading days, a move that would have crossed the take-profit level the strategy itself specified, confirming the directional call. The small quantity, however, restricts the possible profits, as discussed in Section 4.5. The evidence field, contains only a structural path reference rather than a substantive citation; this is a weakness of the completion independent of its profitability, and the factual-grounding reward of Section 3.4.4 is calibrated precisely to keep penalizing such shallow evidence even when the outcome is favourable.

### *4.6.2 Case B: stop-loss trigger on JPM, 2026-03-06*

Listing 4.2: Model output for JPM on 2026-03-06 (run $k$=4). A held JPM position has its stop-loss set at \$300.0 and the current price of \$293.55 is below that level, so the stop-loss

has triggered and the model sells. Forward five-day return on the underlying is −2.09%.

```
{
  "reasoning thoughts": "The stock is experiencing a downtrend with a
     recent low at 293.55. The MACD is bearish, and the RSI is below 30,
     indicating a possible oversold condition. The EMA_7 and EMA_14 are
     also bearish. Considering the recent news about JPMorgan Chase's
     workforce reduction and legal challenges, the sentiment is bearish.
     With a 2.99% notional (current_account_value x desired_fraction =
     9515.55828 x 0.10 = 951.56) and a current_price of 293.55, the
     number_of_shares = min(sweet_shares, budget_cap_shares) = min(9.55,
     3.24) = 3.24. Since the position is already held, we should trim the
     position to avoid dust. The stop loss price is 300.0, which is above
     the current price. Therefore, we should sell the position.",
  "evidence for decision": "The stock is experiencing a downtrend, and the
      MACD and RSI are bearish. The recent news about JPMorgan Chase's
     workforce reduction and legal challenges also contribute to the
     bearish sentiment.",
  "strategy": {
    "ticker": "JPM",
    "action": "sell",
    "number of shares": 3.24,
    "current price": 293.55,
    "target price": 300.0,
    "stop loss price": 300.0
  }
}
```

This completion exercises the closed-loop risk-management mechanism that motivates the TP/SL fields in the action schema. The held position's stop-loss was set at $300.0, and the current price of $293.55 has crossed below that level; the reasoning paragraph states the trigger condition explicitly ("The stop loss price is 300.0, which is above the current price. Therefore, we should sell the position") and reinforces it with consistent technical and news evidence. The quantitative passage embedded in the reasoning is also notable: the model

reproduces the sizing arithmetic used by the regulation reward of Section 3.4.5, computing a notional target against the account value and then taking the minimum of the sweet-spot share count and the budget cap to derive a final quantity of 3.24 shares. Over the next five trading days, the underlying value fell by an additional 2.09%, so the stop-loss prevented a further loss on the held position and the decision to exit is retroactively vindicated.

### *4.6.3 Case C: coherent but incorrect buy on AAPL, 2026-02-06*

Listing 4.3: Model output for AAPL on 2026-02-06 (run $k$=5). A balanced bullish thesis leads to a buy with an asymmetric TP/SL band. Forward five-day return on the underlying is −7.95%; the self-specified stop-loss of $265.0 (equivalent to −4.6% from entry) would have contained the larger part of the loss on the held position.


```
{
  "reasoning thoughts": "Apple's recent strong earnings report and record
     iPhone sales, along with the company's diversification efforts and
     bullish sentiment, suggest a potential buy opportunity. However, the
     mixed sentiment from institutional investors and concerns over rising
      costs and supply chain challenges may impact the stock's long-term
     growth potential. Considering these factors, I will buy 5.0 shares of
      AAPL.",
  "evidence for decision": "strong earnings report, record iPhone sales,
     and bullish sentiment",
  "strategy": {
    "ticker": "AAPL",
    "action": "buy",
    "number of shares": 5.0,
    "current price": 277.86,
    "target price": 295.0,
    "stop loss price": 265.0
  }
}
```

This case is the mirror image of Case A: the model again builds a balanced view with supportive and cautionary clauses, but this time the market moves against the bullish thesis. The reasoning is not the failure mode. The cited factors (strong earnings, record iPhone sales, diversification) were all true as of the observation date; the "however" clause acknowledges institutional sentiment and supply-chain risk; and the chosen TP/SL band (+6.2% / −4.6%) is a defensible risk-reward structure. What the model cannot see from its observation window is the regime shift that drives AAPL down by 7.95% over the following five trading days, the same regime shift that keeps AAPL's aggregate return negative in the RQ1 table of Section 4.3. The self-specified stop-loss at $265 would have triggered an exit near the −4.6% level, containing most of the drawdown on the held position: the closed-loop mechanism functions even when the entry decision is wrong, which is precisely its purpose.

The three cases together illustrate complementary properties of the trained policy. The model conditions on news and financial context rather than purely on technical indicators (Cases A and C); it articulates sizing and trigger logic that mirrors the reward-side rules of Sections 3.4.5–3.4.6 (Case B); and it couples each entry with a TP/SL envelope that provides downside containment independent of the directional correctness of the entry (Cases A and C). Individual completions also expose failure modes that the aggregate metrics hide, most notably the weak evidence field in Case A, which motivates retaining the factual-grounding reward even after correctness dominates the Stage 2 objective.

## *4.7 Summary of Findings*

Across the three research questions defined in Section 1.2, the experiments produce a consistent picture.

The model-comparison study (Section 4.3) shows that, under a LoRA-based fine-tuning regime, the absolute trading capability of the resulting policy is bounded by the backbone's underlying numeric and reasoning competence. Within that constraint, however, every backbone we evaluated received a measurable uplift in cumulative return over its base counterpart, and both LLM families dominated the DQN baseline on every metric.

The training-budget study (Section 4.4) shows that further training is only useful as long as the corresponding reward has not saturated. On our two-year training span the correctness reward plateaus after roughly half an epoch; extending to a full epoch yields a small CR regression even though clean rate, format, and regulation rewards continue to improve. This per-component saturation profile is exactly what motivates the two-stage schedule of Section 3.5.

The action-space study (Section 4.5) shows that the quantity dimension accounts for a large share of the gap between our reported CR numbers and those of comparable LLM-trader work. Removing the quantity field and reverting to full-position trading expands CR substantially, but at the cost of a deeper maximum drawdown — characterising explicit quantity modelling as a risk-control variable rather than a performance concession.

The qualitative analysis (Section 4.6) reinforces the quantitative findings: the trained policy reasons in concrete, news-conditioned terms, exits positions when its own stop-loss threshold is crossed, and accepts losing trades it could not have foreseen rather than refusing to act.

# Chapter 5

# CONCLUSION AND FUTURE WORK

## *5.1 Conclusion*

This thesis presents a pure reinforcement-learning framework that trains a Large Language Model to produce closed-loop trading strategies from raw multi-source market data, without any supervised cold-start on professional analyst corpora. The empirical study in Chapter 4 supports three contributions to the literature on RL-based LLM trading.

First, pure RL is sufficient to convert a generic instruction-tuned LLM into a competent, rule-respecting trader. Prior LLM-trading systems entangle SFT on expert-curated trading analyses with the RL phase, leaving the contribution of RL alone unmeasured. By showing that GRPO with a multi-component reward, applied directly to off-the-shelf Llama-3.2-3B and Qwen3-4B backbones, produces measurable trading uplift over each base model and a substantial margin over a value-based DQN baseline, this work isolates RL fine-tuning as a stand-alone training signal for LLM trading. On the small backbones we could afford, the dominant design lever proved to be not parameter count but the joint engineering of the reward decomposition, the system prompt, and the synthetic portfolio dataset.

Second, reward components saturate at different rates, and an effective training schedule must be split accordingly. The training-budget study identifies that the path-dependent correctness reward — a tanh squash of baseline-relative excess return — collapses into a flat region long before the structural shaping rewards (format, length, factual grounding, regulation) finish converging. Doubling the Stage-2 step budget on a fixed corpus did not improve cumulative return but did substantially raise clean rate. This differential-saturation result is what justifies the two-stage curriculum at the heart of the framework, and points to asymmetric per-reward training schedules as a concrete next step for RL-fine-tuned LLM

traders.

Third, per-trade quantity is a risk-control variable, not a performance variable. Removing the sizing field from the action space and retraining end-to-end recovered full parity with the Buy & Hold benchmark on cumulative return, at the cost of a deeper maximum drawdown. The quantity-aware policy therefore gives up roughly one-third of the raw cumulative return in exchange for substantially shallower drawdowns and a less aggressive risk profile. This characterises explicit per-trade sizing not as a CR concession but as a path-stability lever, and shows that risk-management constraints normally imposed outside the policy can be folded into the policy itself.

Together, these three contributions support the broader claim that GRPO with a multi-component reward, applied directly to a generic instruction-tuned LLM, is sufficient to produce an interpretable, rule-respecting trading agent — one that reasons in news-conditioned terms, exits positions through its own self-specified TP/SL, and accepts losing trades rather than refusing to act — without any supervised pre-training on a professional analyst corpus.

### *5.2 Limitations*

The binding constraint on this work is iteration cost. A single end-to-end Stage 1 + Stage 2 run (500 + 1000 steps) takes roughly 60 hours on the 24GB A4500 GPU used for development, since the card does not have enough headroom to host both the GRPO trainer and a vLLM rollout server simultaneously. Migrating to a cloud A100-80GB with vLLM brings the same run down to about 9 hours, on which our final reported numbers were collected; everything prior to the final run, however, had to be debugged in the slower regime. This budget bounds two design choices reported in Chapter 4. We did not extend training beyond the correctness-reward saturation point identified in Section 4.4, because a full epoch over the two-year dataset would have roughly doubled wall-clock time without a measurable profit-signal gain. And we capped the ticker universe at five symbols — one per diversity bucket — rather than the originally planned ten (two symbols per bucket). The resulting evaluation is therefore industry-representative across market leader, cyclical, defensive, financial, and

high-volatility regimes, but not statistically representative within any single bucket.

A second limitation concerns the synthetic portfolio and account-state distribution of Section 3.2.2. In the absence of real trader logs, the joint distribution of cash, holdings, and TP/SL state used during training is constructed from first principles rather than calibrated to empirical trader behaviour. We have tried to make this construction as realistic as the available signal allows, but any residual mismatch with the portfolio states the policy actually encounters at sequential evaluation time cannot be ruled out as a small source of bias.

### *5.3 Future Work*

The most immediate extensions are two ablation studies that fell outside the compute budget of Section 5.2. A TP/SL ablation would retrain the policy with the take-profit and stop-loss fields stripped from the dataset and the output schema, isolating their marginal contribution to CR, MDD, and clean rate. A sentiment ablation would compare three input variants — no news, raw news, and the LLM-summarised news of Section 3.2.3 — to quantify the contribution of the sentiment channel and validate the token-economy trade-off it implies.

A second direction is to restructure the system prompt around an explicit chain-of-thought template. Section 4.3 attributed the Llama clean-rate ceiling primarily to budget-aware quantity arithmetic; forcing the model to externalise its intermediate calculations in a fixed order — affordable quantity, projected profit and loss under the proposed TP/SL, a TP/SL trigger check on the held position, and only then a final action — should reduce arithmetic slips and expose a finer-grained surface for the regulation reward.

A third direction is backbone scaling. Verifying whether this trend continues to Llama-3 8B or Qwen2.5-14B would test whether clean rate saturates, whether the correctness reward learns more sample-efficiently, and whether the resulting policies begin to recover the gap to Buy & Hold in trending regimes.

A fourth direction is to reshape the profit reward itself. Replacing our algorithms with a Sortino-style ratio might give the policy a first-class incentive to control negative deviation, plausibly improving Sharpe and MDD without requiring CR gains.

# BIBLIOGRAPHY

[1] Kathryn M. Kaminski and Andrew W. Lo. When do stop-loss rules stop losses? *Journal of Financial Markets*, 18:234–254, 2014.

[2] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Chris Wang. Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance. *ArXiv*, abs/2011.09607, 2020.

[3] Wo Long, Wenxin Zeng, Xiaoyu Zhang, and Ziyao Zhou. Integrating large language models and reinforcement learning for sentiment-driven quantitative trading, 2025.

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[5] John Joseph Murphy. *Technical Analysis of the Futures Markets: A Comprehensive Guide to Trading Methods and Applications*. New York Institute of Finance, 1986.

[6] Craig Rainey and Min Chen. Algorithmic stock trading strategies. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 458–464, 2024.

[7] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[9] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance, 2023.

[10] Yijia Xiao, Edward Sun, Tong Chen, Fang Wu, Di Luo, and Wei Wang. Trading-r1: Financial trading with llm reasoning via reinforcement learning, 2025.

[11] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*, 2023.

[12] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy, 2025.

# Appendix A

# NEWS SUMMARISATION

## *A.1 Summarisation Prompt*

Daily news for each ticker is condensed by a single GPT-4o-mini call. The exact prompt and rendering convention are reproduced below.

```
System: Summarize the following news in 2-4 sentences for a
        trader: key facts, sentiment, and market relevance.

User:   Title: <article 1 title>
        Content: <article 1 body, truncated to 500 chars>

        ---

        Title: <article 2 title>
        Content: <article 2 body, truncated to 500 chars>

        ---
        ...
```

Figure A.1: Prompt template used to summarise a single trading day's news. Per-article body text is truncated at 500 characters and the concatenated user message is truncated at 6000 characters; the response is capped at 200 tokens.

### *A.2 Raw News to Summary Example*

Section 3.2.3 reports that AAPL raw news has mean length of 831 tokens with a maximum of 16,054 tokens, whereas the produced summary has mean 113 tokens and maximum 167 tokens — a roughly 7× mean reduction and 96× worst-case reduction. Figure A.3 shows one trading day's input and output to make the transformation concrete.

```
RAW INPUT (2024-08-13, 12 news in total but 4 are shown):

  Title: Google unveils Pixel 9 phones earlier thanusual as AI
  race with Apple heats up
  Content: MOUNTAIN VIEW, Calif. | Google on Tuesday unveiled
  its next generation of Pixel phones, providing the maker of
  Android software a head start on the next iPhone in the race
  to bring more artificial-intelligence services...

  Title: Stock Of The Day: Is Apple The Alpha Stock?
  Content: Different stocks or sectors lead the market at different
  times. Sometimes, things can be very confusing.

  Title: Google Unveils Gemini AI Assistant, Pixel 9 Phones.
  Takes Aim At Apple IPhone 16 Launch.
  Content: Google stock may get a boost when parent Alphabet
  unveils AI-ready Pixel 9 mobile phones ahead of the debut
  of new Apple iPhone 16 models.

  Title: Berkshire Dumped Apple, Then Why Should You Pay More
  For It And Get Less?
  Content: Why would you bet on Apple (NASDAQ:AAPL) when
  you can get a better reward by paying less elsewhere? Here
  are two stocks that are valued lower than Apple...

  ........................................(rest of 8 news)
```

Figure A.2: Representative transformation: four raw articles for a single AAPL trading day are reduced to a 2-4 sentence summary covering key facts, sentiment, and market relevance.

```
SUMMARY OUTPUT:

  Google has unveiled its Pixel 9 phones ahead of Apple's iPhone 16
  launch, aiming to stay competitive in the growing AI-driven market
  for mobile devices. The early release is seen as a strategic move
  to gain traction as both companies intensify their focus on
  integrating artificial intelligence into their products. This
  development may positively impact Google's stock, while raising
  questions regarding Apple's market position amidst ongoing antitrust
  scrutiny and changing investor sentiment.
```

Figure A.3: Representative transformation: A single day new summary which extracts key information based on the raw news titles and contents

# Appendix B

# TECHNICAL INDICATOR FORMULAS

All four technical indicators in Section 3.2.1 are computed from the daily closing price series $\{P_t\}$.

**Exponential Moving Average (EMA-7, EMA-14).** With smoothing constant $\alpha_N = 2/(N+1)$ and the no-warmup convention used by pandas `ewm(adjust=False)`,

$$\mathrm{EMA}_N(t) = \alpha_N \, P_t + (1 - \alpha_N) \, \mathrm{EMA}_N(t-1), \qquad \mathrm{EMA}_N(0) = P_0. \tag{B.1}$$

We use $N = 7$ for the short-term trend signal and $N = 14$ for the medium-term trend.

**Relative Strength Index (RSI-14).** Let $\Delta_t = P_t - P_{t-1}$ and split into gain and loss:

$$G_t = \max(\Delta_t, \, 0), \qquad L_t = \max(-\Delta_t, \, 0). \tag{B.2}$$

The 14-day simple moving averages $\bar{G}_t$ and $\bar{L}_t$ yield the relative strength $\mathrm{RS}_t = \bar{G}_t / (\bar{L}_t + \epsilon)$ with a small $\epsilon = 10^{-9}$ for numerical stability, and

$$\mathrm{RSI}_t = 100 - \frac{100}{1 + \mathrm{RS}_t}. \tag{B.3}$$

RSI is bounded in $[0, 100]$; values above 70 are conventionally read as overbought and below 30 as oversold.

**MACD (12, 26, 9).** The MACD line is the difference of two EMAs of the closing price, and the signal line is a 9-day EMA of the MACD line:

$$\mathrm{MACD}_t = \mathrm{EMA}_{12}(t) - \mathrm{EMA}_{26}(t), \tag{B.4}$$

$$\mathrm{MACD}_t^{\mathrm{sig}} = \mathrm{EMA}_9(\mathrm{MACD})_t. \tag{B.5}$$

Both the MACD line and the signal line are exposed to the model, allowing it to read the histogram $\text{MACD}_t - \text{MACD}_t^{\text{sig}}$ implicitly.

# Appendix C

# RETAINED FINANCIAL STATEMENT FIELDS

Section 3.2.3 motivated reducing each quarterly filing to a compact subset of fields. Tables C.1 and C.2 list the exact fields that survive preprocessing, where they come from in the raw FMP payload, and which ones are computed by us. "Type = raw" means the field is copied directly (with number formatting); "Type = derived" means the field is computed from one or more raw fields and is included to save the model from doing the arithmetic itself.

Table C.1: Balance-sheet fields retained for each quarterly filing.

| Field name in observation | Source / formula | Type |
|---|---|---|
| date | FMP `date` | raw |
| filingDate | FMP `filingDate` | raw |
| period | FMP `period` | raw |
| cash and cash equivalents | FMP `cashAndCashEquivalents` | raw |
| short term investments | FMP `shortTermInvestments` | raw |
| net receivables | FMP `netReceivables` | raw |
| inventory | FMP `inventory` | raw |
| total current assets | FMP `totalCurrentAssets` | raw |
| total current liabilities | FMP `totalCurrentLiabilities` | raw |
| short term debt | FMP `shortTermDebt` | raw |
| current ratio | totalCurrentAssets / totalCurrentLiabilities | derived |
| quick ratio | (cashAndShortTermInvestments + netReceivables) / totalCurrentLiabilities | derived |
| net debt | totalDebt - cashAndShortTermInvestments | derived |
| debt to equity ratio | totalDebt / totalStockholdersEquity | derived |

Table C.2: Income-statement fields retained for each quarterly filing.

| Field name in observation | Source / formula | Type |
|---|---|---|
| date | FMP `date` | raw |
| filingDate | FMP `filingDate` | raw |
| period | FMP `period` | raw |
| revenue | FMP `revenue` | raw |
| gross profit | FMP `grossProfit` | raw |
| operating income | FMP `operatingIncome` | raw |
| net income | FMP `netIncome` | raw |
| eps diluted | FMP `epsDiluted` | raw |
| gross margin | grossProfit / revenue | derived |
| operating margin | operatingIncome / revenue | derived |
| net margin | netIncome / revenue | derived |

# Appendix D

# PER-TICKER RQ1 RESULTS

This appendix expands the aggregate metrics of Section 4.3 into per-ticker, per-run breakdowns. Tables D.1, D.2, D.3, and D.4 cover the four LLM-based policies; each table reports five evaluation runs ($k = 1, \ldots, 5$) for every ticker, plus a per-ticker mean across runs. Table D.5 reports the Craig DQN baseline under the same evaluation configuration. All cumulative returns and clean rates are reported as percent; maximum drawdown is reported as a positive magnitude regardless of the underlying sign convention used in the source files.

Table D.1: Per-ticker, per-run RQ1 metrics for Llama-3.2-3B (RL).

| Ticker | Run | CR (%) | SR | MDD (%) | Clean (%) |
|---|---|---|---|---|---|
| AAPL | k=1 | -8.27 | -1.27 | 13.07 | 17.82 |
| | k=2 | -0.33 | -0.14 | 10.95 | 30.69 |
| | k=3 | -4.40 | -1.17 | 9.45 | 30.69 |
| | k=4 | -7.71 | -1.13 | 12.89 | 16.83 |
| | k=5 | -6.80 | -1.08 | 13.78 | 17.82 |
| | **mean** | **-5.50** | **-0.96** | **12.03** | **22.77** |
| CAT | k=1 | +16.54 | +1.30 | 13.18 | 14.85 |
| | k=2 | +16.59 | +1.31 | 13.70 | 14.85 |
| | k=3 | +25.07 | +2.14 | 9.76 | 31.68 |
| | k=4 | +18.15 | +1.36 | 13.86 | 12.87 |
| | k=5 | +17.17 | +1.34 | 13.01 | 3.96 |
| | **mean** | **+18.70** | **+1.49** | **12.70** | **15.64** |
| KO | k=1 | +9.05 | +1.42 | 7.71 | 31.68 |
| | k=2 | +4.80 | +1.64 | 1.86 | 51.49 |
| | k=3 | +4.12 | +0.83 | 5.50 | 62.38 |
| | k=4 | +8.72 | +1.31 | 7.80 | 31.68 |
| | k=5 | -1.31 | -0.55 | 7.55 | 48.51 |
| | **mean** | **+5.08** | **+0.93** | **6.08** | **45.15** |
| JPM | k=1 | -7.84 | -1.05 | 14.09 | 14.85 |
| | k=2 | -5.73 | -0.79 | 11.57 | 31.68 |
| | k=3 | -6.64 | -0.99 | 13.34 | 28.71 |
| | k=4 | -7.73 | -0.94 | 13.11 | 21.78 |
| | k=5 | -8.94 | -1.15 | 11.60 | 27.72 |
| | **mean** | **-7.38** | **-0.98** | **12.74** | **24.95** |
| GME | k=1 | -0.89 | -2.08 | 1.68 | 33.66 |
| | k=2 | -1.18 | -4.46 | 1.22 | 35.64 |
| | k=3 | +0.36 | -0.54 | 1.89 | 59.41 |
| | k=4 | -2.15 | -1.49 | 4.24 | 63.37 |
| | k=5 | -1.75 | -1.49 | 3.84 | 49.50 |
| | **mean** | **-1.12** | **-2.01** | **2.57** | **48.32** |

Table D.2: Per-ticker, per-run RQ1 metrics for Llama-3.2-3B (base).

| Ticker | Run | CR (%) | SR | MDD (%) | Clean (%) |
|---|---|---|---|---|---|
| AAPL | k=1 | -8.14 | -1.19 | 12.70 | 14.85 |
| | k=2 | -13.94 | -2.48 | 18.55 | 34.65 |
| | k=3 | -4.83 | -0.91 | 13.66 | 24.75 |
| | k=4 | -9.52 | -1.35 | 13.61 | 10.89 |
| | k=5 | -8.86 | -1.25 | 13.97 | 15.84 |
| | **mean** | **-9.06** | **-1.43** | **14.50** | **20.20** |
| CAT | k=1 | +17.40 | +1.34 | 13.53 | 9.90 |
| | k=2 | +16.83 | +1.40 | 13.24 | 11.88 |
| | k=3 | +7.27 | +0.71 | 12.13 | 39.60 |
| | k=4 | +18.66 | +1.38 | 13.62 | 7.92 |
| | k=5 | +15.62 | +1.23 | 13.55 | 6.93 |
| | **mean** | **+15.15** | **+1.21** | **13.22** | **15.25** |
| KO | k=1 | +8.81 | +1.45 | 7.76 | 35.64 |
| | k=2 | +0.36 | -0.21 | 4.83 | 60.40 |
| | k=3 | +7.51 | +1.52 | 5.45 | 47.52 |
| | k=4 | +8.75 | +1.36 | 7.72 | 40.59 |
| | k=5 | +12.30 | +2.18 | 7.69 | 38.61 |
| | **mean** | **+7.55** | **+1.26** | **6.69** | **44.55** |
| JPM | k=1 | -9.71 | -1.31 | 15.33 | 18.81 |
| | k=2 | +0.05 | -0.06 | 10.81 | 22.77 |
| | k=3 | -6.58 | -1.63 | 11.38 | 53.47 |
| | k=4 | -7.72 | -0.91 | 15.37 | 13.86 |
| | k=5 | -6.75 | -0.91 | 12.81 | 23.76 |
| | **mean** | **-6.14** | **-0.96** | **13.14** | **26.53** |
| GME | k=1 | -1.93 | -1.47 | 3.49 | 46.53 |
| | k=2 | +2.03 | +0.31 | 4.08 | 49.50 |
| | k=3 | -3.71 | -3.14 | 3.96 | 38.61 |
| | k=4 | -2.58 | -1.59 | 4.26 | 53.47 |
| | k=5 | -0.73 | -1.38 | 1.93 | 43.56 |
| | **mean** | **-1.38** | **-1.45** | **3.54** | **46.34** |

Table D.3: Per-ticker, per-run RQ1 metrics for Qwen3-4B (RL).

| Ticker | Run | CR (%) | SR | MDD (%) | Clean (%) |
|---|---|---|---|---|---|
| AAPL | k=1 | +0.32 | -0.60 | 1.74 | 99.01 |
| | k=2 | -0.92 | -1.89 | 2.11 | 99.01 |
| | k=3 | +1.51 | +0.21 | 2.65 | 96.04 |
| | k=4 | +0.16 | -0.70 | 2.68 | 100.00 |
| | k=5 | -2.04 | -3.34 | 2.67 | 96.04 |
| | **mean** | **-0.19** | **-1.26** | **2.37** | **98.02** |
| CAT | k=1 | +8.79 | +1.21 | 6.53 | 97.03 |
| | k=2 | -4.15 | -0.83 | 9.21 | 96.04 |
| | k=3 | +13.66 | +2.19 | 4.08 | 99.01 |
| | k=4 | +2.43 | +0.29 | 8.18 | 97.03 |
| | k=5 | -0.30 | -0.19 | 6.29 | 96.04 |
| | **mean** | **+4.09** | **+0.54** | **6.86** | **97.03** |
| KO | k=1 | +0.73 | -1.22 | 0.33 | 100.00 |
| | k=2 | +1.06 | -0.22 | 0.36 | 100.00 |
| | k=3 | +1.24 | +0.13 | 0.63 | 100.00 |
| | k=4 | +1.44 | +0.55 | 0.39 | 100.00 |
| | k=5 | +0.19 | -1.93 | 0.65 | 100.00 |
| | **mean** | **+0.93** | **-0.54** | **0.47** | **100.00** |
| JPM | k=1 | -1.09 | -1.87 | 1.67 | 100.00 |
| | k=2 | -0.92 | -0.91 | 3.61 | 99.01 |
| | k=3 | -0.69 | -1.63 | 1.42 | 99.01 |
| | k=4 | -3.30 | -2.09 | 3.92 | 99.01 |
| | k=5 | +1.67 | +0.29 | 1.53 | 100.00 |
| | **mean** | **-0.87** | **-1.24** | **2.43** | **99.41** |
| GME | k=1 | +0.88 | -0.55 | 0.61 | 99.01 |
| | k=2 | -1.59 | -2.18 | 2.20 | 100.00 |
| | k=3 | +0.07 | -2.12 | 0.57 | 97.03 |
| | k=4 | +0.41 | -1.61 | 0.30 | 100.00 |
| | k=5 | +2.96 | +1.44 | 0.39 | 100.00 |
| | **mean** | **+0.55** | **-1.00** | **0.81** | **99.21** |

Table D.4: Per-ticker, per-run RQ1 metrics for Qwen3-4B (base).

| Ticker | Run | CR (%) | SR | MDD (%) | Clean (%) |
|---|---|---|---|---|---|
| AAPL | k=1 | -0.73 | -2.13 | 2.16 | 99.01 |
| | k=2 | -1.83 | -2.97 | 2.13 | 100.00 |
| | k=3 | -3.76 | -3.47 | 3.95 | 100.00 |
| | k=4 | -7.71 | -1.13 | 12.89 | 16.83 |
| | k=5 | -6.80 | -1.08 | 13.78 | 17.82 |
| | **mean** | **-4.17** | **-2.16** | **6.98** | **66.73** |
| CAT | k=1 | +4.82 | +0.72 | 6.88 | 95.05 |
| | k=2 | +12.38 | +1.61 | 4.23 | 96.04 |
| | k=3 | -11.31 | -1.56 | 11.66 | 88.12 |
| | k=4 | +18.15 | +1.36 | 13.86 | 12.87 |
| | k=5 | +17.17 | +1.34 | 13.01 | 3.96 |
| | **mean** | **+8.24** | **+0.69** | **9.93** | **59.21** |
| KO | k=1 | +0.49 | -1.64 | 0.72 | 100.00 |
| | k=2 | +0.84 | -0.75 | 0.45 | 97.03 |
| | k=3 | +1.30 | +0.29 | 0.30 | 99.01 |
| | k=4 | +8.72 | +1.31 | 7.80 | 31.68 |
| | k=5 | -1.31 | -0.55 | 7.55 | 48.51 |
| | **mean** | **+2.01** | **-0.27** | **3.37** | **75.25** |
| JPM | k=1 | +0.24 | -0.89 | 1.58 | 100.00 |
| | k=2 | -0.39 | -1.13 | 1.55 | 99.01 |
| | k=3 | -0.74 | -0.99 | 2.65 | 99.01 |
| | k=4 | -7.73 | -0.94 | 13.11 | 21.78 |
| | k=5 | -8.94 | -1.15 | 11.60 | 27.72 |
| | **mean** | **-3.51** | **-1.02** | **6.10** | **69.50** |
| GME | k=1 | +2.09 | +0.83 | 0.64 | 98.02 |
| | k=2 | +0.46 | -1.62 | 0.39 | 100.00 |
| | k=3 | +1.41 | +0.27 | 0.56 | 100.00 |
| | k=4 | -2.15 | -1.49 | 4.24 | 63.37 |
| | k=5 | -1.75 | -1.49 | 3.84 | 49.50 |
| | **mean** | **+0.01** | **-0.70** | **1.94** | **82.18** |

Table D.5: Per-ticker, per-run RQ1 metrics for the Craig DQN baseline

| Ticker | Run | CR (%) | SR | MDD (%) |
|---|---|---|---|---|
| AAPL | k=1 | -4.87 | -6.58 | 4.87 |
| | k=2 | -0.55 | -0.50 | 4.52 |
| | k=3 | -1.36 | -0.89 | 5.27 |
| | k=4 | +0.97 | +0.38 | 2.56 |
| | k=5 | -0.77 | -0.52 | 5.54 |
| | **mean** | **-1.32** | **-1.62** | **4.55** |
| CAT | k=1 | +5.05 | +0.56 | 15.90 |
| | k=2 | -1.07 | -0.10 | 8.72 |
| | k=3 | -7.32 | -0.80 | 14.80 |
| | k=4 | +5.89 | +0.68 | 10.94 |
| | k=5 | -17.78 | -2.73 | 19.90 |
| | **mean** | **-3.05** | **-0.48** | **14.05** |
| KO | k=1 | -7.47 | -2.00 | 10.60 |
| | k=2 | -3.95 | -0.97 | 11.56 |
| | k=3 | +3.18 | +0.62 | 6.60 |
| | k=4 | -3.53 | -1.22 | 8.02 |
| | k=5 | -2.89 | -0.97 | 6.15 |
| | **mean** | **-2.93** | **-0.91** | **8.58** |
| JPM | k=1 | -20.95 | -3.35 | 23.63 |
| | k=2 | -19.17 | -3.90 | 19.49 |
| | k=3 | -32.60 | -5.86 | 32.60 |
| | k=4 | -19.78 | -3.55 | 20.73 |
| | k=5 | -21.05 | -3.06 | 21.91 |
| | **mean** | **-22.71** | **-3.94** | **23.67** |
| GME | k=1 | -18.73 | -2.01 | 25.22 |
| | k=2 | -11.53 | -1.19 | 14.90 |
| | k=3 | -6.54 | -0.51 | 17.75 |
| | k=4 | -0.16 | -0.03 | 11.26 |
| | k=5 | -5.34 | -0.48 | 16.78 |
| | **mean** | **-8.46** | **-0.84** | **17.18** |