electronics

MDPI

Article

When Memory Is Not Enough: What Drives Robustness in AI Decision Systems? Evidence from Sequential Financial Decision-Making

Jarosław A. Chudziak * and Feliks Bańka

The Institute of Computer Science, Faculty of Electronics and Information Technology, Warsaw University of Technology, 00-661 Warsaw, Poland; feliks.banka.stud@pw.edu.pl
* Correspondence: jaroslaw.chudziak@pw.edu.pl

Abstract

AI systems are increasingly used to support repeated decisions in environments where conditions change over time. Many of these systems combine large language models, retrieved experience, and rule-based execution layers, but their behavior is often evaluated as a single end-to-end system. As a result, when such systems become more stable or risk-aware, it remains unclear whether the improvement comes from model reasoning, relevant memory, stochasticity, or the surrounding control logic. This paper investigates this question in a sequential financial decision setting using Apple, Tesla, and NVIDIA over 2021–2023, where memory access, coordinator choice, retrieval quality, stochasticity, and execution-module configurations are examined through targeted comparisons. The analysis suggests that memory can provide useful context but does not by itself produce robust behavior under adverse or shifting conditions. The full configuration exhibits stronger risk-adjusted and downside behavior than configurations that remove memory or use retrieved experience without model-based synthesis. The strongest observed robustness occurs when relevant memory is combined with bounded model-based coordination and constrained execution. Within this financial testbed, the findings suggest that retrieval relevance, bounded model-based coordination, and execution constraints warrant joint evaluation rather than attributing observed robustness to coordinator choice alone.

Keywords: modular decision systems; large language models; retrieval-augmented systems; sequential decision-making; robustness; episodic memory

Check for updates

Academic Editor: Zihao Lei

Received: 21 August 2026
Revised: 3 November 2026
Accepted: 4 November 2026
Published: 8 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

AI systems are increasingly used to support repeated decisions in environments where conditions change over time [1,2]. Rather than relying only on a single predictive model, many modern systems combine large language model (LLM)-based synthesis, retrieved experience, deterministic execution components, and explicit safety or feasibility constraints [3,4]. This systems-level design is attractive because it can combine heterogeneous information, expose intermediate decisions, and keep final actions within bounded rules [5–7]. It also offers a practical alternative to fully end-to-end learning in settings where reliability and auditability matter [8,9].

Many such systems are evaluated primarily end-to-end [10,11]. When behavior becomes more stable, less risky, or more robust in shifting environments, the source of that improvement is often unclear [1,12]. It may come from model choice, better reasoning,

retrieved memory, reduced stochasticity, constrained execution, or the way these mechanisms are configured together [13–15]. In LLM-supported systems, this attribution problem is especially important because observed performance can be too easily assigned to the coordinating model, even when much of the behavior may be shaped by memory and control structure [3,11,12].

This leads to the central question of the paper: what drives robustness in AI decision systems that combine memory, model-based coordination, and constrained execution? More specifically, when such systems become more stable under shifting conditions, how much of the effect comes from retrieved episodes, how much depends on the coordinating model, and how much is produced by deterministic control and execution constraints? We address this question through a financial sequential-decision setting in which the coordinator, memory retrieval, stochasticity, and execution constraints are examined through targeted configuration changes [2,12,15]. From a systems perspective, this is a control-design question: robustness may depend not only on the predictive or reasoning component, but also on how information, coordination, and admissible actions are structured around it [5,6,14].

We study this question in a financial sequential-decision setting. This domain is useful empirically because it involves partial observability, changing regimes, delayed outcomes, transaction frictions, and measurable downside behavior [16,17]. The main evaluation uses Apple (AAPL), Tesla (TSLA), and NVIDIA (NVDA), three highly liquid large-cap equities with actively traded options and heterogeneous volatility profiles over 2021–2023. Together, they provide a compact testbed spanning distinct volatility and regime characteristics while preserving comparable execution conditions. Consistent with broader work on utility-based and risk-aware decision-making [18], the analysis focuses on risk-adjusted behavior, drawdown, tail risk, and stability under perturbation.

This article directly extends the MultiHedge architecture introduced in our prior work [19]. Whereas that study presented the retrieval-augmented modular control system and evaluated its end-to-end behavior, the present work shifts the unit of analysis to robustness attribution. We separate the effects of retrieved experience, coordinator choice, stochasticity, retrieval quality, and constrained execution through targeted component comparisons. The experiments therefore examine how memory, model-based coordination, and constrained execution are associated with system behavior under a common evaluation setting. Code and experimental configuration files are publicly available at https://github.com/latent-systems-lab/Modular-Decision-Systems-for-Robustness (accessed on 3 November 2026).

Figure 1 summarizes the research design and the system components examined through the targeted comparisons.

This study examines robustness attribution in AI-supported sequential decision-making, with particular attention to retrieved memory, model-based coordination, stochasticity, and constrained execution. Its experimental design compares full coordination with no-memory, retrieval-only, random-retrieval, lightweight-coordinator, stochastic, and execution-module variants. The resulting evidence distinguishes the contextual value of retrieved experience from the stronger downside control observed when relevant memory is combined with bounded model-based synthesis and explicit execution constraints.

The remainder of the article is organized as follows. Section 2 reviews related work on LLM-based decision systems, retrieval memory, and robustness in sequential decision-making. Section 3 describes the decision system and component-isolation strategy. Section 4 presents the main ablations, retrieval-isolation analysis, baseline comparisons, robustness checks, and efficiency analysis. Section 5 discusses implications and limitations. Section 6 concludes.

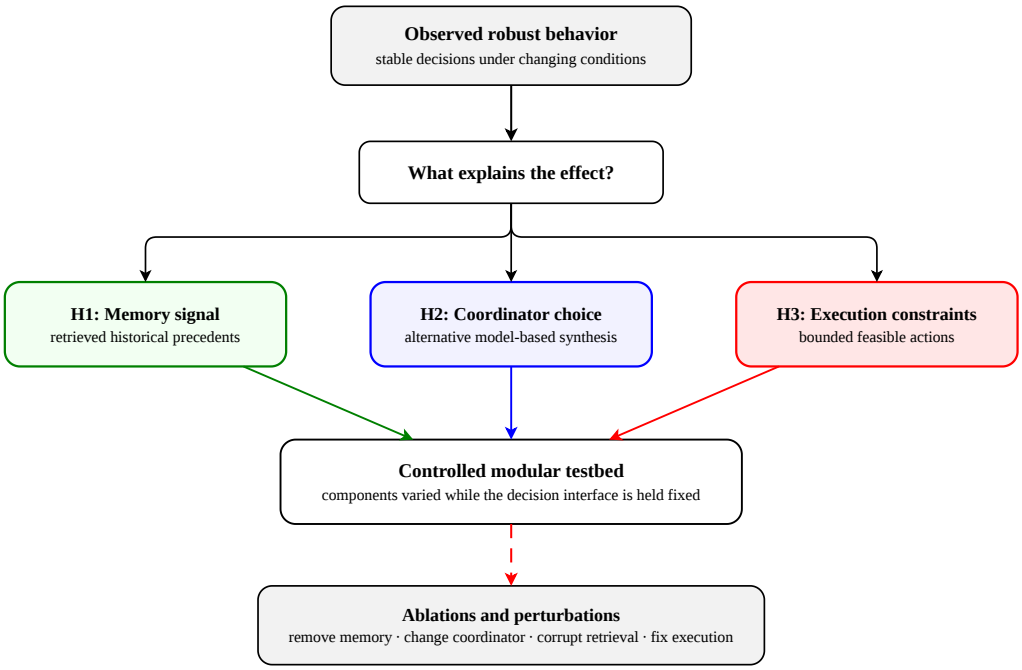


Figure 1. Conceptual framing of the study. The paper asks what drives robust behavior in AI decision systems that combine retrieved memory, model-based coordination, and constrained execution. The empirical analysis varies memory access, retrieval quality, coordinator choice, stochasticity, and execution modules to distinguish memory effects from coordination and control effects. Colors distinguish the three hypotheses; arrows show the conceptual flow from these hypotheses to the testbed, with the dashed arrow indicating the subsequent ablation and perturbation stage.

2. Related Work

This paper connects three strands of work: LLM-based AI decision systems, memory and retrieval mechanisms, and robustness in sequential decision-making [1,13,20]. The closest systems differ in which of these components they emphasize, making component-level attribution difficult when performance is reported only end-to-end.

2.1. LLM-Based AI Decision Systems

Large language models are increasingly embedded as reasoning or synthesis components inside larger decision systems [1,3]. In these settings, the model does not necessarily act as an end-to-end policy; instead, it processes heterogeneous inputs, summarizes context, invokes tools, or produces structured intermediate decisions [2,11]. In finance, FinBERT focuses on domain-specific language representations for financial text [21], BloombergGPT develops a finance-oriented language model [22], and FinGPT explores data-centric financial LLM adaptation and was presented at the FinLLM Symposium at IJCAI 2023 [23]. FinAgent moves closer to the present setting by using an LLM-centered architecture for financial decision-making [11]. These works demonstrate the value of language models in financial pipelines, but they do not by themselves isolate how retrieved episodic experience, coordinator choice, and downstream execution constraints contribute to stability.

Recent work also studies multi-agent and modular LLM systems, where specialized components interact to perform complex tasks [1,4]. These systems are aligned with earlier work on modular and multi-agent decision systems because they separate sensing, reasoning, coordination, and execution [6,7]. However, reported gains are often discussed in terms of model scale, prompting, or reasoning capability, while the surrounding system structure receives less explicit attention [10,11]. This leaves open whether improved stability comes

from the LLM itself or from the way the LLM is bounded by memory, tools, and execution rules [1,12,15].

2.2. Memory and Retrieval in AI Systems

Retrieval-augmented generation conditions model outputs on external information, such as documents, examples, or retrieved past cases [12,20,24]. This idea is closely related to case-based reasoning, where decisions are informed by similar prior situations [25]. In sequential settings, retrieval can be interpreted as a non-parametric memory over past state-action-outcome trajectories, complementing learned representations without requiring immediate retraining [4,26].

Retrieval mechanisms have also been used to incorporate historical context into forecasting and decision systems [27]. Knowledge-graph-supported decision support [2] and clinical retrieval-augmented workflows [28] illustrate how retrieval can become one component of a broader decision system rather than only a question-answering aid. In much of this work, retrieval is evaluated as an enhancement to a larger model rather than as a mechanism that can be isolated [20,24]. The present study therefore distinguishes the stored episodic memory buffer from the state-dependent retrieved neighborhood and from the retrieved context presented to the coordinator.

2.3. Robustness and Control in Sequential Decision-Making

Robustness in sequential decision-making is a long-standing concern in reinforcement learning and control theory, where systems must act under partial observability, delayed feedback, non-stationarity, and distribution shift [13,14]. In financial decision-making, these issues appear as regime shifts, volatility clustering, transaction frictions, and heavy-tailed downside risk [16,17]. These properties make the domain useful for evaluating whether an AI-supported decision system remains stable when conditions change [8,29].

Deep reinforcement learning approaches learn adaptive policies for such environments [8,30], including distributional or risk-sensitive methods for limiting heavy losses [29,31]. Multi-agent and ensemble methods can improve stability by combining specialized components [32–34], but they often rely on learned coordination mechanisms and do not isolate the separate role of retrieved memory [35]. More broadly, modular architectures can improve reliability by separating components and constraining their interaction [5,6]. The present study addresses this gap by examining the contributions of memory, model-based coordination, stochasticity, and constrained execution through controlled component comparisons.

3. Decision System Architecture and Component Isolation

This section describes the decision system used to isolate the sources of stable behavior. The underlying modular architecture is inherited from MultiHedge [19]; the extension developed here is the component-isolation framework used to analyze its robustness mechanisms. Memory, model-based coordination, stochasticity, and execution constraints are separated so that their contributions can be compared. Figure 2 illustrates the separation between retrieved memory, bounded coordination, and constrained execution.

3.1. Problem Formulation

We model the empirical domain as a sequential control problem under partial observability, where the system must repeatedly choose bounded actions under regime variation and transaction frictions. The financial variables make robustness observable through drawdown and tail-risk metrics, but the analytic role of the setting is to expose repeated decision behavior under uncertainty.

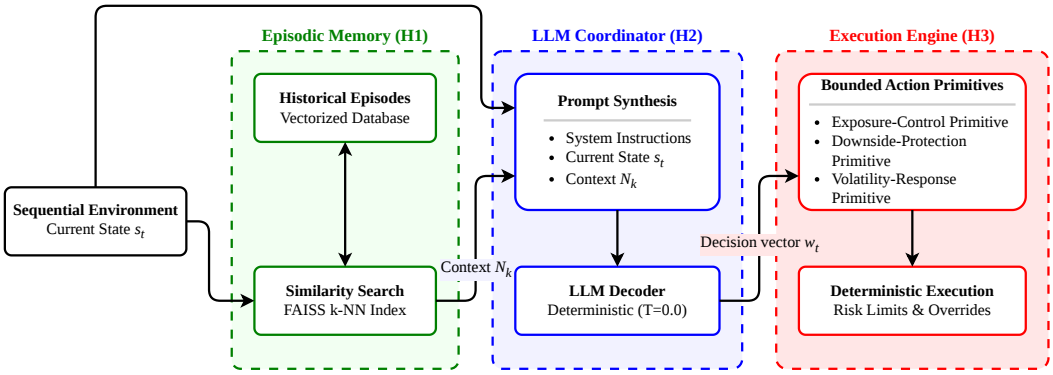


Figure 2. Decision system used for component isolation. The architecture separates retrieved episodic memory, bounded model-based coordination, and deterministic execution through fixed action primitives. In the financial implementation, these generic primitives correspond to concrete hedging modules, but the diagram emphasizes their role as bounded action operators. This separation supports targeted comparisons of memory access, retrieval quality, coordinator choice, stochasticity, and execution constraints in the analyses reported in Section 4.

Formally, the setting is represented as a Partially Observable Markov Decision Process (POMDP) [13]:

$$\langle \mathcal{S}, \mathcal{A}, O, \mathcal{T}, \mathcal{R} \rangle,$$

where $\mathcal{S}$ denotes the latent market–portfolio state space, $\mathcal{A}$ the mixed action space, O the observable inputs, $\mathcal{T}$ the transition dynamics, and $\mathcal{R}$ the reward functional.

The POMDP framing is used here to motivate retrieval rather than to introduce a separate planning algorithm. Because the true state and regime are only partially observed through O, retrieved episodes provide an empirical context variable that can help condition the coordinator on prior situations that resemble the current observation [4,26].

To account for both market information and internal portfolio conditions, we define the controller state representation at time t as

$$s_t = [V_t, \mathbf{S}_t, \boldsymbol{\sigma}_t, \mathbf{X}_t, \mathbf{O}_t, Z_t, B_t].$$

Here, V_t is the total portfolio value, $\mathbf{S}_t$ collects price-based market descriptors, $\boldsymbol{\sigma}_t$ summarizes volatility-related features, $\mathbf{X}_t$ denotes current equity exposures, and $\mathbf{O}_t$ captures existing option positions. The variable Z_t summarizes regime-related descriptors available to the controller, while B_t encodes safety and constraint-related indicators. This state representation combines external signals with internal portfolio status, which is important because the later experiments vary memory and control logic while keeping the state interface fixed.

The objective is not defined purely in terms of gain. Instead, the generic reward combines return, risk, benchmark-relative performance, and execution costs:

$$r_t = 0.5R_t - 0.3Q_t + 0.1B_t - 0.1C_t,$$

where R_t is the realized-return component, Q_t is the risk penalty, B_t captures performance relative to the benchmark, and C_t aggregates execution costs, including transaction costs and turnover-related penalties. The weights correspond to `return_weight` $= 0.5$, `risk_weight` $= 0.3$, `benchmark_weight` $= 0.1$, and `cost_weight` $= 0.1$, and are held fixed across the compared configurations. The reward coefficients are fixed, representative design choices used consistently across the reported experiments; they are not presented as optimized values, and coefficient tuning is outside the scope of this study. This objective makes explicit that the empirical setting balances return with risk, benchmark-relative behavior, and trading intensity rather than raw gain alone.

3.2. Coordination and Modular Decision Flow

The system separates decision-making into distinct stages so that the comparative roles of memory, coordination, and execution can be examined through targeted component ablations [19]. The LLM-based coordinator has a bounded role: it synthesizes state information and retrieved context into structured allocation weights, but it does not execute unconstrained free-form actions. The decision flow is written as

$$g_t = \pi_{\text{regime}}(s_t), \tag{1}$$

$$w_t = \pi_{\text{alloc}}(s_t, g_t, E_t), \tag{2}$$

$$a_{i,t}^H = \pi_{\text{agent}}^H(s_t, w_t), \ \forall H \in \{\text{collar, straddle, delta}\}. \tag{3}$$

In the first stage, π_{regime} produces a structured regime summary g_t. In the second stage, π_{alloc} generates allocation weights conditioned on the current state representation, the inferred regime summary, and the retrieved episodic context E_t. In the third stage, each execution module H transforms the resulting allocation into executable action parameters.

The final portfolio action is obtained by weighted aggregation of the outputs of the available modules. Let $\delta_{i,t}^H \in [0, 1]$ denote the weight assigned to module H, with

$$\sum_H \delta_{i,t}^H \leq 1.$$

Any residual mass, $1 - \sum_H \delta_{i,t}^H$, is interpreted as an unhedged or cash-preserving allocation. The aggregated action then takes the form

$$a_{i,t}^{\text{final}} = \sum_H \delta_{i,t}^H \cdot a_{i,t}^H.$$

This formulation is important for the later analysis because it ensures that the system composes decisions from interpretable primitives instead of generating unconstrained free-form actions. The controller therefore operates over a bounded mixture of modules, which makes it possible to compare full coordination against simpler or partially disabled variants.

3.3. Episodic Memory and Retrieval Logic

The memory component is treated here as an explicit non-parametric mechanism rather than as an implicit property of the language model [20,25]. Recent work on LLM agents similarly treats episodic memory as a distinct design element, including both world-model construction and safety analysis of memory-related risks [15,26]. Defining memory independently of the coordinating model enables its contribution to be evaluated through direct component comparisons.

Each past episode stored in memory is represented as a tuple

$$e_j = (s_j, w_j, a_j, r_j, s_{j+1}),$$

where s_j is the observed state, w_j the historical allocation weights, a_j the executed action, r_j the realized reward, and s_{j+1} the subsequent state. Episodes are formed from completed transitions: r_j and s_{j+1} are available only after the corresponding outcome has been observed, so an episode cannot inform the decision that generated it. The episodic memory buffer is then

$$\mathcal{M} = \{e_1, e_2, \ldots, e_N\}.$$

This representation is deliberately more structured than plain textual memory: each stored element links a state, an allocation, a decision, and an observed outcome, which allows memory to function as reusable experience rather than as unstructured context.

To retrieve relevant experience, each current state is mapped into an embedding space through a representation function $\phi(\cdot)$. The controller then searches for the most relevant

stored episodes in an embedding space in which vectors are L2-normalized, so that exact nearest-neighbor search with an L2 index is equivalent to cosine-based ranking. In compact form, the retrieved neighborhood is

$$\mathcal{N}_k(s_t) = \mathrm{TopK}_{e_j \in \mathcal{M}} \ \mathrm{sim}\Big(\phi(s_t), \phi(s_j)\Big),$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes similarity induced by nearest-neighbor ranking in the normalized embedding space and $\mathcal{N}_k(s_t)$ returns the top-k nearest episodes. We use the terms *episodic memory buffer* exclusively for the full store $\mathcal{M}$, *retrieved neighborhood* for $\mathcal{N}_k(s_t)$, and *retrieved context* for the representation E_t passed to the coordinator.

This retrieval mechanism matters analytically because it can later be studied in its own right. In Section 4.3, we use the same formal memory definition to construct a retrieval-only controller and to run memory corruption tests. The present formulation therefore makes memory a directly testable object rather than a hidden implementation detail of the broader system.

3.4. Stability Constraints and Execution Layer

A key design choice is the functional decoupling of decision synthesis and action execution. Coordinator outputs are generated using a structured-output mode with a fixed schema, restricting the LLM to machine-parseable allocation weights rather than free-form instructions. These weights (w_t) are then passed to the execution engine, where specialized agents perform deterministic feasibility and execution checks.

In the financial implementation, the execution layer instantiates bounded action primitives through collar, straddle, and delta-neutral modules, following earlier work on multi-agent option-optimization systems [36]. These modules are autonomous in their contract-level implementation; they transform high-level allocation targets into specific trades by screening for market availability, liquidity, and maturity. Crucially, the execution layer does not "learn" hedging logic; it maps the coordinator's mixing coefficients into discrete positions based on fixed payoff profiles. For example, the terminal payoff Π of the straddle agent, applied to an underlying asset with price S_T at expiration and strike K, is defined as:

$$\Pi_{\text{straddle}}(S_T) = \max(S_T - K, 0) + \max(K - S_T, 0) - C_0 - P_0, \tag{4}$$

where C_0 and P_0 are the initial premiums. Similarly, the collar agent bounds downside risk by financing a protective put through a covered call, yielding a capped payoff:

$$\begin{aligned} \Pi_{\text{collar}}(S_T) = S_T &+ \max(K_{\text{put}} - S_T, 0) \\ &- \max(S_T - K_{\text{call}}, 0) - P_0 + C_0. \end{aligned} \tag{5}$$

By embedding these non-linear payoff functions directly into the execution modules, the system ensures that the coordinator acts as a selector and mixer of mathematically bounded action primitives.

An additional safety layer within the execution stage defines a deterministic capital-preservation override when portfolio drawdown exceeds 10% relative to the portfolio value at the most recent rebalance. This threshold is a simple operational safeguard intended to approximate a fund-style capital-preservation rule rather than a separately optimized feature or an object of empirical study. Inspection of the three recorded Full System trajectories found no threshold-crossing event across the 753 evaluation states in any run. The reported trajectories therefore reflect the normal execution path rather than override activation, while the layer remains part of the predefined execution specification for adverse conditions [14].

The resulting flow integrates retrieved memory, bounded synthesis, and constrained execution while supporting targeted component-level comparisons. The next sections use this decomposition to examine which mechanisms are associated with stable behavior and where memory by itself is incomplete.

4. Empirical Analysis of Robustness Drivers

This section examines which components are associated with stability in the empirical decision task. Return-based metrics are treated as observable outputs of the decision process, while the analysis emphasizes drawdown behavior, tail-risk compression, risk-adjusted behavior, and stability under perturbation.

4.1. Evaluation Protocol and Environmental Setup

The evaluation prioritizes repeated decision behavior under realistic frictions over nominal profitability under idealized assumptions. We use daily historical market data for highly liquid large-cap equities, namely Apple (AAPL), Tesla (TSLA), and NVIDIA (NVDA), and evaluate the decision system over the period from January 2021 to December 2023. The episodic memory is initialized from pre-evaluation observations from 2016–2020. During chronological evaluation, completed episodes are appended to the memory only after their outcomes become observable and are therefore available exclusively to subsequent decisions. This temporal separation ensures that the initial memory does not contain episodes from the 2021–2023 evaluation period. The selected universe combines active options markets with materially different volatility profiles and regime behavior. The evaluation trace contains 753 daily decision states per run, with recorded stock and option positions at each step, enabling component-level analysis across a dense sequence of decisions rather than only aggregate portfolio endpoints. Market states are evaluated daily, while portfolio rebalancing follows a lower-frequency schedule, limiting unnecessary turnover and transaction costs and reflecting a periodic portfolio-management setting rather than a high-frequency trading regime.

Transaction frictions are explicitly enforced. Equity orders incur a commission cost of 10 basis points (bps), option trades incur 20 bps, and all transactions additionally include a uniform 2 bps slippage component. Liquidity and feasibility checks are applied before execution. The coordinating language-model layer is executed with temperature fixed at $T = 0.0$ to minimize sampling-based variability. Despite this setting, the three recorded full-system runs produced different structured outputs; these differences propagate through the sequential system even though the input data, prompt templates, and execution rules are fixed.

Option execution is handled through deterministic feasibility checks after the coordinator selects contracts from strategy-relevant candidates. This keeps the empirical setting realistic while ensuring that all finalized actions remain bounded by budget, availability, and predefined risk limits.

To evaluate robustness rather than raw gain alone, we group the metrics into two categories. The first category consists of core performance indicators: Sharpe Ratio (SR) [37], Sortino Ratio (SoR), Calmar Ratio (CaR), Total Return (TR), and Maximum Drawdown (MD). The second category focuses on tail-risk and stability: Value at Risk (VaR), Conditional Value at Risk (CVaR), Worst Month (WM), Downside Deviation (DDv), Time in Drawdown above 10% (TDD), and Maximum Consecutive Loss Days (MCL). The CaR is computed as compound annual growth rate (CAGR) divided by MD. Because metrics are first computed at the run level and then averaged, ratio-based quantities such as the CaR are not necessarily equal to the ratio obtained by recomputing them from the averaged TR and averaged MD reported in the table. Results are reported over repeated evaluations, with standard deviations included where appropriate. For the learned baselines, variation reflects different random seeds used during training and evaluation. For the full system, three independent end-to-end evaluations use the same data and control settings with $T = 0.0$; the reported dispersion captures run-level differences arising from non-bitwise-deterministic hosted inference and their sequential propagation through later decisions.

To formalize our focus on tail-risk compression, we specify the previously introduced MD and CVaR metrics below. Let V_t denote the portfolio value at time t. The drawdown at time t is the decline from the historical peak:

$$DD_t = \frac{\max_{\tau \le t} V_\tau - V_t}{\max_{\tau \le t} V_\tau}.$$

The MD over the evaluation period T is simply $\text{MD} = \max_{t \in [0,T]} DD_t$.

For tail losses, let $L = -\Delta V$ denote the portfolio loss distribution. Given a confidence level $\alpha \in (0, 1)$ (e.g., $\alpha = 0.95$), the CVaR measures the expected loss strictly in the worst-case tail [38]:

$$\text{CVaR}_\alpha = \mathbb{E}[L \mid L \ge \text{VaR}_\alpha],$$

where VaR_α is the standard VaR threshold. Throughout the paper, CVaR is reported as a loss magnitude, so lower values indicate better tail-risk control. These formalizations underscore that the analysis emphasizes both pathwise deterioration (DD_t) and distributional tail losses (CVaR_α).

To support reproducibility, retrieval and generation operate with explicitly defined hyperparameters. For the episodic memory index, market states are vectorized using the `all-MiniLM-L6-v2` SentenceTransformer encoder, and the resulting embeddings are L2-normalized before exact nearest-neighbor search with the Facebook AI Similarity Search (FAISS) `IndexFlatL2` index. This makes the implemented retrieval rule consistent with cosine-based similarity ranking. The episodic neighborhood size is fixed at $k = 5$ nearest neighbors across all reported configurations. It is treated as a fixed retrieval budget rather than as a tuned optimum. The evaluation experiments explicitly override the temperature to $T = 0.0$ to minimize decoding variability. The full-system coordinator uses the primary GPT-5 endpoint specified in the public run configuration, while the named lightweight coordinator variants correspond to lower-cost or smaller model endpoints used at evaluation time. The public repository provides the implementation, exact prompt templates, decoding and model settings, seed-controlled experiment runners, dependency requirements, and reproduction instructions. As a post-hoc retrieval-level sensitivity check, the stored-state projections were also evaluated with alternative neighborhood sizes. The $k = 5$ and $k = 10$ settings produced identical retrieved-allocation outputs over all 753 tested states, whereas $k = 1$ and $k = 3$ changed a subset of allocations. This check concerns retrieval outputs rather than end-to-end portfolio performance and is not used to claim that $k = 5$ is optimal.

Table 1 summarizes the main evaluation settings, retrieval parameters, transaction costs, and bootstrap configuration.

Table 1. Compact reproducibility summary for the main empirical configuration.

Element	Setting
Initial episodic memory period	2016–2020
Evaluation period	2021–2023
Assets	AAPL, TSLA, NVDA
Retrieval encoder	all-MiniLM-L6-v2
Index	FAISS `IndexFlatL2`
Neighborhood size	$k = 5$
LLM decoding	$T = 0.0$; stochastic ablation $T = 0.7$
Costs	10 bps equity, 20 bps options, 2 bps slippage
Bootstrap	10,000 resamples, block size 20
Repository	Public GitHub repository linked in the Introduction

4.2. Component Ablation: Memory, Coordinator Variants, and Stochasticity

We begin the empirical analysis by isolating components of the decision system before comparing the full system against external benchmark configurations. In particular, we ask whether stability is mainly attributable to the choice of coordinating language model, to low-temperature decoding, or to access to relevant past experiences through episodic retrieval. The full-system row uses the primary GPT-5 coordinator. GPT-5 Nano, Gemini-2.5-Flash Lite, and GPT-4o Mini are treated as alternative lightweight coordinator variants under comparable prompting and orchestration, rather than as claims about a precisely ordered scale.

The control ablation in Table 2 provides the clearest answer to the paper's first question. Removing episodic memory produces a much stronger deterioration than replacing the main coordinating model with smaller variants. In particular, the transition from the full system to the No Memory configuration increases maximum drawdown from 16.22% to 46.18% and worsens CVaR from 2.29% to 4.78%. The degradation is also visible in worst-month losses, downside deviation, and time spent in severe drawdown, indicating that the effect is not limited to one isolated metric.

Table 2. Component ablation of coordinator variants, stochasticity, and memory. Values represent averages over three repeated evaluations.

	Core Metrics					Tail-Risk & Stability					
Configuration	**SR**	**SoR**	**CaR**	**TR (%)**	**MD (%)**	**VaR (%)**	**CVaR (%)**	**WM (%)**	**DDv (%)**	**TDD (%)**	**MCL**
Full System	1.69 ± 0.04	3.00	2.07	152.08	16.22 ± 3.28	1.63	2.29 ± 0.27	−4.58	10.93	15.18	9.7
GPT-5 Nano	1.23 ± 0.24	2.04	1.31	119.01	23.40 ± 6.63	2.22	2.91 ± 0.28	−10.58	14.09	35.50	8.0
Gemini-2.5-Flash Lite	1.16 ± 0.29	1.90	1.32	91.65	19.15 ± 4.10	1.88	2.58 ± 0.46	−7.84	12.52	27.45	8.3
GPT-4o Mini	1.14 ± 0.30	1.86	1.34	97.44	20.41 ± 6.00	2.08	2.87 ± 0.33	−9.26	13.77	33.38	8.3
Stochastic ($T = 0.7$)	0.96 ± 0.05	1.52	0.88	84.58	28.28 ± 12.17	2.44	3.28 ± 1.23	−12.27	15.86	40.95	8.3
No Memory	0.80 ± 0.10	1.23	0.62	93.94	46.18 ± 5.25	3.64	4.78 ± 0.19	−19.95	22.89	58.12	7.0

Bootstrap analysis supports this interpretation most strongly for risk-adjusted and downside metrics: compared to the No Memory variant, the full system yields strictly positive improvements in MD and CVaR across all evaluated resamples ($p < 0.001$). The same comparison also supports a significant improvement in SR ($p = 0.0085$), whereas total-return differences remain less stable across resamples. By contrast, the lightweight coordinator variants show smaller nominal differences than the no-memory ablation, indicating that access to relevant memory and the surrounding system structure are more consequential than coordinator choice in this setting.

Introducing stochasticity also weakens stability. When the same control logic is executed with non-zero temperature ($T = 0.7$), downside risk becomes more variable and the resulting trajectories exhibit larger drawdowns and heavier tail losses. This supports the interpretation that low-temperature decoding matters not because it makes the system "smarter" but because it reduces unnecessary variance in a bounded sequential decision process.

Most importantly, Table 2 motivates the next stage of the analysis. If memory supplies an important component of the robustness mechanism, then the natural next question is whether the memory signal alone can reproduce the observed performance, or whether the Full System configuration provides additional benefits. Section 4.3 addresses this directly.

4.3. Is Memory Enough? Retrieval-Only and Random-Retrieval Analysis

The previous subsection showed that the stability effect weakens sharply when memory is removed. However, because memory was evaluated there as an input to the coordinating

model, it remains unclear how much of the observed stability comes from retrieval itself and how much from the model that processes it. This subsection addresses that question directly through two tests: a retrieval-only controller and a random-retrieval setting.

4.3.1. The Retrieval-Only Controller

To separate the informational value of memory from the coordinating role of the language model, we construct a *Retrieval-Only Controller*. In this configuration, the language model is removed from the allocation stage, while the memory buffer and deterministic execution layer remain unchanged. The controller is intentionally simple: retrieval is used to select relevant historical episodes, but no model-based synthesis or learned weighting function is applied. This makes it a conservative test of whether stored experience alone carries enough structure to support stable behavior.

Recall from Section 3 that the retrieval mechanism returns a neighborhood of past episodes

$$\mathcal{N}_k(s_t),$$

where each stored episode contains a historical state together with its associated allocation and realized outcome. Let

$$\mathrm{sim}(\phi(s_t), \phi(s_j))$$

denote the similarity between the current state and a retrieved historical state. In the retrieval-only controller, similarity is used only to select the top-k neighborhood. The subsequent allocation is constructed directly from the stored historical allocations and actions in that neighborhood, rather than from a language-model synthesis step.

For the portfolio-level allocation component, let ρ_j denote the outcome-based adjustment associated with retrieved episode e_j, derived from its realized reward or performance signal. The portfolio-level retrieval-only allocation is written compactly as

$$\tilde{w}_t^{\mathrm{RO}} = \sum_{e_j \in \mathcal{N}_k(s_t)} \rho_j w_j,$$

followed by normalization and feasibility checks:

$$w_t^{\mathrm{RO}} = \mathrm{Normalize}\big(\tilde{w}_t^{\mathrm{RO}}\big).$$

For asset-level allocation components, the controller aggregates the retrieved stock and option allocation percentages using an unweighted average over the retrieved neighborhood, followed by the same deterministic normalization and execution checks. Execution-module weights are similarly aggregated from the retrieved actions and normalized before being passed to the fixed execution layer.

This allocation is then passed directly to the same deterministic execution layer used by the full system. No free-form synthesis or model-based coordination is involved. As a result, the retrieval-only controller serves as a strict baseline: it preserves access to episodic memory while removing the model's ability to reinterpret, combine, or refine the retrieved signal.

4.3.2. Memory Corruption and Temporal Structure Sensitivity

If the retrieval-only controller performs reasonably well, one possible explanation is that almost any historical context regularizes the decision process. To test whether this is really a memory effect rather than a generic smoothing effect, we introduce a corrupted-memory setting that breaks the alignment between the current state and the retrieved episodes.

Specifically, *Random Retrieval* preserves the same memory construction and eligibility filters but replaces similarity-based ranking with uniform sampling among eligible episodes:

$$\mathcal{N}_k^{\mathrm{rand}}(s_t) \sim \mathcal{U}(\mathcal{M}_{\mathrm{eligible}}(s_t)).$$

Under this configuration, the controller still receives valid historical episodes and their associated allocations, but the similarity-based relationship between the current state s_t and the retrieved states is removed. Thus, the sampled episodes remain historically eligible, but they are no longer ranked by similarity to the current state.

This test allows us to distinguish between two possibilities. If robustness comes mainly from mixing historical allocations in general, then the random-retrieval controller should remain relatively stable. If, by contrast, robustness depends on retrieving *relevant* past situations, then corrupting the retrieval rule should produce a marked deterioration in risk-adjusted and tail-risk behavior.

4.3.3. Comparative Analysis: Retrieval, Corruption, and Full Coordination

Table 3 compares the Full System, Retrieval-Only, Random Retrieval, and No Memory configurations. Retrieval-Only achieves a higher average Total Return than No Memory (100.03% vs. 93.94%), but also a larger Maximum Drawdown (52.08% vs. 46.18%). This contrast suggests that retrieved historical allocations may provide a useful directional signal without necessarily improving downside control. The Full System, which combines relevant retrieval with model-based coordination and constrained execution, records a substantially lower Maximum Drawdown of 16.22%. These differences characterize the evaluated configurations rather than independently identifying the effects of individual components.

The comparison also highlights a distinction between retrieving past actions and synthesizing them in the current decision context. Retrieval-Only directly aggregates historical allocations, whereas the Full System uses retrieved episodes as inputs to model-based coordination. Because these configurations differ in both coordination and allocation construction, the observed performance gap cannot be attributed to either mechanism in isolation.

The random-retrieval condition further clarifies the mechanism. Relative to the full system, replacing nearest-neighbor retrieval with uniform sampling substantially reduces risk-adjusted performance and increases run-level variability. Average TR falls to 59.11% with high dispersion across runs, and the SR decreases to 0.64. At the same time, random retrieval does not uniformly underperform the no-memory variant on all risk metrics: its average MD and CVaR remain lower than those of the no-memory configuration. This pattern is compatible with a regularizing role for historical allocations and suggests that state-aligned retrieval may be relevant to the observed performance differences. However, the comparison does not establish that retrieval relevance or model-based synthesis is independently necessary for low-tail-risk behavior.

Table 3. Mechanistic isolation of retrieval effects. The comparison evaluates whether memory alone is sufficient, whether retrieval relevance matters, and how full coordination changes downside behavior. Values are reported as mean ± standard deviation across the evaluated runs. Retrieval-Only is reported over two complete evaluations; configurations with three repetitions use $n = 3$.

Configuration	SR	TR (%)	MD (%)	CVaR (%)	TDD (%)
Full System	1.69 ± 0.04	152.08 ± 25.26	16.22 ± 3.28	2.29 ± 0.27	15.18 ± 8.29
Retrieval-Only	0.787 ± 0.005	100.03 ± 0.67	52.08 ± 0.19	5.35 ± 0.08	55.91 ± 0.00
Random Retrieval	0.64 ± 0.17	59.11 ± 41.51	33.70 ± 13.99	3.85 ± 1.39	54.67 ± 4.83
No Memory	0.80 ± 0.10	93.94 ± 19.43	46.18 ± 5.25	4.78 ± 0.19	58.12 ± 3.66

The paired bootstrap comparison confirms this boundary condition. Relative to Random Retrieval, the full system improves SR in 99.77% of resamples ($p = 0.0023$), MD in 99.95% of resamples ($p = 0.0005$), and CVaR in 100% of resamples ($p < 0.001$). Total-return improvements remain directionally favorable but less statistically stable ($p = 0.1120$), again indicating that the dominant effect concerns risk-adjusted and downside behavior.

Taken together, these tests show that, within this empirical setting, the strongest observed stability occurs when relevant episodic memory, model-based coordination, and constrained execution are present together.

This distributional shift is visualized in Figure 3, which compares the daily return densities of the full system against the retrieval-only and no-memory variants. The vertical lines, marking the 95% CVaR, highlight differences in left-tail losses between the Full System and the evaluated partially ablated configurations without isolating the causal contribution of any single component.

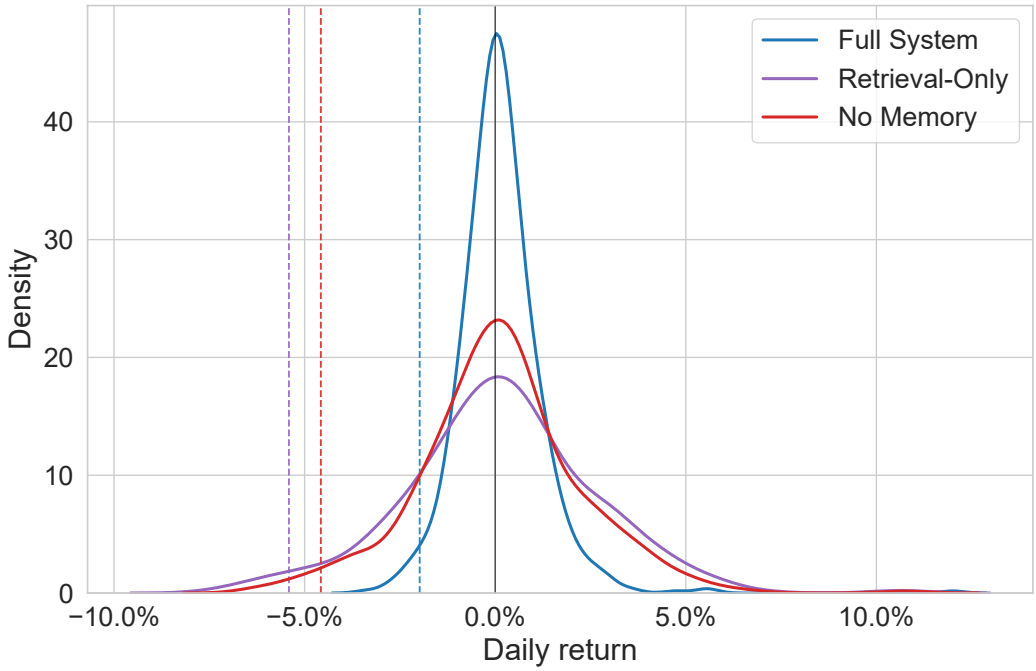


Figure 3. Kernel density estimate (KDE) of daily returns for the Full System and partially ablated configurations. Vertical dashed lines mark the negative of the 95% CVaR loss magnitude on the return axis, summarizing average losses in the worst 5% tail rather than a quantile cutoff. The comparison does not isolate the causal contribution of any single component.

4.4. Execution Modules and Adaptive Coordination

The preceding analyses examined memory removal and retrieval isolation. Here the question is different. Even if memory matters, the observed effect could still be explained by a single strong execution primitive or by a fixed combination of modules. To test this possibility, we compare the full coordinated system against isolated primitives and their static pairwise combinations.

The execution-module ablation in Table 4 shows that robustness cannot be reproduced simply by selecting one good primitive and applying it consistently. Some isolated modules perform reasonably well along one dimension, for example by limiting drawdown or maintaining a moderate SR, but none of them reproduce the combined behavior of the full system across both return-sensitive and tail-sensitive metrics. Static pairwise combinations improve on some isolated variants, yet they remain materially less stable than the adaptively coordinated system.

The reduced table reports the metrics most directly tied to the section's claim: risk-adjusted behavior, total outcome, drawdown depth, tail loss, and time spent in severe drawdown. The omitted diagnostics follow the same qualitative pattern for SoR, CaR, VaR, WM, and DDv, while maximum consecutive losses are not used as a primary claim because they do not uniformly favor the full system.

Table 4. Execution-module ablation: isolated primitives, static combinations, and adaptive coordination (C = collar, S = straddle, D = delta-neutral). Values represent averages over three evaluations. Checkmarks indicate included execution modules. Bold values indicate the best result for each metric (higher SR and TR; lower MD, CVaR, and TDD).

Modules			Core Outcomes		Downside Stability		
C	**S**	**D**	**SR**	**TR (%)**	**MD (%)**	**CVaR (%)**	**TDD (%)**
	✓		1.11 ± 0.06	122.67	30.92 ± 0.58	3.54 ± 0.11	53.06
		✓	1.05 ± 0.14	74.29	19.89 ± 5.92	2.58 ± 0.45	27.40
✓			0.98 ± 0.04	69.51	23.81 ± 1.42	2.58 ± 0.39	50.20
✓	✓		1.01 ± 0.07	80.28	23.59 ± 3.02	2.91 ± 0.05	43.82
	✓	✓	1.01 ± 0.08	87.66	25.00 ± 5.22	3.09 ± 0.33	50.36
✓		✓	1.03 ± 0.08	82.33	21.31 ± 2.00	2.98 ± 0.12	38.65
✓	✓	✓	**1.69 ± 0.04**	**152.08**	**16.22 ± 3.28**	**2.29 ± 0.27**	**15.18**

This pattern is consistent with a benefit from context-dependent recombination rather than the unconditional superiority of any single primitive. Across the evaluated variants, the strongest observed configuration combines adaptive coordination, retrieved experience, and execution constraints.

4.5. Contextual Benchmarking of System Behavior

After isolating the core mechanisms, we provide contextual validation by comparing the full system against standard benchmark configurations and learning-based baselines. We specifically select Proximal Policy Optimization (PPO), FinRL, and the recent FinAgent architecture because they represent established approaches to end-to-end learned control and financial decision-making [11,39,40], providing a clear contrast to a decoupled memory–coordination–execution design. All benchmark configurations are evaluated over the same asset universe and 2021–2023 evaluation horizon, using the same initial capital, transaction-cost assumptions, and metric computation procedure where technically applicable. Method-specific action spaces are retained where required by the original architectures; in particular, baselines that do not support option-based execution operate over their native equity action spaces. The comparison therefore holds the evaluation environment and reporting protocol constant while preserving the native decision mechanisms of each baseline. These comparisons are intended as contextual reference points rather than as a claim of state-of-the-art superiority, since the architectures and native action spaces differ materially from those of the proposed system. It positions the proposed system relative to representative passive, reinforcement-learning, and agent-based approaches under the evaluated setting; the episodic retrieval buffer is specific to the proposed system. Figure 4 complements the tabular summary by showing the drawdown trajectories over time.

The main observation from Table 5 and Figure 4 is that the full system materially compresses the left tail of the outcome distribution while maintaining stronger risk-adjusted performance. Relative to both rule-based and reinforcement-learning baselines, the system exhibits substantially lower maximum drawdown and lower conditional tail loss.

In particular, the reduction in drawdown from 39.72% for PPO to 16.22% for the full system, and the reduction in CVaR from 3.95% to 2.29%, indicate that the main observed effect is not merely better average performance, but improved containment of adverse trajectories. As shown in the underwater profile, the standard baselines experience deep and persistent capital degradation during difficult market intervals, whereas the full system remains materially closer to prior equity peaks. The improvement is concentrated in drawdown depth, tail-loss magnitude, and time spent in severe drawdown; it does not extend

to every stability indicator, as maximum consecutive loss streaks remain higher than for several baselines.

Table 5. Contextual benchmark configurations for the full AI decision system and external decision variants. Values represent averages over three repeated evaluations where applicable; standard deviations are reported for configurations with run-level variation. Bold values indicate the best result for each metric; for MCL, shorter loss streaks are preferable.

	Core Metrics					Tail-Risk & Stability					
Benchmark Configuration	**SR**	**SoR**	**CaR**	**TR (%)**	**MD (%)**	**VaR (%)**	**CVaR (%)**	**WM (%)**	**DDv (%)**	**TDD (%)**	**MCL**
Buy & Hold	0.81	1.28	0.61	107.08	53.08	4.15	5.36	−21.96	25.34	54.12	7.0
Equal Weight	0.77	1.17	0.56	95.34	53.53	4.16	5.41	−21.29	25.73	54.65	7.0
PPO	0.74 ± 0.07	1.13	0.53	66.46	39.72 ± 4.26	2.99	3.95 ± 0.23	−15.59	18.68	47.87	**6.7**
FinAgent	0.63 ± 0.03	0.93	0.50	69.74	54.60 ± 7.13	4.29	5.99 ± 0.62	−21.92	28.69	60.20	8.7
FinRL	0.56 ± 0.07	0.86	0.43	57.08	61.51 ± 8.18	4.93	6.48 ± 0.64	−25.19	30.78	73.97	7.0
Full System	**1.69 ± 0.04**	**3.00**	**2.07**	**152.08**	**16.22 ± 3.28**	**1.63**	**2.29 ± 0.27**	**−4.58**	**10.93**	**15.18**	9.7

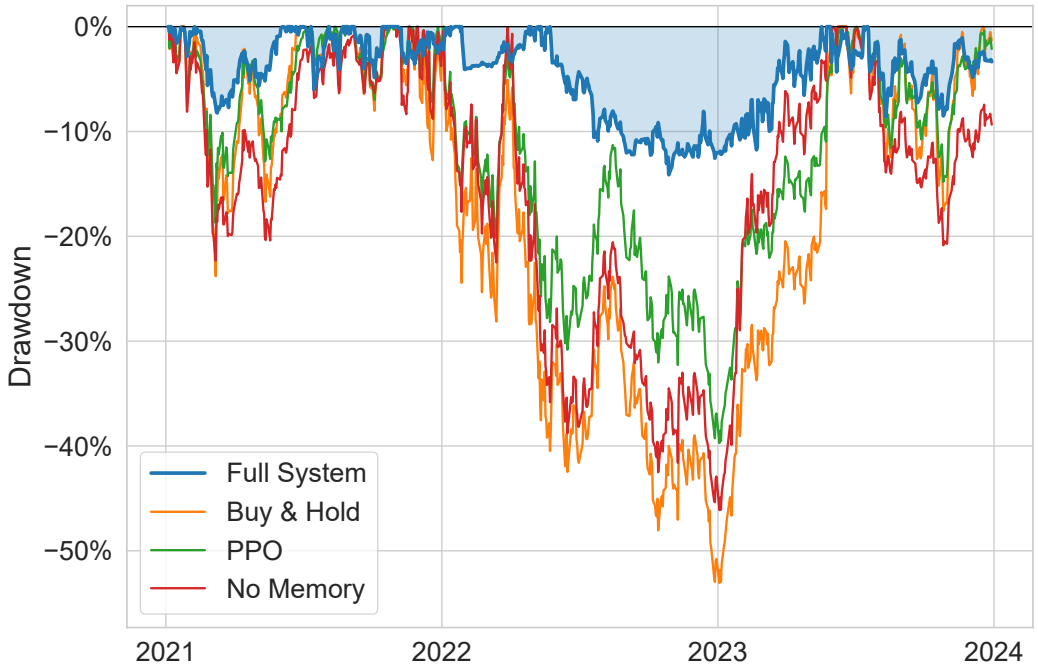


Figure 4. Underwater drawdown profile of the benchmark configurations and the full AI decision system. The figure is used as a path-dependent stability diagnostic and highlights the shallower drawdown trajectory observed for the full system during the 2022 market shock. Light-blue shading highlights the Full System drawdown.

To assess the stability of the observed risk compression under temporal resampling, we apply a paired circular block-bootstrap analysis to aligned daily return series (N = 10,000 resamples, block size 20 trading days). In each bootstrap draw, the same sampled block indices are applied to the two configurations being compared, preserving run-level pairing and local serial dependence before metric differences are recomputed. The strongest and most consistent evidence appears in the robustness-oriented metrics. For both Full System vs. Buy & Hold and Full System vs. PPO, improvements in MD and CVaR are positive in 100% of bootstrap resamples, with one-sided bootstrap p-values below 0.001. Sharpe-ratio improvements are also statistically supported in these comparisons ($p = 0.0077$ against Buy & Hold and $p = 0.0067$ against PPO). By contrast, improvements in total return are less stable across resamples: positive total-return deltas occur in 58.51% of resamples against Buy & Hold and 89.08% against PPO, with non-significant one-sided p-values. These results

indicate that the clearest observed differences concern risk-adjusted and downside behavior rather than total return alone.

These contextual comparisons show that the mechanism-oriented findings occur in a system whose external behavior is materially more stable than the behavior of both static and learned baselines.

4.6. Robustness Checks and Boundary Conditions

To examine whether the observed pattern is stable across market regimes and perturbations, we split the recorded 2021–2023 decision trace into regime-level windows and conduct three auxiliary tests: a broad-market stress year, a synthetic environment, and a temporal-disruption placebo [41,42], with results summarized in Tables 6 and 7. The synthetic environment provides a complementary check under generated dynamics that do not correspond to the realized historical path used in the main evaluation.

Table 6. Regime-level MD and CVaR for the Full System, No Memory, and Retrieval-Only configurations within the main evaluation trace. The comparison examines whether the observed downside-control pattern persists across different market conditions. Bold values indicate the lowest MD and CVaR in each regime.

	Maximum Drawdown (%)			CVaR (%)		
Regime	**Full**	**No Memory**	**Retrieval-Only**	**Full**	**No Memory**	**Retrieval-Only**
Full 2021–2023	**16.22 ± 3.28**	46.18 ± 5.25	52.21	**2.29 ± 0.27**	4.78 ± 0.19	5.41
2021 expansion	**8.27 ± 2.80**	24.44 ± 0.80	23.41	**2.39 ± 0.43**	4.20 ± 0.05	4.69
2022 bear market	**16.22 ± 3.28**	45.41 ± 5.25	51.60	**2.03 ± 0.31**	5.50 ± 0.39	6.26
2023 rebound	**9.02 ± 0.97**	20.89 ± 2.01	20.02	**2.35 ± 0.23**	3.40 ± 0.16	3.81
Volatile 2021–2022	**12.02 ± 3.61**	38.79 ± 1.14	41.54	**2.40 ± 0.46**	5.53 ± 0.43	6.03

Table 6 checks whether the observed downside-risk compression is driven by a single aggregate evaluation period. Across the full period, the 2021 expansion, the 2022 bear market, the 2023 rebound, and the volatile 2021–2022 window, the full system retains lower drawdown and CVaR than both the no-memory and retrieval-only variants. This suggests that the effect is not limited to one endpoint of the evaluation trace, but remains visible across different market conditions.

Full System and No Memory values in Table 6 summarize repeated evaluations; Retrieval-Only regime values are computed from the trajectory used for the chronological regime decomposition.

Table 7 reports the stress-year, synthetic-environment, and temporal-disruption results.

The stress-year and synthetic-environment tests examine the same robustness pattern under additional market and generated conditions. The stress-year test uses the SPDR S&P 500 exchange-traded fund (ETF), ticker SPY, during 2022 as a liquid broad-market proxy for S&P 500 conditions. The synthetic environment provides a controlled, perturbed market-like process with known generated dynamics. In both settings, the full system preserves the qualitative pattern observed in the main experiment: weaker downside exposure, shallower drawdowns, and more stable tail behavior than the reference baseline. In the stress-year panel, the full system improves SR, TR, MD, and CVaR relative to Buy & Hold, while the WM is marginally more severe. Although absolute returns are naturally compressed during the stress year, the relative improvement in downside protection remains intact on the main tail-risk metrics.

The temporal-disruption test probes the role of chronological structure directly. The evaluation sequence is partitioned into contiguous 10-trading-day blocks; the block order is permuted for each run using the corresponding experiment seed, while observa-

tions within each block retain their original order. Under this disruption, risk behavior deteriorates sharply, supporting the interpretation that retrieval benefits from meaningful temporal or regime alignment rather than only time-invariant statistical regularities.

Together with the retrieval-isolation analysis above, these tests show that the observed downside-control pattern persists across the stress-year and synthetic settings, while the temporal-disruption placebo demonstrates the importance of chronological alignment.

Table 7. Robustness checks and boundary conditions: stress-year, synthetic-environment, and temporal-disruption tests. Bold values indicate the best result for each metric within Panels A and B; Panel C is not highlighted.

Panel A: SPY Broad-Market Stress (2022)					
Configuration	**SR**	**TR (%)**	**MD (%)**	**CVaR (%)**	**WM (%)**
Buy & Hold	−0.81	−19.95	25.43	3.35	**−9.34**
Full System	**0.34 ± 0.09**	**5.56 ± 2.31**	**18.41 ± 0.56**	**2.78 ± 0.22**	−9.67 ± 0.41
Panel B: Synthetic Environment					
Configuration	**SR**	**TR (%)**	**MD (%)**	**CVaR (%)**	**WM (%)**
Buy & Hold	0.11	−0.18	41.71	3.47	−15.47
Full System	**0.46 ± 0.09**	**24.55 ± 6.26**	**25.53 ± 2.86**	**2.98 ± 0.48**	**−10.15 ± 1.68**
Panel C: Temporal Disruption Placebo					
Configuration	**SR**	**TR (%)**	**MD (%)**	**CVaR (%)**	**WM (%)**
Buy & Hold	0.64	51.40	30.91	3.85	−12.23
Full System	0.63 ± 0.05	69.66 ± 25.03	52.11 ± 20.29	12.10 ± 2.06	−31.68 ± 4.74

4.7. Efficiency and Latency Analysis

Robust behavior in this system is not free. The combination of retrieval, structured coordination, and constrained execution introduces computational overhead that must be understood if the system is to remain practical in realistic decision settings.

In the current implementation, an average daily decision cycle uses approximately 24,962 tokens in aggregate across its model calls, including prompts, retrieved context, and structured outputs. Mean model inference latency is 2.54 s per decision call, while similarity search over the episodic memory index adds only sub-millisecond overhead. These numbers indicate that the dominant cost comes from the coordinating model, not from the retrieval mechanism itself.

This is analytically useful for two reasons. First, it shows that the memory layer can be made rich without dominating runtime. This suggests that the non-parametric memory buffer can be expanded more easily than the coordinating model, provided that retrieval remains efficient. Second, it suggests that the full system is most appropriate in low- to medium-frequency control settings, such as daily portfolio adjustment, where a delay of a few seconds remains operationally acceptable.

The results of this section establish three points. First, in this empirical setting, removing memory hurts robustness more than replacing the primary coordinator with lightweight alternatives. Second, retrieval by itself is incomplete: the strongest observed stability is associated with the configuration that combines relevant memory, model-based synthesis, and adaptive coordination across execution modules. Third, the full system exhibits substantially more stable behavior than standard baselines as contextual validation, while the observed pattern persists across additional stress-year and synthetic settings but depends on meaningful temporal and retrieval structure. The main and auxiliary experiments therefore characterize robustness within the selected assets across multiple evaluated market regimes

and controlled perturbations, while motivating evaluation across broader universes and decision frequencies in future work.

5. Discussion

The results are best interpreted in relation to three strands of prior research: retrieval-augmented and memory-based AI systems, LLM-centered financial agents, and modular or constrained decision architectures. Rather than establishing the independent causal contribution of every component, the experiments identify performance differences among targeted system configurations. The following discussion examines how these observations relate to earlier work, which interpretations are supported by the ablations, and which questions remain unresolved.

5.1. Memory Is Useful but Incomplete

The memory ablations provide a more qualified perspective on the value of retrieved experience than is typically available from end-to-end evaluations. Retrieval-augmented generation and case-based reasoning establish the broader rationale for using external information and similar prior cases to condition model behavior [20,25]. TradingGPT similarly incorporates layered memory into a financial multi-agent architecture [4], while AriGraph explores episodic memory as part of an agent's learned world representation [26]. These studies motivate the use of memory, but their architectural objectives and evaluation settings differ from the component-level robustness comparisons conducted here.

Our findings are consistent with the premise that retrieved experience can provide a useful conditioning signal, but they qualify any assumption that memory access alone is sufficient. In the evaluated financial setting, removing the episodic memory component is associated with substantially greater drawdown and tail losses. However, the Retrieval-Only controller does not reproduce the downside profile of the Full System, and Random Retrieval substantially weakens risk-adjusted performance relative to similarity-based retrieval within the full architecture. Thus, the results extend the discussion of memory-enabled systems by examining not only whether memory is available, but also how retrieved experience is selected and used.

This distinction is important because the Retrieval-Only comparison changes the allocation mechanism as well as the presence of model-based synthesis. Its weaker results cannot be attributed exclusively to the absence of an LLM coordinator. Likewise, the temporal-disruption placebo changes the chronological structure of the evaluation and does not isolate retrieval quality as its only affected mechanism. Taken together, the tests are consistent with a role for retrieval relevance and temporal alignment, but they do not identify an independent causal effect for every memory-related mechanism. More detailed episode-to-decision tracing would be needed to establish which retrieved experiences influence particular allocations.

5.2. Joint Configuration of Memory, Coordination, and Execution

The findings also need to be positioned against earlier financial agent architectures. FinAgent combines multiple information modalities and tools in an LLM-centered framework for financial trading [11]. Modularized portfolio-management systems such as MSPM pursue a different approach, structuring coordination through specialized reinforcement-learning agents [35]. Both lines of research show that financial decisions can be organized around multiple interacting components. The present study differs in emphasis: rather than introducing a new general-purpose financial agent, it compares selected memory, coordination, retrieval, and execution configurations within a common modular testbed.

The contextual benchmark illustrates both the promise and the limits of this comparison. Under the reported evaluation protocol, the Full System achieves a SR of 1.69, MD of 16.22%, and CVaR of 2.29%, compared with 0.63, 54.60%, and 5.99%, respectively, for the evaluated FinAgent baseline. These differences establish stronger observed risk-adjusted and downside performance for the Full System in this particular benchmark setting. They do not establish an architecture-level superiority over FinAgent, because the systems retain distinct decision mechanisms and native action spaces. In particular, the benchmark is not a controlled experiment in which only the coordinator or memory component differs.

Within the proposed architecture, replacing the primary coordinator with lightweight alternatives produces smaller nominal degradations than removing memory, while increasing decoding temperature is associated with weaker downside behavior. These observations suggest that coordinator configuration and decoding stability may matter alongside retrieval. However, the experiments do not isolate a general causal contribution of LLM reasoning or establish that the coordinating model itself produces the observed downside protection. The Retrieval-Only controller also changes how historical actions are aggregated, so its comparison with the Full System cannot separate model-based synthesis from all other allocation differences.

Finally, the execution-module experiments show that none of the tested isolated primitives or fixed pairwise combinations reproduces the Full System's combined performance profile. This is compatible with the value of context-dependent allocation across bounded execution modules, an interpretation aligned with the broader rationale for modular system design [5,14]. Nevertheless, these experiments compare predefined module configurations rather than independently removing every feasibility or risk constraint. Accordingly, the results support a configuration-level interpretation, not a separately identified causal effect of constrained execution.

5.3. Implications for AI-Supported Decision Systems

A practical implication of these findings is methodological rather than universal: robustness claims for modular AI decision systems should be examined at the level of their interacting components, not inferred solely from the performance of the coordinating model. Prior research on retrieval-augmented generation and modular decision support motivates the combination of external context, model-based processing, and structured downstream operations [2,20]. Our results add an empirical illustration of why comparisons that remove or perturb individual components can change the interpretation of end-to-end performance.

For systems with similar sequential decision structures, three design hypotheses merit further testing. First, retrieval relevance may be more informative than memory availability alone, given the observed deterioration under random retrieval. Second, coordinator behavior should be evaluated together with the allocation and execution mechanisms through which its outputs affect the environment. Third, bounded action interfaces and explicit feasibility checks are plausible safeguards, although their independent contribution to robustness has not been established by the present experiments.

These are conditional recommendations derived from one financial testbed, not demonstrated principles for AI decision systems in general. The relevance of analogous component-isolation studies in other domains is suggested by work on retrieval-based decision support [2,28], but transferring the present empirical conclusions to such domains would require dedicated validation.

5.4. Scope, Limitations, and Future Directions

Several limitations constrain the interpretation and generalizability of the results. The main evaluation covers three large-cap US equities over 2021–2023, with additional

stress-year, synthetic-environment, and temporal-disruption checks. Although these tests expose the architecture to different conditions, they do not establish robustness across broader asset universes, longer market histories, different execution frequencies, or non-financial decision domains. The external benchmarks also differ in architecture and supported action spaces, limiting their interpretation as controlled comparisons of individual design choices.

A particularly important limitation concerns retrospective evaluation with a hosted pretrained language model. The external episodic memory is initialized from observations preceding the evaluation period and is updated chronologically only after outcomes become available. This procedure restricts explicit look-ahead through the memory pipeline. It does not rule out the possibility that the pretrained coordinator contains parametric knowledge of market events or price-related information from 2021–2023. Consequently, the reported differences cannot be interpreted as a fully prospective test of decision-making under information unavailable to the model during pretraining. The synthetic-environment experiment provides complementary evidence under generated market dynamics but does not eliminate this limitation. Future work should consider anonymized asset and date representations, explicit historical-knowledge probes, and prospective or genuinely out-of-sample evaluations.

The targeted ablations also do not constitute a complete factorial design. In particular, comparisons involving Retrieval-Only change both model-based synthesis and allocation construction, while execution-module ablations do not independently disable every downstream feasibility constraint. A more complete factorial analysis would be required to estimate interaction effects among memory, coordination, and execution. Similarly, the relatively small number of independent system evaluations limits run-level uncertainty estimation. The paired block bootstrap evaluates sensitivity to temporal resampling of recorded returns, but does not substitute for additional independent model and system runs; this distinction is relevant to established concerns about backtest overfitting and out-of-sample validation [41].

Finally, the study does not provide episode-level causal attribution. Tracking which retrieved cases enter each coordinator input, how they influence structured outputs, and how these outputs propagate through execution would offer a more direct test of the proposed mechanisms. The present results should therefore be read as comparative evidence within a specified modular architecture rather than a complete causal decomposition of AI decision-system robustness.

6. Conclusions

AI decision systems increasingly combine language models, retrieved experience, and rule-based execution mechanisms to support decisions over time. Yet, evaluating such systems as a single pipeline makes it difficult to understand which elements contribute to their observed robustness. This paper investigated that question through targeted comparisons of memory access, retrieval quality, coordinator configuration, stochasticity, and execution-module configurations in a sequential financial decision setting. The study focused on understanding how different system configurations behave under a shared evaluation framework.

The central finding is that memory helps, but memory is not enough. Removing episodic memory substantially weakened risk-adjusted performance and downside control in the evaluated setting, while retrieval-only and random-retrieval configurations did not reproduce the performance of the full architecture. The strongest observed downside control was achieved by the configuration combining relevant memory, model-based coordination, and constrained execution. These comparisons do not isolate the independent causal effect of each component, but they motivate evaluating robustness beyond memory availability

or coordinator choice alone. The main evaluation covers three stocks over 2021–2023, supplemented by stress-year, synthetic-environment, and temporal-disruption checks; the findings remain specific to the implemented evaluation protocol, and possible historical knowledge in the pretrained coordinator further limits their interpretation.

For researchers developing memory-augmented agents, the study offers a component-oriented framework for comparing observed stability across system configurations rather than treating the agent as an indivisible system. For practitioners designing sequential decision pipelines, it highlights the value of assessing retrieval relevance, coordination behavior, and execution constraints together, instead of relying exclusively on aggregate performance measures. Extending these comparisons to prospective settings, broader environments, and more controlled component combinations is a natural next step toward understanding when memory-augmented decision systems can be considered robust.

Author Contributions: Conceptualization, F.B. and J.A.C.; methodology, F.B.; software, F.B.; validation, F.B. and J.A.C.; formal analysis, F.B.; investigation, F.B.; resources, F.B. and J.A.C.; data curation, F.B.; writing—original draft preparation, F.B.; writing—review and editing, F.B. and J.A.C.; visualization, F.B.; supervision, J.A.C.; project administration, J.A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded through a project administered by the Polish Ministry of Science and Higher Education under "Support for Students in Developing Their Competences and Skills" (Polish: "Wsparcie studentów w zakresie podniesienia ich kompetencji i umiejętności"), FERS.01.05, contract number MNiSW/2025/DPI/600, implemented within the European Funds for Social Development 2021–2027.

Data Availability Statement: Daily open, high, low, close, and volume (OHLCV) data for AAPL, TSLA, NVDA, and SPY were obtained from Yahoo Finance (https://finance.yahoo.com/). The public repository at https://github.com/latent-systems-lab/Modular-Decision-Systems-for-Robustness (accessed on 3 November 2026) provides source code, prompt templates, experiment configurations, reproduction scripts, and the synthetic-environment generator. Historical option-chain data are subject to the provider's commercial license and therefore cannot be redistributed.

Acknowledgments: The authors used AI-assisted tools, including ChatGPT (OpenAI; GPT-5.6 and GPT-6) and Codex (OpenAI; GPT-5.6 and GPT-6), to support English-language editing, restructuring and refinement of manuscript passages, and the implementation and debugging of research code. These tools were used in an assistive capacity under human supervision. The authors independently reviewed and verified the scientific content, experimental methodology, reported results, and final manuscript, and take full responsibility for the published work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAPL	Apple Inc. (stock ticker)
bps	Basis points
CAGR	Compound Annual Growth Rate
CaR	Calmar Ratio
CVaR	Conditional Value at Risk
ETF	Exchange-Traded Fund
FAISS	Facebook AI Similarity Search
KDE	Kernel Density Estimate
DDv	Downside Deviation
LLM	Large Language Model
MCL	Maximum Consecutive Loss Days
MD	Maximum Drawdown
NVDA	NVIDIA Corporation (stock ticker)

OHLCV	Open, High, Low, Close, Volume
POMDP	Partially Observable Markov Decision Process
PPO	Proximal Policy Optimization
SoR	Sortino Ratio
SPY	SPDR S&P 500 ETF Trust (ticker)
SR	Sharpe Ratio
TDD	Time in Drawdown above 10%
TR	Total Return
TSLA	Tesla, Inc. (stock ticker)
VaR	Value at Risk
WM	Worst Month

References

1. Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, Jeju, Republic of Korea, 3–9 August 2024; pp. 8048–8057. [CrossRef] [PubMed]
2. Wang, S.; Yang, H.; Bai, G. Construction of Intelligent Decision Support Systems through Integration of Retrieval-Augmented Generation and Knowledge Graphs. *Sci. Rep.* **2025**, *15*, 35462. [CrossRef] [PubMed]
3. Li, D.; Tan, Z.; Qian, P.; Li, Y.; Chaudhary, K.S.; Hu, L.; Shen, J. SMoA: Improving Multi-Agent Large Language Models with Sparse Mixture-of-Agents. In *Proceedings of the Advances in Knowledge Discovery and Data Mining*; Lecture Notes in Computer Science; Springer Nature: Singapore, 2025; Volume 15872, pp. 54–65. [CrossRef]
4. Li, Y.; Yu, Y.; Li, H.; Chen, Z.; Khashanah, K. TradingGPT: Multi-Agent System with Layered Memory and Distinct Characters for Enhanced Financial Trading Performance. *arXiv* **2023**, arXiv:2309.03736. [CrossRef]
5. Auda, G.; Kamel, M.S. Modular Neural Networks: A Survey. *Int. J. Neural Syst.* **1999**, *9*, 129–151. [CrossRef] [PubMed]
6. Macal, C.M.; North, M.J. Tutorial on Agent-Based Modelling and Simulation. *J. Simul.* **2010**, *4*, 151–162. [CrossRef]
7. Fink, A.; Gerhards, P. Negotiation Mechanisms for the Multi-Agent Multi-Mode Resource Investment Problem. *Eur. J. Oper. Res.* **2021**, *295*, 261–274. [CrossRef]
8. Buehler, H.; Gonon, L.; Teichmann, J.; Wood, B. Deep Hedging. *Quant. Finance* **2019**, *19*, 1271–1291. [CrossRef]
9. Tan, W.L.; Roberts, S.J.; Zohren, S. Deep Learning for Options Trading: An End-To-End Approach. In *Proceedings of the 5th ACM International Conference on AI in Finance*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 487–495. [CrossRef]
10. Bańka, F.; Chudziak, J.A. Options Pricing Platform with Neural Networks, LLMs and Reinforcement Learning. In *Proceedings of the Recent Challenges in Intelligent Information and Database Systems*; Springer Nature: Singapore, 2025; pp. 202–216. [CrossRef]
11. Zhang, W.; Zhao, L.; Xia, H.; Sun, S.; Sun, J.; Qin, M.; Li, X.; Zhao, Y.; Zhao, Y.; Cai, X.; et al. A Multimodal Foundation Agent for Financial Trading: Tool-Augmented, Diversified, and Generalist. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 4314–4325. [CrossRef]
12. Liang, J.; Sugang; Lin, H.; Wu, Y.; Zhao, R.; Li, Z. Reasoning RAG via System 1 or System 2: A Survey on Reasoning Agentic Retrieval-Augmented Generation for Industry Challenges. In *Proceedings of the Findings of the Association for Computational Linguistics: IJCNLP 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 1954–1966. [CrossRef]
13. Puterman, M.L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*; Wiley: Hoboken, NJ, USA, 1994. [CrossRef]
14. Liberzon, D. *Switching in Systems and Control*; Birkhäuser: Basel, Switzerland, 2003. [CrossRef]
15. DeChant, C. Episodic Memory in AI Agents Poses Risks that Should Be Studied and Mitigated. In *Proceedings of the 2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*; IEEE: New York, NY, YSA, 2025; pp. 321–332. [CrossRef]
16. Shu, Y.; Yu, C.; Mulvey, J.M. Dynamic Asset Allocation with Asset-Specific Regime Forecasts. *Ann. Oper. Res.* **2025**, *346*, 285–318. [CrossRef]
17. Chavas, J.P.; Li, J.; Wang, L. Option Pricing Revisited: The Role of Price Volatility and Dynamics. *J. Commod. Mark.* **2024**, *33*, 100381. [CrossRef]
18. Vamplew, P.; Foale, C.; Hayes, C.F.; Mannion, P.; Howley, E.; Dazeley, R.; Johnson, S.; Källström, J.; Ramos, G.; Rădulescu, R.; et al. Utility-Based Reinforcement Learning: Unifying Single-Objective and Multi-Objective Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2024; pp. 2717–2721.
19. Bańka, F.; Chudziak, J.A. MultiHedge: Adaptive Coordination via Retrieval-Augmented Control. In *Proceedings of the Computational Science—ICCS 2026*; Neumann, P., Puma, M.J., Lees, M.H., Groen, D., Dongarra, J.J., Sloot, P.M.A., Eds.; Lecture Notes in Computer Science; Springer Nature: Cham, Switzerland, 2026; Volume 16784, pp. 446–454. [CrossRef]

20. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. e NLP Tasks. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 9459–9474.
21. Araci, D. FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models. *arXiv* **2019**, arXiv:1908.10063. [CrossRef]
22. Wu, S.; Irsoy, O.; Lu, S.; Dabravolski, V.; Dredze, M.; Gehrmann, S.; Kambadur, P.; Rosenberg, D.S.; Mann, G. BloombergGPT: A Large Language Model for Finance. *arXiv* **2023**, arXiv:2303.17564. [CrossRef]
23. Yang, H.; Liu, X.Y.; Wang, C.D. FinGPT: Open-Source Financial Large Language Models. *arXiv* **2023**, arXiv:2306.06031. [CrossRef]
24. Yang, D.; Wang, F. Towards Retrieval-Augmented Large Language Model-Based Conversational Recommender System. In *Proceedings of the Advances in Knowledge Discovery and Data Mining*; Lecture Notes in Computer Science; Springer Nature: Singapore, 2025; Volume 15872, pp. 317–330. [CrossRef]
25. Aamodt, A.; Plaza, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Commun.* **1994**, *7*, 39–59. [CrossRef]
26. Anokhin, P.; Semenov, N.; Sorokin, A.; Evseev, D.; Kravchenko, A.; Burtsev, M.; Burnaev, E. AriGraph: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents. In Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 16–22 August 2025; pp. 12–20. [CrossRef]
27. Xiao, M.; Jiang, Z.; Qian, L.; Chen, Z.; He, Y.; Xu, Y.; Jiang, Y.; Li, D.; Weng, R.L.; Peng, M.; et al. Retrieval-Augmented Large Language Models for Financial Time Series Forecasting. *arXiv* **2025**, arXiv:2502.05878. [CrossRef]
28. Mugambiwa, T.; Ndlovu, B. Multi-Agent Retrieval Augmented Generation for Clinical Decision Support: A Systematic Review and Integrative Conceptual Framework. *J. Appl. Inform. Comput.* **2026**, *10*, 171–183. [CrossRef]
29. Malekzadeh, P.; Poulos, Z.; Chen, J.; Wang, Z.; Plataniotis, K.N. EX-DRL: Hedging Against Heavy Losses with Extreme Distributional Reinforcement Learning. In *Proceedings of the 5th ACM International Conference on AI in Finance*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 370–378. [CrossRef]
30. Sun, S.; Wang, R.; An, B. Reinforcement Learning for Quantitative Trading. *ACM Trans. Intell. Syst. Technol.* **2023**, *14*, 44. [CrossRef]
31. Cao, J.; Chen, J.; Farghadani, S.; Hull, J.; Poulos, Z.; Wang, Z.; Yuan, J. Gamma and Vega Hedging Using Deep Distributional Reinforcement Learning. *Front. Artif. Intell.* **2023**, *6*, 1129370. [CrossRef] [PubMed]
32. Shavandi, A.; Khedmati, M. A Multi-Agent Deep Reinforcement Learning Framework for Algorithmic Trading in Financial Markets. *Expert Syst. Appl.* **2022**, *208*, 118124. [CrossRef]
33. Yu, Y.; Zhai, Z.; Li, W.; Ma, J. Target-Oriented Multi-Agent Coordination with Hierarchical Reinforcement Learning. *Appl. Sci.* **2024**, *14*, 7084. [CrossRef]
34. Shi, F.; Qiu, J.; Gu, X.; Shu, L. Integrating Minds: An Ensemble Approach to Portfolio Optimization. *J. Bank. Finance* **2026**, *192*, 107815. [CrossRef]
35. Huang, Z.; Tanaka, F. MSPM: A Modularized and Scalable Multi-Agent Reinforcement Learning-Based System for Financial Portfolio Management. *PLoS ONE* **2022**, *17*, e0263689. [CrossRef] [PubMed]
36. Bańka, F.; Chudziak, J.A. DeltaHedge: A Multi-Agent Framework for Portfolio Options Optimization. In Proceedings of the PACIS 2025 Proceedings, Kuala Lumpur, Malaysia, 5–9 July 2025; Paper 25.
37. Sharpe, W.F. The Sharpe Ratio. *J. Portf. Manag.* **1994**, *21*, 49–58. [CrossRef]
38. Rockafellar, R.T.; Uryasev, S. Optimization of Conditional Value-at-Risk. *J. Risk* **2000**, *2*, 21–41. [CrossRef]
39. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [CrossRef]
40. Liu, X.Y.; Yang, H.; Gao, J.; Wang, C.D. FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance. In Proceedings of the Second ACM International Conference on AI in Finance, Virtual, 3–5 November 2021; ICAIF '21, pp. 1–9. [CrossRef]
41. Arian, H.; Norouzi Mobarekeh, D.; Seco, L. Backtest Overfitting in the Machine Learning Era: A Comparison of Out-of-Sample Testing Methods in a Synthetic Controlled Environment. *Knowl.-Based Syst.* **2024**, *305*, 112477. [CrossRef]
42. Hahn, J.; Shi, R. Synthetic Control and Inference. *Econometrics* **2017**, *5*, 52. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.