

MemTrial: Learning When to Trust Memory in LLM Portfolio Agents

Guanghao Wu*, Zhuo Cai*, Shoujin Wang[†]
University of Technology Sydney

Abstract

Large language model (LLM) agents for portfolio management learn from experience: they credit each experience in their memory with the outcome of the decisions that used it. In financial markets, however, this outcome mostly reflects the market move shared by all decisions on that date, so the credit tracks the market rather than the experience, and these agents often do worse than simply holding the equal-weight ($1/N$) portfolio. We ask how an agent can measure what an experience changes, and answer by putting memory on trial: drafts of the same decision with and without an experience face the same market, so the outcome they share cancels in their difference. Our agent, MemTrial, generates eight drafts of each decision with subsets of its retrieved experiences chosen by a fractional factorial design, and estimates each experience’s contribution as its Banzhaf value, the average of these differences. As each date occurs once and each draft is a noisy LLM sample, these contributions are noisy and may not hold on unseen dates. MemTrial therefore pools them into values across dates and similar experiences with a hierarchical Bayesian model, acts on the values only after they have predicted unseen dates, and otherwise stays anchored at a conservative reference such as $1/N$. On four benchmarks, it not only benefits from experiences that matter (the best of 15 methods on a semi-synthetic benchmark with known experience quality) but also, through its anchor, limits its losses when its values do not hold (at most 2.2% below $1/N$ on PortBench and InvestorBench, against 15–38% for the best experience-learning agent). Averaged over five benchmark settings, its utility is 21.2% higher than that of the best experience-learning agent, and with eight LLMs it beats every LLM-based baseline on InvestorBench. Our code is available [here](#).

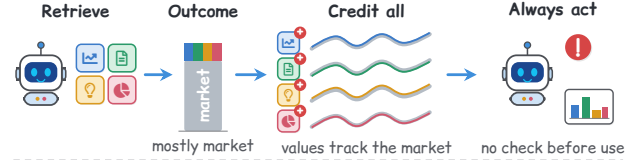
Keywords

LLM agents; agent memory; experience learning; credit assignment; counterfactual attribution; portfolio management

1 Introduction

LLM agents are increasingly used to manage investment portfolios [34, 54, 57]. To improve with use, they learn from experience: when the outcome of a decision is known, the agent writes an experience, such as a short lesson, into its memory, and at later dates it retrieves the most relevant experiences and conditions its decision on them [45, 59]. Which experiences it comes to rely on is decided by *outcome credit*, in one of two ways: an experience used in a decision that turned out well gains importance or value [54, 56], or lessons are distilled from episodes judged by their outcomes to be successes or failures [45, 59].

(a) Existing agents: credit the whole outcome



(b) MemTrial: put memory on trial

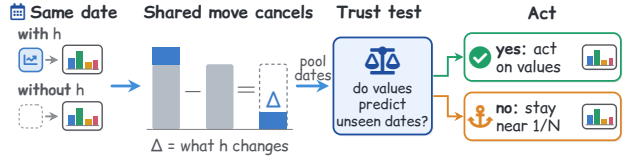


Figure 1: (a) Existing agents credit experiences with market-driven outcomes and always use these credits. (b) MemTrial measures what each experience changes on the same date and acts on its values only after they have predicted unseen dates; otherwise it stays near $1/N$.

In portfolio management, this rule has a blind spot. The outcome of a decision is dominated by the market move over the holding period, and every decision made on that date shares this move. An experience retrieved before a rally is credited with the rally, and one retrieved before a sell-off is blamed for it, much as a treatment effect estimated without a control group mixes the treatment with everything else that happened (Figure 1a). Table 1 measures this on three real benchmarks (§4.1), using the utility of the equal-weight ($1/N$) portfolio over the same holding period as a proxy for the market. On every date, we generate a draft with each subset of the four retrieved experiences and compare two measures for each experience: its outcome credit, the mean utility of the drafts that used it, and its counterfactual contribution, the mean utility of these drafts minus that of the drafts without it. Outcome credit rises and falls with the market almost in lockstep, but is nearly unrelated to the contribution of the experience, which itself is unrelated to the market. As each experience is in half of the drafts, its outcome credit is the date’s mean utility plus half its contribution, and the first term dominates. Agents that accumulate outcome credit therefore rank experiences by the markets in which they were retrieved. Accordingly, every experience-learning agent we test underperforms $1/N$ on PortBench and InvestorBench (Table 3).

Data valuation offers a remedy: it values a data point by its marginal contribution to a utility, averaged over subsets of the other points [18, 50]. Applying it to LLM portfolio agents raises two challenges (CHs).

CH1: How can an agent measure what an experience changes when every measurement is a costly, noisy LLM call on a market date that never comes back? Data valuation evaluates a utility many times on a fixed dataset. For an LLM agent,

*Both authors contributed equally to this research.

[†]Corresponding author.

every evaluation is a paid call whose output varies from sample to sample, and every decision date is a single market episode. A contribution measured on one date is therefore mostly noise: on PortBench, the retrieved experiences change the agent’s utility detectably on only 2–4 of 56 months, about as often as chance alone would produce (Table 1).

Table 1: Outcome credit tracks the market, not the experience.

Benchmark	Spearman correlation (ρ)			Dates with a detectable effect
	Outcome credit vs. $1/N$	Contribution vs. $1/N$	Outcome credit vs. contribution	
PortBench-Full	+0.91 $\pm$ 0.00	+0.04 $\pm$ 0.03	+0.11 $\pm$ 0.06	3–4 / 56
PortBench-Raw	+0.92 $\pm$ 0.00	+0.03 $\pm$ 0.04	+0.04 $\pm$ 0.03	2–4 / 56
InvestorBench	+0.95 $\pm$ 0.00	−0.07 $\pm$ 0.02	−0.03 $\pm$ 0.02	126–127 / 149
ClassAlloc	+0.97 $\pm$ 0.00	+0.02 $\pm$ 0.07	+0.02 $\pm$ 0.07	53–57 / 71

CH2: How can an agent tell whether a learned value will hold on dates it has not seen? Even where experiences clearly change the decision, as on about 85% of InvestorBench days, the contributions learned from past days do not predict those of the following days (§4.2), and selecting experiences by their average past contribution still trails $1/N$ (Table 3). Existing agents always act on their learned scores: they have no notion of insufficient evidence and no fallback for when their scores do not hold.

To address these challenges, we present **MemTrial** (memory on trial), an experience-learning agent that learns when to trust its memory through a closed loop of four steps (Figures 1b and 2). (1) *Counterfactual attribution*: on every date, the agent generates eight drafts of its decision with combinations of its four retrieved experiences chosen by a fractional factorial design [6], and estimates the contribution of each experience as its Banzhaf value [3, 50], the average difference between the drafts with and without it. Because all drafts face the same market, the outcome they share cancels. (2) *Value estimation*: as in small-area estimation [15], a hierarchical Bayesian model pools these noisy contributions into values across dates and across experiences with similar content. (3) *Trust test*: following prequential analysis [10], the agent scores its predicted values against the contributions measured once the outcome is known, and trusts its values only when these out-of-sample forward scores are significantly positive. (4) *Anchored action*: if the values are trusted, the agent deploys the draft with the highest predicted value; otherwise it deploys a blend of its drafts anchored at a conservative reference, such as $1/N$, and learns online how far to move from it. Steps (1)–(2) address CH1, and steps (3)–(4) address CH2.

We evaluate MemTrial on three real benchmarks, PortBench, InvestorBench and ClassAlloc, and on PlantedMem, a semi-synthetic environment with real prices and real LLM drafts in which the quality of every experience is known. **Our contributions are:** (i) we show that outcome credit, the credit rule of existing experience-learning agents, tracks the market rather than what an experience changes; (ii) we propose MemTrial, a closed learning loop that learns what each experience is worth and when this knowledge can be trusted; (iii) against 14 baselines on four benchmarks and with eight LLMs, MemTrial is the best method when experiences matter and stays close to $1/N$ when its values do not hold, where

existing experience-learning agents fall behind. Appendices A–H follow the references.

2 Related Work

LLM agents for financial decision making. LLM agents now trade single assets and allocate across many. FinMem [54] ranks memories by recency, relevance and an importance that trading feedback reinforces; FinCon [55] updates the investment beliefs of a team of agents by verbal reinforcement from realized profit and loss; FinAgent [57] and CryptoTrade [35] reflect on past trades; and TradingAgents [52] simulates the roles of a trading firm. InvestorBench [34] and PortBench [60] evaluate such agents on stocks, cryptocurrencies and multi-asset portfolios. These agents judge a memory, reflection or strategy by the outcome of the trades that used it, which the market move dominates, and none of them checks whether what it has learned holds on later dates.

Experience learning and memory in LLM agents. Outside finance, Reflexion [45] turns feedback on failed attempts into verbal reflections, ExpeL [59] distills insights from successful and failed trajectories, Voyager [49] keeps a library of skills that it reuses when they succeed, and Generative Agents [41], MemoryBank [61] and MemGPT [40] decide which memories to retrieve, keep or forget by recency, importance and relevance. MemRL [56] learns a value for each memory from the rewards of the episodes that used it, and retrieval-augmented generation [32] retrieves by similarity alone. These methods credit outcomes to the memories involved or distill lessons from successes and failures, and never test whether what they learn holds later. This works when outcomes reflect the agent’s own choices, as in games, coding or multi-agent environments, but not when a shock shared by all decisions dominates them.

Counterfactual credit and data valuation. Data Shapley [18, 23], Beta Shapley [29] and Data Banzhaf [50] value a training point by its averaged marginal contributions to a model’s performance, and Data Banzhaf is the most robust of them when the utility is noisy. Influence functions [26] approximate the effect of removing a point. Uplift modelling [21] and counterfactual learning from logged decisions [5, 46] estimate the effect of an action from observational data. These methods value data for a fixed model and test set, or estimate effects from logs in which, in our setting, the market and the choice of memory are confounded. None of them measures what a memory changes in an LLM’s decisions, or asks whether values learned on past dates predict future ones.

Safe deployment and portfolio choice under estimation error. Safe policy improvement deploys a learned policy only when it beats a baseline with high confidence [30, 47], and prequential analysis [10] judges a model by its sequential out-of-sample predictions. In portfolio choice, estimation error makes optimized portfolios hard to trust: none of those evaluated by DeMiguel et al. [12] consistently beats $1/N$ out of sample, which motivates shrinkage towards conservative targets [24, 31, 48]. MemTrial brings these ideas to the memory of LLM agents: a trust test on forward scores decides whether its learned values are trusted, and until then an anchored blend learns online how far to move from a conservative reference.

What is new in MemTrial is not the Banzhaf value [3] but how it is measured and used: on LLM decisions through a designed experiment on each date, pooled into a value model, and trusted

only after this model has predicted unseen dates. Please refer to Appendix A for more related work.

3 MemTrial: Learning When to Trust Memory

3.1 Problem Formulation and Overview

Portfolio decisions under an investor mandate. An agent manages a portfolio of N assets and cash for an investor. At each decision date $t = 1, \dots, T$, it observes the market state s_t and outputs portfolio weights x_t , which are held over a holding period. A projection P_θ enforces the investor’s mandate, a cap M on the share of risky assets and a floor m on the share of defensive assets; the investor profile $\theta = (M, m, \gamma)$ also contains the risk aversion γ . A decision is scored by the investor utility of the executed portfolio, a certainty-equivalent (CEQ) return [12],

$$u_t(x) = R_t(P_\theta(x)) - \frac{\gamma}{2} V_t(P_\theta(x)), \quad (1)$$

where R_t is the realized return over the holding period net of proportional trading costs and V_t is the variance of daily returns times the number of days in that period (for one day, the squared return).

An LLM agent with an experience memory. The agent is an LLM policy π that maps the state and a set of experiences S to weights, $x \sim \pi(\cdot | s_t, S)$. Its memory holds experiences written once the outcomes of earlier decisions were known: lessons, or records of earlier decisions and their outcomes. At date t , the retriever returns the $k = 4$ experiences whose embeddings are most similar to that of the current state, $R_t = (h_{t,1}, \dots, h_{t,k})$. Because the LLM samples at a positive temperature, the same prompt yields different weights; we call each sample a *draft*. Experience learning selects which retrieved experiences enter the prompt, $S_t \subseteq R_t$, based on scores learned from earlier dates. Existing agents credit an experience used at date t with $u_t(x_t)$, the utility of the whole decision [54, 56], which Table 1 shows mostly measures the market.

Our goal and the MemTrial loop. We aim to maximize the investor’s cumulative utility $\sum_t u_t(x_t)$ by learning what each experience is worth, from what it changes in decisions, and when this knowledge can be trusted. Figure 2 shows the loop (Algorithm F.1 in Appendix F gives its pseudocode). Before the outcome of date t is known, MemTrial generates drafts with designed subsets of R_t and acts (§3.5) according to the trust test (§3.4). After the outcome is known, it attributes the change in utility to each experience (§3.2), updates the values (§3.3) and scores their predictions (§3.4).

3.2 Counterfactual Attribution: What Does an Experience Change?

Designed drafts. To know what an experience changes, MemTrial compares decisions made on the same date with and without it, as in a controlled experiment: both face the same market path, so the outcome they share cancels in their difference. Comparing every subset of k experiences would need 2^k drafts, 16 for $k = 4$. A fractional factorial design [6] measures the effect of each experience with half as many drafts: the half fraction

$$\mathcal{D} = \{S \subseteq R_t : |S| \text{ is even}\}, \quad (2)$$

a 2^{4-1} design of resolution IV. Its eight subsets include the empty and the full set, every experience appears in half of them and every

pair in two, so the effect of one experience is not aliased (mixed) with any interaction between two others (Appendix C.1).

Contribution as an estimated Banzhaf value. Let $U_t(S) = u_t(x_t^S)$ be the utility that the draft made with subset S would have earned at realized prices (the agent is a price taker). The counterfactual contribution of the j -th retrieved experience is

$$c_{t,j} = \text{mean}\{U_t(S) : S \in \mathcal{D}, j \in S\} - \text{mean}\{U_t(S) : S \in \mathcal{D}, j \notin S\}, \quad (3)$$

the main effect of experience j in the 2^{4-1} design. Over all 16 subsets, the main effect is the Banzhaf value [3] of j in the game U_t ; in the half fraction, it is aliased with the interaction of the other three experiences, which adds 13–17% to its noise variance on the benchmarks where all 16 subsets were drafted (Appendix C.1). The Banzhaf value weights every context of the other experiences equally, which makes it the most robust semivalue under noisy utilities [50], as for sampled LLM outputs. Any utility shared by all drafts of date t cancels in $c_{t,j}$, and the market move enters it only through the weights that experience j changes. Since U_t is evaluated after the mandate projection, a change that the mandate removes contributes nothing. On the design, the outcome credit of j , the mean utility of the drafts that used it, equals $\bar{U}_t + c_{t,j}/2$, where $\bar{U}_t$, the mean utility of the eight drafts, follows the market (Table 1). Finally, we centre the contributions within each date,

$$y_{t,j} = c_{t,j} - \frac{1}{k} \sum_{j'=1}^k c_{t,j'}, \quad (4)$$

which removes what all retrieved experiences do in common on that date and keeps what distinguishes them.

3.3 Value Estimation: Pooling Noisy Evidence

Hierarchical value model. A contribution measured on a single date is mostly noise, and a new experience has no measurements at all. Small-area estimation faces the same problem when it estimates the income of a small town from a few survey responses, and solves it by borrowing strength from similar towns [15]. Let n_h be the number of dates with a realized outcome on which experience h was retrieved and m_h the mean of its centred contributions. The value model is

$$m_h = v_h + \varepsilon_h, \quad \varepsilon_h \sim \mathcal{N}(0, \sigma^2/n_h), \\ v_h = z_h^\top \beta + b_h, \quad \beta \sim \mathcal{N}(0, \sigma_\beta^2 I), \quad b_h \sim \mathcal{N}(0, \tau^2), \quad (5)$$

where v_h is the value of experience h , $z_h \in \mathbb{R}^d$ a content feature (the first $d = 8$ standardized principal components of its retrieval embedding), β maps content to value, and b_h is the experience’s own deviation. This is the Fay–Herriot model with content as the covariate, or a Gaussian process with a kernel linear in content plus an identity term [42]. The noise variance σ^2 is estimated from the spread of each experience’s contributions, and (σ_β^2, τ^2) maximize the marginal likelihood

$$\log p(m | \sigma_\beta^2, \tau^2) = \log \mathcal{N}(m; 0, K), \\ K = \sigma_\beta^2 Z Z^\top + \tau^2 I + \sigma^2 \text{diag}(1/n_h), \quad (6)$$

over a grid. The posterior means are

$$\hat{v} = (\sigma_\beta^2 Z Z^\top + \tau^2 I) K^{-1} m, \quad \hat{\beta} = \sigma_\beta^2 Z^\top K^{-1} m, \quad (7)$$

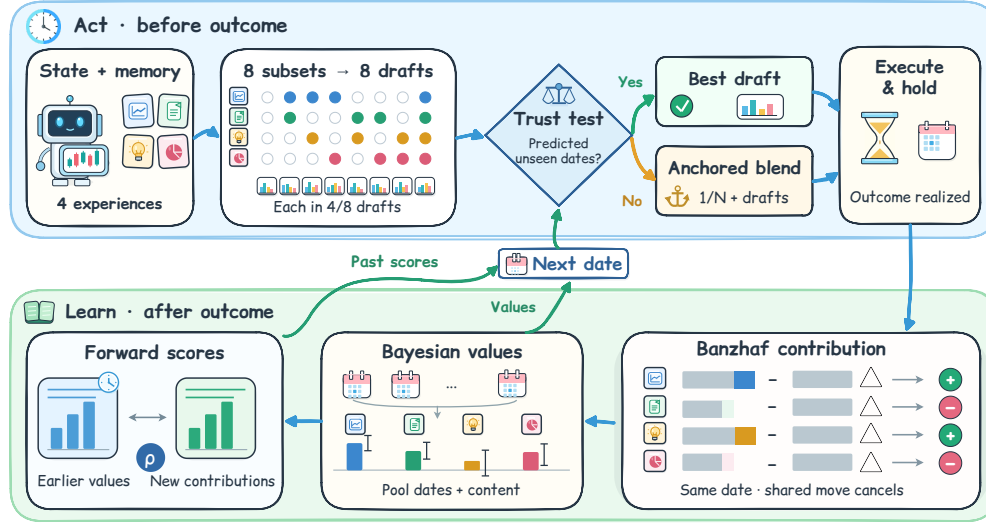


Figure 2: One decision date of MemTrial. Before the outcome, it acts on its learned values only if they have predicted unseen dates; after the outcome, it updates the values and scores their predictions. Contributions compare drafts of one date, so their shared market move cancels.

so an experience with few measurements is shrunk towards the value predicted from its content, and a new one receives $\hat{v}_h = z_h^\top \hat{\beta}$ before it is ever used.

Learner selection. Content may or may not predict value, so MemTrial keeps two learners, the content-aware model (5) and a per-experience model with $\sigma_\beta^2 = 0$. It scores both forward (§3.4) and uses the one with the higher mean forward score once both have at least 10 scores (before that, the per-experience model).

3.4 Trust Test: Has the Learned Value Predicted the Future?

Forward scores. A weather forecaster is judged by how well she predicted tomorrow’s weather, not by how well her model fits yesterday’s, and prequential analysis judges a statistical model in the same way [10]. Before date t , each learner is fitted only on dates whose outcomes are known and predicts the values $\hat{v}_{t,j}$ of the experiences retrieved at t . Once the outcome is known, the prediction is scored against the measured centred contributions,

$$\rho_t = \text{corr}((\hat{v}_{t,1}, \dots, \hat{v}_{t,k}), (y_{t,1}, \dots, y_{t,k})), \quad (8)$$

the Pearson correlation across the retrieved experiences, which is out of sample by construction.

Trust test. Let $\rho_1, \dots, \rho_n$ be the forward scores of the selected learner available at date t , with mean $\bar{\rho}$ and standard deviation s_ρ . As in safe policy improvement [47], MemTrial trusts its values only on strong evidence: when $n \geq 10$ and

$$p_t = \Pr\left(T_{n-1} \geq \frac{\bar{\rho}}{s_\rho/\sqrt{n}}\right) < \alpha, \quad (9)$$

where T_{n-1} follows a Student t distribution with $n - 1$ degrees of freedom and $\alpha = 0.05$. Trust thus means that the values have ranked the retrieved experiences better than chance on unseen

dates; used as a decision rule on every date, the test rarely passes where experiences carry no signal (§4.5). When trusted, MemTrial deploys the designed draft with the highest total predicted value,

$$S_t^* = \arg \max_{S \in \mathcal{D}} \sum_{j \in S} \hat{v}_{t,j}, \quad (10)$$

with ties broken towards fewer experiences, provided that this total is positive, i.e., that the subset’s experiences are predicted to beat the average retrieved one; §4.5 measures whether trusted drafts beat the anchored blend. This draft was already generated for Eq. (3), so trusting costs no extra LLM call. The test asks whether the values have predicted the future, not whether experiences differed in the past (§4.3 compares the two).

3.5 Anchored Action: What to Do Until Memory Can Be Trusted

Anchored blend. The $1/N$ portfolio is hard to beat out of sample [12], and combining it with a more ambitious strategy, weighted by how strongly the data support the latter, can improve on both [48]. MemTrial applies this idea online. When its values are not trusted, it deploys a blend of a conservative reference x^{ref} (on the real benchmarks, the $1/N$ portfolio projected to the mandate) and the average draft $\bar{x}_t = \frac{1}{|\mathcal{D}|} \sum_{S \in \mathcal{D}} x_t^S$,

$$x_t(q) = q x_t^{\text{ref}} + (1 - q) \bar{x}_t,$$

$$q_t = \arg \max_{q \in [0,1]} \frac{1}{n} \sum_{s < t} u_s(x_s(q)) - \lambda_n \text{KL}(q \| \alpha_0), \quad \lambda_n = \frac{\lambda_0 \hat{\sigma}_n}{\sqrt{n}}, \quad (11)$$

where $\text{KL}(q \| \alpha_0)$ is the Kullback–Leibler divergence between Bernoulli distributions, $\alpha_0 = 0.9$ is the prior weight on the reference, $\lambda_0 = 4$, n is the number of dates with a realized outcome, and $\hat{\sigma}_n$, half the mean absolute utility gap between the reference and the average draft, makes the regularizer unit-free. This is

follow-the-regularized-leader [43] with a regularizer centred at the prior: with little evidence, q_t stays close to α_0 ; as evidence accumulates, the regularization weakens as $1/\sqrt{n}$ and q_t follows the blend that performed best on these dates.

3.6 The Closed Loop and Its Cost

The same eight drafts serve every step: they yield the contributions (3), the trusted action (10) and the average draft in (11). MemTrial therefore makes $|\mathcal{D}| = 8$ drafts per decision, which run in parallel; its LLM cost is eight times that of a single-draft agent. Fitting the value model costs $O(G \cdot H^3)$ for G grid points and H experiences, a few milliseconds per decision (Appendix D.3).

4 Experiments

4.1 Experimental Setup

Table 2: Three real benchmarks and one semi-synthetic benchmark with planted experiences of known quality.

	PortBench	InvestorBench	ClassAlloc	PlantedMem
Assets	99 (6 classes)	4 stocks, cash	5 classes	4 classes
Decisions	monthly	daily	monthly	monthly
Development	2019–22	none	none	15 worlds
Test dates	20 (2023–24)	149 (10/20–05/21)	71 (05/20–03/26)	120 / world
Experiences	338 decisions	63 lessons	58 lessons	12/48 planted
Seeds	3	3	3	100 worlds

Benchmarks. Table 2 summarizes the four benchmarks, and Appendix D.1 gives their construction. On *PortBench* [60], each decision starts from the equal-weight portfolio, holds for 20 trading days and pays 15 bp on the amount traded. PortBench’s native prompt shows prices for only eight of its 99 assets: PortBench-Raw keeps this prompt, and PortBench-Full shows every eligible asset; the prompts of the other benchmarks show every asset. On *InvestorBench* [34], the agent also reads daily news and writes its lessons during a warm-up from July to September 2020. *ClassAlloc* tests agents after the LLM’s knowledge cutoff: built from PortBench’s prices, it ends 21 months after that of gpt-4.1-mini [38] (June 2024), carries holdings from month to month, and gives every learner an outcome only after its 20 trading days. Experiences are retrieved by embedding similarity, with Qwen3-Embedding-0.6B [58] on PortBench and text-embedding-3-small [37] on InvestorBench and ClassAlloc. *PlantedMem* keeps real prices and real LLM drafts but plants twelve experiences of known quality: three useful, six neutral and three misleading. A useful (misleading) experience moves a share β of the draft to an asset class, with probability 0.7 the one that does best (worst) next (Appendix D.3). Retrieval depends on the market regime, which reproduces the market confound. Five core regimes vary how strongly experiences move a draft (no influence, noise only, and $\beta \in \{0.1, 0.25, 0.5\}$), and two more have 48 experiences whose content is uninformative (*many experiences*) or informative (*informative content*) about their quality.

Investors and metric. We use PortBench’s conservative ($M = 0.40$, $m = 0.40$, $\gamma = 10$), balanced ($M = 0.65$, $m = 0.20$, $\gamma = 5$) and aggressive ($M = 0.90$, $m = 0.05$, $\gamma = 2$) investors. We report the utility (1) per decision period, averaged over dates and investors, in % per month (bp per day on InvestorBench); higher is better.

Baselines (Appendix D.2). (i) Rule-based: $1/N$ projected to the mandate [12] and Minimum variance [36]. (ii) LLM agents without experience learning: Memory-free (zero-shot) [7], Self-consistency [51] (the average of three memory-free drafts) and Similarity retrieval [32] of the two or four most similar experiences. (iii) Experience-learning agents: FinMem [54], MemRL [56], Reflexion [45] and ExpeL [59]. (iv) Credit-based and ensemble learners: Uplift credit [21], which regresses outcomes on experience usage with a market control; Counterfactual selection, which selects experiences by our contributions (3) without value estimation or trust test; Draft averaging [4], which deploys the average of MemTrial’s eight drafts; and Hedge [16] over $1/N$ and the memory configurations. Where a baseline’s own module cannot run, we re-implement its memory-scoring rule (marked †); on PortBench, ExpeL† votes over the experiences, and the FinMem and MemRL modules receive no feedback after July 2019 (Appendix D.2).

Implementation and statistics. All LLM agents use gpt-4.1-mini (2025-04-14) at temperature 0.7, which gives the distinct drafts that MemTrial, Self-consistency and Draft averaging need. MemTrial uses $k = 4$, the design (2), $d = 8$, $\alpha = 0.05$ with at least 10 forward scores, $\alpha_0 = 0.9$ and $\lambda_0 = 4$. Its reference is $1/N$ on the real benchmarks and the average of eight memory-free drafts in PlantedMem, where $1/N$ ignores the risk aversion. It was frozen before any InvestorBench decision, so InvestorBench, ClassAlloc and the runs with other LLMs are held-out tests. On the real benchmarks, each of three seeds is an independent deployment in which every draft is sampled anew; PlantedMem uses 100 test seeds, each a new world. Tables report the mean $\pm$ sd over seeds, with the best result in **bold** and the second best underlined. *Improv.* is the gain of MemTrial over the best experience-learning agent of each setting, in percent of that agent’s utility, and * marks a significant improvement ($p < 0.05$, one-sided paired t -test over test dates, or over seeds in PlantedMem). The *trust rate* is the share of decisions on which MemTrial acts on its learned values. Appendix D.3 gives further implementation details and Appendix G the prompts.

4.2 Main Results

Overall performance. Table 3 (PlantedMem: mean of the five core regimes) shows that MemTrial is the best of all 15 methods in PlantedMem and the best LLM-based method on PortBench and InvestorBench. Averaged over the five settings, its utility is 21.2% higher than that of the best experience-learning agent. In PlantedMem, where experiences carry real signal, the gain is 0.10 pp per month. On PortBench and InvestorBench, no LLM-based method beats $1/N$: the LLM’s drafts fail to outperform equal weighting, and their frequent reallocation costs fees and adds risk (Appendix D.4). MemTrial is the only LLM-based method that stays close to $1/N$ there, at most 2.2% below it, whereas the best existing experience-learning agent falls 15–38% behind. On PortBench-Full, the lead of $1/N$ and MemTrial depends on the last test month (November 2024), in which cryptocurrencies rose by 129%: without it, Similarity retrieval (top-4) and Hedge come out slightly ahead of both (Appendix D.6). On ClassAlloc, the memory-free LLM beats $1/N$, but no form of memory improves on it (Appendix D.11); §4.5 shows that this advantage lies in the months before the LLM’s knowledge cutoff (June 2024).

Table 3: Main results: utility per decision period (higher is better). MemTrial is the best LLM-based method on PortBench and InvestorBench, where it stays within 2.2% of $1/N$, and the best method in PlantedMem.

	Method	PortBench-Full %/month	PortBench-Raw %/month	InvestorBench bp/day	ClassAlloc %/month	PlantedMem %/month
Rule-based	$1/N$ [12]	2.02 ± 0.00	2.02 ± 0.00	10.45 ± 0.00	0.623 ± 0.000	0.57 ± 0.00
	Minimum variance [36]	0.25 ± 0.00	0.25 ± 0.00	5.53 ± 0.00	0.230 ± 0.000	-0.04 ± 0.00
LLM w/o experience learning	Memory-free [7]	1.60 ± 0.05	1.07 ± 0.02	8.43 ± 0.29	<u>0.657 ± 0.001</u>	0.76 ± 0.08
	Self-consistency [51]	1.56 ± 0.11	1.17 ± 0.03	7.89 ± 0.21	0.686 ± 0.007	0.79 ± 0.03
	Similarity retrieval (top-2) [32]	1.43 ± 0.19	1.24 ± 0.29	8.67 ± 0.19	0.649 ± 0.033	0.72 ± 0.07
	Similarity retrieval (top-4) [32]	1.70 ± 0.06	1.05 ± 0.02	8.12 ± 0.61	0.643 ± 0.039	0.72 ± 0.06
Experience- learning agents	FinMem [54]	1.63 ± 0.31	1.26 ± 0.14	$8.47 \pm 0.17^\dagger$	$0.594 \pm 0.017^\dagger$	$0.75 \pm 0.06^\dagger$
	MemRL [56]	1.58 ± 0.29	1.05 ± 0.06	$8.90 \pm 0.39^\dagger$	$0.633 \pm 0.012^\dagger$	$0.74 \pm 0.08^\dagger$
	Reflexion [45]	1.40 ± 0.18	1.26 ± 0.03	7.29 ± 0.66	0.590 ± 0.061	$0.78 \pm 0.06^\dagger$
	ExpEL [59]	$1.53 \pm 0.05^\dagger$	$1.10 \pm 0.06^\dagger$	8.93 ± 0.32	0.609 ± 0.006	$0.77 \pm 0.06^\dagger$
Credit-based and ensemble learners	Uplift credit [21]	$1.38 \pm 0.09^\dagger$	$1.20 \pm 0.10^\dagger$	$8.44 \pm 0.25^\dagger$	$0.645 \pm 0.022^\dagger$	$0.78 \pm 0.09^\dagger$
	Counterfactual selection [50]	1.49 ± 0.29	1.30 ± 0.34	8.54 ± 0.26	0.650 ± 0.033	<u>0.85 ± 0.08</u>
	Draft averaging [4]	1.62 ± 0.05	1.19 ± 0.02	8.69 ± 0.09	0.651 ± 0.008	0.75 ± 0.03
	Hedge [16]	1.73 ± 0.09	1.59 ± 0.04	8.97 ± 0.23	0.634 ± 0.008	$0.75 \pm 0.03^\dagger$
MemTrial (ours)		<u>1.98 ± 0.00</u>	<u>2.01 ± 0.00</u>	<u>10.40 ± 0.04</u>	0.613 ± 0.017	$0.87 \pm 0.04^*$
Improv. (%)		$+21.3 \pm 19.3$	$+59.0 \pm 2.6$	$+16.4 \pm 3.2$	-3.2 ± 1.4	$+12.6 \pm 10.1$
Trust rate of MemTrial (%)		0.0 ± 0.0	0.6 ± 1.0	0.3 ± 0.5	14.6 ± 21.7	33.7 ± 9.2

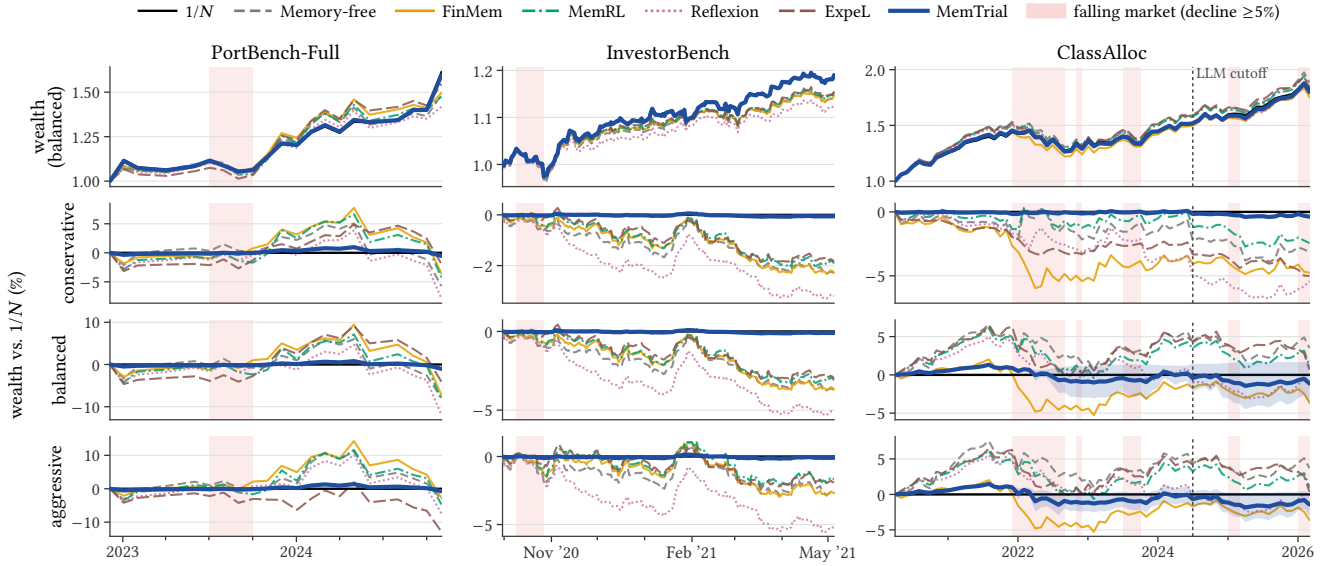


Figure 3: Wealth of the balanced investor (top, start = 1) and relative to $1/N$ for the conservative, balanced and aggressive investors (rows 2–4). For every investor, MemTrial stays close to $1/N$, while the other LLM-based methods drift away, mostly below.

Wealth over time. Figure 3 follows the wealth of the three investors on PortBench-Full, InvestorBench and ClassAlloc (red shading marks falling markets and the dotted line the knowledge cutoff); on PortBench, whose decisions each start from the equal-weight portfolio, wealth compounds their net returns. For every investor, MemTrial tracks $1/N$ through rising and falling markets, while the other LLM-based methods drift away from it, mostly below. In Table 4, MemTrial has the best Sharpe ratio and draw-down of the LLM-based methods on every benchmark, and the highest return on PortBench and InvestorBench, where no LLM-based method has a higher return than $1/N$. Across all 162 runs

(both PortBench variants and eight LLMs on InvestorBench and ClassAlloc, each with three investors and three seeds), the wealth of MemTrial never fell more than 5% behind that of $1/N$, whereas the memory-free LLM and the experience-learning agents did so in 22–44% of their runs (Appendix D.6).

Why does MemTrial help? Three components account for this. First, counterfactual attribution removes the market from the contribution. In PlantedMem, counterfactual selection beats selection by outcome credit (MemRL †), whereas adjusting outcome credit for the market by regression (Uplift credit) closes only a third of this gap. Second, the trust test keeps MemTrial from acting on

Table 4: Return and risk of the balanced investor: cumulative net return (CR), annualized Sharpe ratio (SR) and maximum drawdown (MDD). On PortBench and InvestorBench, MemTrial is the only LLM-based method close to 1/N in both return and Sharpe ratio.

Method		PortBench-Full			InvestorBench			ClassAlloc		
		CR % $\uparrow$	SR $\uparrow$	MDD % $\downarrow$	CR % $\uparrow$	SR $\uparrow$	MDD % $\downarrow$	CR % $\uparrow$	SR $\uparrow$	MDD % $\downarrow$
Rule-based	1/N [12]	61.2 $\pm$ 0.0	1.77 $\pm$ 0.00	5.6 $\pm$ 0.0	19.1 $\pm$ 0.0	2.87 $\pm$ 0.00	5.7 $\pm$ 0.0	82.0 $\pm$ 0.0	<u>1.22 $\pm$ 0.00</u>	<u>12.0 $\pm$ 0.0</u>
	Minimum variance [36]	5.4 $\pm$ 0.0	1.19 $\pm$ 0.00	1.5 $\pm$ 0.0	9.9 $\pm$ 0.0	1.47 $\pm$ 0.00	6.4 $\pm$ 0.0	19.1 $\pm$ 0.0	4.19 $\pm$ 0.00	0.2 $\pm$ 0.0
LLM w/o exp. learning	Memory-free [7]	48.5 $\pm$ 0.8	1.47 $\pm$ 0.06	7.3 $\pm$ 1.0	14.6 $\pm$ 0.5	2.24 $\pm$ 0.08	6.3 $\pm$ 0.1	<u>88.3 $\pm$ 1.5</u>	1.11 $\pm$ 0.00	13.9 $\pm$ 0.3
Experience-learning agents	FinMem [54]	49.6 $\pm$ 9.2	1.53 $\pm$ 0.21	6.2 $\pm$ 1.4	14.7 $\pm$ 0.4	2.27 $\pm$ 0.03	6.1 $\pm$ 0.1	75.4 $\pm$ 2.6	1.07 $\pm$ 0.03	15.4 $\pm$ 0.5
	MemRL [56]	47.7 $\pm$ 11.1	1.45 $\pm$ 0.19	6.9 $\pm$ 1.1	15.4 $\pm$ 0.7	2.33 $\pm$ 0.09	6.1 $\pm$ 0.3	84.0 $\pm$ 1.2	1.06 $\pm$ 0.03	15.4 $\pm$ 1.1
	Reflexion [45]	41.8 $\pm$ 6.1	1.41 $\pm$ 0.14	7.2 $\pm$ 1.4	12.9 $\pm$ 1.0	1.97 $\pm$ 0.14	6.3 $\pm$ 0.1	80.6 $\pm$ 8.3	1.10 $\pm$ 0.08	15.1 $\pm$ 1.8
	ExpeL [59]	54.6 $\pm$ 3.3	1.61 $\pm$ 0.21	6.8 $\pm$ 1.1	15.5 $\pm$ 0.5	2.43 $\pm$ 0.07	<u>5.8 $\pm$ 0.1</u>	83.2 $\pm$ 0.4	1.04 $\pm$ 0.01	16.9 $\pm$ 0.3
Credit / ensemble	Counterfactual selection [50]	46.9 $\pm$ 8.9	1.44 $\pm$ 0.15	6.1 $\pm$ 1.1	14.8 $\pm$ 0.5	2.29 $\pm$ 0.07	6.0 $\pm$ 0.0	88.6 $\pm$ 5.1	1.10 $\pm$ 0.04	14.0 $\pm$ 1.2
	Hedge [16]	51.3 $\pm$ 3.2	<u>1.64 $\pm$ 0.07</u>	5.6 $\pm$ 0.3	15.8 $\pm$ 0.3	2.43 $\pm$ 0.05	6.0 $\pm$ 0.0	83.9 $\pm$ 1.6	1.10 $\pm$ 0.01	14.2 $\pm$ 0.1
MemTrial (ours)		<u>59.6 $\pm$ 0.1</u>	1.77 $\pm$ 0.01	<u>5.5 $\pm$ 0.0</u>	<u>19.0 $\pm$ 0.1</u>	<u>2.86 $\pm$ 0.02</u>	5.7 $\pm$ 0.0	79.9 $\pm$ 5.1	1.14 $\pm$ 0.10	13.1 $\pm$ 1.5

	PortBench		Investor-Bench	Class-Alloc	PlantedMem			
	Full	Raw			no signal	signal	inform. content	
per-experience learner	=	-0.01	=	=	+0.2	-0.3	-5.3	values
content-aware learner	=	-3.4	=	=	+0.1	-0.3	+0.5	values
in-sample F-test	=	-0.01	-0.3	+2.6	+0.2	-0.8	-5.6	test
none (always act)	-4.2	-11.8	-4.7	-0.1	-2.3	+0.9	+2.4	test
never passes	=	-0.01	-0.3	+2.6	+0.4	-15.4	-8.5	test
draft average ($q=0$)	-18.3	-40.7	-16.4	+5.8	-4.8	-2.9	-2.9	anchor
uniform prior ($\alpha_0=0.5$)	-7.3	-4.6	-6.6	+3.2	-1.0	-0.8	-0.8	anchor
no online learning	+0.9	-3.1	-0.1	+0.1	-0.1	-0.1	-0.05	anchor

Figure 4: Ablation: change in utility when one step of MemTrial is changed (% , mean over seeds; =: same actions; bold: $p < 0.05$). On real markets, the anchor and the trust test matter most; with signal, acting on the learned values does.

values that do not carry over to unseen dates. On PortBench and InvestorBench, the values learned from past dates did not predict the following dates, so the test almost never passed (on at most 0.6% of decisions, Table 3). Counterfactual selection, which acts on past contributions without such a test, trails 1/N as much as the other agents. Third, the anchor protects MemTrial on these benchmarks. The drafts alone do not: Draft averaging deploys the same eight drafts and falls well behind 1/N. With the same anchor, however, every LLM-based baseline also comes within 1% of MemTrial. With signal in PlantedMem, MemTrial earns 11% more than the best anchored baseline, as it acts fully on values that have predicted unseen dates (Appendix D.5).

4.3 Ablation Study

Figure 4 changes one step of MemTrial at a time; PlantedMem is split into the core regimes without signal (no influence, noise only), those with signal ($\beta > 0$), and the regime with 48 experiences and informative content. On real markets, the anchor and the trust test matter: replacing the anchored blend by the draft average costs 16–41% on PortBench and InvestorBench, and acting on the values without the trust test 4–12%, as these values do not hold on unseen

dates. The prior weight on the reference sets a trade-off: a uniform prior loses 5–7% on PortBench and InvestorBench but gains 3.2% on ClassAlloc, where the drafts beat 1/N, and the opposite extreme, always keeping the reference, gains where 1/N cannot be beaten but loses where the drafts are better (Appendix D.8). Learning the blend online matters less: fixing it at the prior changes the utility by -3% to +1%. On ClassAlloc, where the test passed most often, never acting on values is 2.6% better (Appendices D.5 and E).

In PlantedMem, the learned values matter where experiences carry signal: never acting on them costs 15% (8.5% with informative content), and almost nothing without signal. The trust test costs little there: removing it gains 0.9–2.4% with signal but loses 2.3% without. Content-aware value estimation and the trust test matter when each experience is retrieved too rarely for per-experience estimates or for an in-sample F-test (5–6% with informative content).

4.4 Robustness to the LLM

We repeat InvestorBench and ClassAlloc with seven other LLMs: gpt-4.1-nano [38], gpt-5-mini [39], Gemini 2.5 Flash [8], Claude Haiku 4.5 [1], and the open-weight Llama-3.3-70B [20], Qwen3-235B [53] and DeepSeek-V3.1 [11]. We also run InvestorBench at temperatures 0, 0.3 and 1.0. Appendix D.12 gives every LLM, temperature and method. On InvestorBench, no LLM-based baseline beats 1/N in any of the eleven configurations, and MemTrial beats all of them in every configuration, by 4–33% over the best experience-learning agent. On ClassAlloc, MemTrial stays within 0.021 pp per month of 1/N with every LLM; it beats the best experience-learning agent with four of the eight LLMs and trails it with the other four. Without its trust test, MemTrial does worse in most configurations.

4.5 Analysis

Does MemTrial learn what each experience is worth? In PlantedMem, the quality of every experience is known, so we can check how well the learned values rank them (Figure 5; balanced investor). Outcome credit hardly ranks them: after ten years of monthly decisions, the correlation is 0.08 at $\beta = 0.25$ and only 0.29 even at $\beta = 0.5$, because it follows the markets in which an experience was retrieved. Counterfactual contributions rank them well, and the pooled values of MemTrial reach the same level (0.81 at $\beta = 0.25$); with 48 experiences whose content is informative, pooling across

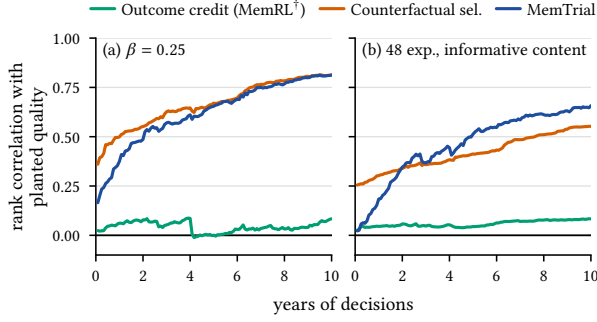


Figure 5: Outcome credit does not learn which experiences are useful; counterfactual contributions do, and pooling them by content helps when content is informative.

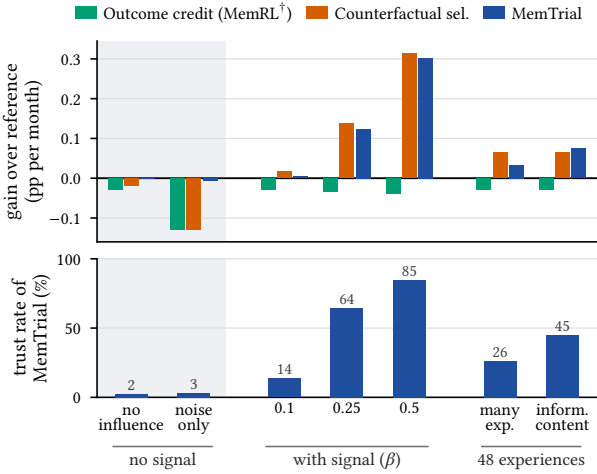


Figure 6: PlantedMem by regime. MemTrial abstains when experiences carry no signal and acts on its values when they matter.

similar experiences ranks them better than the raw contributions (0.66 against 0.55). Early on, the shrunk values are less accurate and MemTrial rarely acts on them; at $\beta = 0.25$, its trust rate rises from 12% in the first year to 91% in the tenth as evidence accumulates.

When does MemTrial trust its values? Figure 6 shows PlantedMem by regime. When experiences carry no signal, the trust test passes on only 2–3% of decisions and MemTrial stays at the reference, whereas Counterfactual selection and outcome credit fall below it. As experiences matter more, the test passes more often, on up to 85% of decisions, and MemTrial gains up to 0.30 pp per month over the reference. On the real benchmarks, the test responds to whether learned values carry over to unseen dates, not to whether experiences matter: on InvestorBench, experiences change the decision on 85% of days, yet the test passed on only 0.3% of decisions. Where it passed most often (15% of ClassAlloc decisions), the trusted drafts earned 0.05 pp less (s.e. 0.07) than the anchored blend they replaced (Appendices D.11 and E).

What happens after the LLM’s knowledge cutoff? On ClassAlloc, the memory-free LLM beats 1/N by 0.08 pp per month

before the knowledge cutoff of gpt-4.1-mini and trails it by as much afterwards ($p = 0.36$ for the difference), and five of the six LLMs whose cutoff is published show the same pattern. Given only a date, each LLM recalls the asset-class returns of the following month before its cutoff (a look-ahead bias) but hardly at all after it; yet hiding every date from the agent does not remove the advantage before the cutoff (Appendices D.13–D.14, Figure D.6). Anchored at 1/N, MemTrial gives up most of the drafts’ advantage before the cutoff but loses less than any other LLM-based method after it (Appendix D.11). On PortBench extended to 2026, it is again the best LLM-based method over the eleven months after the cutoff of gpt-4.1-mini (Appendix D.15).

Hyperparameters and cost. Appendix D.7 varies the level α of the trust test (Figure D.4). Up to the default $\alpha = 0.05$, the results on the real benchmarks barely change. Looser levels let chance forward scores pass, on up to 17% of PortBench-Full decisions, and cost up to 4% of the utility, whereas in the informative regime of PlantedMem they let MemTrial act earlier on values that do hold. The anchor needs a strong prior on the reference (§4.3) but not a precise strength, and more retrieved experiences help only when experiences matter (Appendices D.7–D.9). A decision costs about USD 0.05 on PortBench; at the same LLM spend, Draft averaging does clearly worse.

5 Conclusion

When an LLM portfolio agent credits an experience with the outcome of the decisions that used it, the credit tracks the market rather than the experience. MemTrial instead measures what each experience changes with a small designed experiment on every date, pools these measurements into values across dates and similar experiences, trusts the values only after they have predicted unseen dates, and until then keeps an anchored blend near a conservative reference. Where the quality of experiences is known, it achieves the highest utility among 15 methods. On real markets, where the learned values did not carry over to unseen dates, it stayed close to 1/N, unlike existing experience-learning agents.

Limitations. On the real benchmarks, MemTrial’s values rarely passed the trust test, and where they did (ClassAlloc), acting on them did not beat the anchor. Its advantage there comes from the anchor, which protects the baselines just as well, so gains from memory on real markets remain to be shown. LLMs recall market outcomes from before their knowledge cutoffs, a look-ahead bias that can flatter any LLM-based method on earlier periods; PlantedMem and the months after the cutoff are free of it. When the LLM’s drafts beat the reference, as on ClassAlloc before the cutoff, the anchored action gives up most of their advantage (Appendix H).

References

- [1] Anthropic. 2025. Claude Haiku 4.5 system card. <https://www.anthropic.com/system-cards>.
- [2] Susan Athey and Guido Imbens. 2016. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences* 113, 27 (2016), 7353–7360.
- [3] John F Banzhaf III. 1964. Weighted voting doesn’t work: A mathematical analysis. *Rutgers L. Rev.* 19 (1964), 317.
- [4] John M Bates and Clive WJ Granger. 1969. The combination of forecasts. *Journal of the operational research society* 20, 4 (1969), 451–468.
- [5] Léon Bottou, Jonas Peters, Joaquin Quiñero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013.

- Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research* 14, 11 (2013).
- [6] George EP Box, J Stuart Hunter, and William Gordon Hunter. 2005. *Statistics for Experimenters: Design, Innovation, and Discovery* (2nd ed.). Wiley.
 - [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
 - [8] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261* (2025).
 - [9] Thomas M Cover. 1991. Universal portfolios. *Mathematical finance* 1, 1 (1991), 1–29.
 - [10] A Philip Dawid. 1984. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)* 147, 2 (1984), 278–290.
 - [11] DeepSeek-AI. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437* (2024).
 - [12] Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. 2009. Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *The review of Financial studies* 22, 5 (2009), 1915–1953.
 - [13] Francis X Diebold and Robert S Mariano. 2002. Comparing predictive accuracy. *Journal of Business & economic statistics* 20, 1 (2002), 134–144.
 - [14] Bradley Efron and Carl Morris. 1975. Data analysis using Stein’s estimator and its generalizations. *J. Amer. Statist. Assoc.* 70, 350 (1975), 311–319.
 - [15] Robert E Fay III and Roger A Herriot. 1979. Estimates of income for small places: an application of James-Stein procedures to census data. *J. Amer. Statist. Assoc.* 74, 366a (1979), 269–277.
 - [16] Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55, 1 (1997), 119–139.
 - [17] Lorenzo Garlappi, Raman Uppal, and Tan Wang. 2007. Portfolio selection with parameter and model uncertainty: A multi-prior approach. *The Review of Financial Studies* 20, 1 (2007), 41–81.
 - [18] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*. PMLR, 2242–2251.
 - [19] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 477 (2007), 359–378.
 - [20] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).
 - [21] Pierre Gutierrez and Jean-Yves G  rardy. 2017. Causal inference and uplift modelling: A review of the literature. In *International conference on predictive applications and APIs*. PMLR, 1–13.
 - [22] David P Helmbold, Robert E Schapire, Yoram Singer, and Manfred K Warmuth. 1998. On-line portfolio selection using multiplicative updates. *Mathematical finance* 8, 4 (1998), 325–347.
 - [23] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve G  rel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards efficient data valuation based on the shapley value. In *The 22nd international conference on artificial intelligence and statistics*. PMLR, 1167–1176.
 - [24] Philippe Jorion. 1986. Bayes-Stein estimation for portfolio analysis. *Journal of Financial and Quantitative analysis* 21, 3 (1986), 279–292.
 - [25] Raymond Kan and Guofu Zhou. 2007. Optimal portfolio choice with parameter uncertainty. *Journal of Financial and Quantitative Analysis* 42, 3 (2007), 621–656.
 - [26] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*. Pmlr, 1885–1894.
 - [27] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. 2009. Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery* 18, 1 (2009), 140–181.
 - [28] S  ren R K  nzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences* 116, 10 (2019), 4156–4165.
 - [29] Yongchan Kwon and James Zou. 2021. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. *arXiv preprint arXiv:2110.14049* (2021).
 - [30] Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. 2019. Safe policy improvement with baseline bootstrapping. In *International conference on machine learning*. PMLR, 3652–3661.
 - [31] Olivier Ledoit and Michael Wolf. 2003. Honey, I shrunk the sample covariance matrix. (2003).
 - [32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K  ttler, Mike Lewis, Wen-tau Yih, Tim Rock-t  schel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.
 - [33] Bin Li and Steven C. H. Hoi. 2014. Online Portfolio Selection: A Survey. *Comput. Surveys* 46, 3, Article 35 (2014).
 - [34] Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, Kp Subbalakshmi, Jimin Huang, et al. 2025. Investorbench: A benchmark for financial decision-making tasks with llm-based agent. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2509–2525.
 - [35] Yuan Li, Bingqiao Luo, Qian Wang, Nuo Chen, Xu Liu, and Bingsheng He. 2024. CryptoTrade: A reflective LLM-based agent to guide zero-shot cryptocurrency trading. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 1094–1106.
 - [36] Harry M Markowitz. 1955. *Portfolio selection*. Ph. D. Dissertation. The University of Chicago.
 - [37] OpenAI. 2024. New embedding models and API updates. <https://openai.com/index/new-embedding-models-and-api-updates/>.
 - [38] OpenAI. 2025. Introducing GPT-4.1 in the API. <https://openai.com/index/gpt-4-1/>.
 - [39] OpenAI. 2025. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267* (2025).
 - [40] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. 2023. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560* (2023).
 - [41] Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.
 - [42] Matthias Seeger. 2004. Gaussian processes for machine learning. *International journal of neural systems* 14, 02 (2004), 69–106.
 - [43] Shai Shalev-Shwartz. 2012. Online learning and online convex optimization. *Foundations and Trends   in Machine Learning* 4, 2 (2012), 107–194.
 - [44] Lloyd S Shapley et al. 1953. A value for n-person games. (1953).
 - [45] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems* 36 (2023), 8634–8652.
 - [46] Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *International conference on machine learning*. PMLR, 814–823.
 - [47] Philip Thomas, Georgios Theodorou, and Mohammad Ghavamzadeh. 2015. High confidence policy improvement. In *International Conference on Machine Learning*. PMLR, 2380–2388.
 - [48] Jun Tu and Guofu Zhou. 2011. Markowitz meets Talmud: A combination of sophisticated and naive diversification strategies. *Journal of Financial Economics* 99, 1 (2011), 204–215.
 - [49] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* (2023).
 - [50] Jiachen T Wang and Ruoxi Jia. 2023. Data banzha: A robust data valuation framework for machine learning. In *International conference on artificial intelligence and statistics*. PMLR, 6388–6421.
 - [51] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).
 - [52] Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2024. TradingAgents: Multi-agents LLM financial trading framework. *arXiv preprint arXiv:2412.20138* (2024).
 - [53] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).
 - [54] Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Jordan W Suchow, Denghui Zhang, and Khaldoun Khashanah. 2025. Finmem: A performance-enhanced llm trading agent with layered memory and character design. *IEEE Transactions on Big Data* (2025).
 - [55] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan W Suchow, Zhenyu Cui, Rong Liu, et al. 2024. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems* 37 (2024), 137010–137045.
 - [56] Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, et al. 2026. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory. *arXiv preprint arXiv:2601.03192* (2026).
 - [57] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. 2024. 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*. 4314–4325.
 - [58] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv*

- preprint *arXiv:2506.05176* (2025).
- [59] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19632–19642.
- [60] Yuxuan Zhao, Sijia Chen, and Ningxin Su. 2026. PortBench: A correlation-aware, full-pipeline benchmark for LLM-driven portfolio management. *arXiv preprint arXiv:2605.27887* (2026).
- [61] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 19724–19731.

Appendices

Contents. **A** More Related Work (**A.1** memory and experience in LLM agents; **A.2** valuation and causal effects; **A.3** portfolio choice under estimation error; **A.4** forecast evaluation and safe deployment) · **B** Notations · **C** Derivations (**C.1** the half-fraction design; **C.2** the value model; **C.3** the trust test; **C.4** the anchored action) · **D** More Experiments (**D.1** dataset statistics; **D.2** baselines; **D.3** implementation details; **D.4** why LLM agents trail 1/N; **D.5** ablation details; **D.6** wealth, market phases and survival; **D.7** hyperparameter analysis; **D.8** strength of the anchor; **D.9** number of retrieved experiences by regime; **D.10** PlantedMem by regime; **D.11** ClassAlloc before and after the knowledge cutoff; **D.12** other LLMs and temperatures; **D.13** ClassAlloc with other LLMs; **D.14** look-ahead checks; **D.15** PortBench after the knowledge cutoff) · **E** Case Study · **F** Algorithms · **G** LLM Prompts (**G.1** InvestorBench; **G.2** ClassAlloc; **G.3** PortBench) · **H** Limitations.

A More Related Work

A.1 Memory and experience in LLM agents

In more detail than §2: MemGPT [40] manages a hierarchy of memory tiers for long interactions, MemoryBank [61] stores memories and lets them fade with a rule inspired by the forgetting curve, and Voyager [49] builds a library of executable skills that it reuses when they succeed; in finance, FinCon [55] updates the investment beliefs of a team of LLM agents by verbal reinforcement from realized profit and loss. These methods decide what to store, keep or reuse from the success of the episodes in which a memory took part, i.e., by outcome credit, and none of them tests whether its rule for using memory predicts future gains before acting on it.

A.2 Valuation and causal effects

The Shapley value [44] and the Banzhaf value [3] are the two classic ways to divide a joint payoff among players. Data valuation adapts them to training data [18, 23, 29, 50]: Beta Shapley reweights subsets by their size to reduce noise, and Data Banzhaf weights all subsets equally, which makes it the most robust semivalue when the utility is noisy. Our contribution is the main effect of a factor in a 2^{4-1} factorial experiment [6], the classic way to estimate several effects at once with few runs; Appendix C.1 shows that it equals the Banzhaf value when the three-way interaction aliased with it is negligible. Estimating what an experience does to a decision is also a problem of heterogeneous treatment effects [2, 28]. Uplift modelling [21] and counterfactual learning from logged decisions [5, 46] estimate such effects from observational data, and online controlled experiments [27] randomize users into groups. MemTrial instead runs the whole designed experiment on every date, so the market is held fixed within each comparison.

A.3 Portfolio choice under estimation error

Mean–variance optimization [36] is sensitive to errors in the estimated means and covariances, which has motivated shrinkage estimators [24, 31], portfolios that account for parameter and model uncertainty [17, 25], and combinations of an optimized portfolio with 1/N [48]. DeMiguel et al. [12] find that none of the optimized portfolios they evaluate consistently beats 1/N out of sample. Online portfolio selection [9, 22, 33] adapts the weights over time with regret guarantees. The anchored action of MemTrial combines two of these ideas: shrinkage towards a conservative portfolio, and an online blend learned with a regularizer.

A.4 Forecast evaluation and safe deployment

Prequential analysis [10] judges a model by its sequential out-of-sample predictions; tests of predictive accuracy [13] and proper scoring rules [19] formalize such judgments. The trust test of MemTrial applies this idea to the values of experiences, as a one-sided test of whether their out-of-sample correlations with the measured contributions are positive. Safe policy improvement [30, 47] deploys a new policy only when it improves on a baseline with high confidence; in MemTrial, the trust test plays the role of this confidence check, and the anchored action that of the baseline policy. Finally, shrinkage across units [14, 15] underlies our value model.

B Notations

C Derivations

C.1 The half-fraction design

Code the presence of the j -th retrieved experience in a subset S as $x_j = +1$ if $j \in S$ and $x_j = -1$ otherwise, and write the utility of a draft as a full factorial model,

$$U_t(S) = \mu_t + \sum_j \frac{a_j}{2} x_j + \sum_{j < l} \frac{a_{jl}}{2} x_j x_l + \sum_{j < l < m} \frac{a_{jlm}}{2} x_j x_l x_m + \frac{a_{1234}}{2} x_1 x_2 x_3 x_4 + \varepsilon_{t,S}, \quad (\text{C.1})$$

where μ_t collects everything the drafts of date t share, including the market move, a_j is the main effect of experience j (the mean utility over the 16 subsets with it minus the mean over those without it), the other a 's are interactions, and $\varepsilon_{t,S}$ is the sampling noise of the draft. Over all 16 subsets, the main effect equals the Banzhaf value of j , because both are the mean over subsets with j minus the mean over subsets without it. Since $k = 4$ is even, a subset has an even number of experiences exactly when $x_1 x_2 x_3 x_4 = +1$, so the design (2) is the half fraction with defining relation $I = x_1 x_2 x_3 x_4$. Multiplying by x_j shows that on the design x_j equals the product of the other three columns: each main effect is aliased with the three-way interaction of the other three experiences, and each two-way interaction with the complementary one ($x_1 x_2 = x_3 x_4$), which is resolution IV. The contribution (3) is the contrast of column x_j over the design, so under (C.1)

$$c_{t,j} = a_j + a_{(-j)} + \bar{\varepsilon}_{t,j}^+ - \bar{\varepsilon}_{t,j}^-, \quad (\text{C.2})$$

where $a_{(-j)}$ is the three-way interaction of the other three experiences and $\bar{\varepsilon}^+$, $\bar{\varepsilon}^-$ are the mean noise of the four drafts with and the four without experience j . Three consequences follow. First,

Table B.1: Notation.

Symbol	Meaning
t, T	decision date, number of dates
s_t	market state at date t
N	number of assets (plus cash)
$\theta = (M, m, \gamma)$	investor profile: maximum share of risky assets, minimum share of defensive assets, risk aversion
P_θ	projection of a portfolio onto the investor’s mandate
$u_t(x); R_t, V_t$	investor utility (certainty-equivalent return) of portfolio x , Eq. (1); net return and realized variance over the holding period
$\pi; x \sim \pi(\cdot s_t, S)$	LLM policy; a draft made with the experiences in S
k ;	number of retrieved experiences (4); experiences
$R_t = (h_{t,1}, \dots, h_{t,k})$	retrieved at date t
$\mathcal{D}$	designed subsets, Eq. (2)
$U_t(S)$	utility of the draft made with subset S , at realized prices
$c_{t,j}; y_{t,j}$	counterfactual contribution of the j -th retrieved experience, Eq. (3); centred contribution, Eq. (4)
$v_h; \hat{v}_h$	value of experience h ; its posterior mean, Eq. (7)
$z_h \in \mathbb{R}^d$	content feature of experience h ($d = 8$ principal components of its embedding)
$\beta; b_h$	map from content to value; experience-specific deviation
$\sigma^2; \sigma_\beta^2, \tau^2$	noise variance; prior variances, Eq. (5)
$n_h; m_h$	number of dates with a realized outcome on which h was retrieved; mean of its centred contributions
$\rho_t; n; \alpha$	forward score at date t , Eq. (8); number of forward scores; level of the trust test, Eq. (9)
S_t^*	trusted subset, Eq. (10)
$x^{\text{ref}}; \bar{x}_t$	reference portfolio; average draft
$q_t; \alpha_0; \lambda_0$	weight on the reference; its prior; regularization strength, Eq. (11)

μ_t , everything the drafts of date t share, cancels; the market move enters $c_{t,j}$ only through the weights that experience j changes. Second, $c_{t,j}$ estimates the main effect without bias when three-way interactions are negligible, and the four contrasts are orthogonal, so the contribution of one experience is not affected by two-way interactions. Third, the noise term has variance $\sigma_\epsilon^2/2$ for draft-level noise variance σ_ϵ^2 , against $\sigma_\epsilon^2/4$ for the full factorial with twice as many drafts. On ClassAlloc and InvestorBench, where all 16 subsets were drafted on every date and seed, the difference between $c_{t,j}$ and the full-factorial value, half the contrast between the two half fractions, has 1.26–1.34 times the mean square that sampling noise alone gives ($\sigma_\epsilon^2/4$, with σ_ϵ^2 estimated from the spread of a draft’s utility across seeds): the three-way interaction aliased with the main effect adds 13–17% to the noise variance of a contribution, and $c_{t,j}$ correlates with the full-factorial value at 0.76–0.86.

C.2 The value model

Stack the H experiences that have at least one measured contribution. Under Eq. (5), the values $v = Z\beta + b$ and the observed means

m are jointly Gaussian with

$$\begin{aligned} v &\sim \mathcal{N}(0, K_v), & K_v &= \sigma_\beta^2 Z Z^\top + \tau^2 I, \\ m &\sim \mathcal{N}(0, K), & K &= K_v + \sigma^2 \text{diag}(1/n_h), \end{aligned} \quad (\text{C.3})$$

which gives the marginal likelihood (6). Gaussian conditioning gives $\mathbb{E}[v | m] = K_v K^{-1} m$ and $\mathbb{E}[\beta | m] = \sigma_\beta^2 Z^\top K^{-1} m$, i.e., Eq. (7). For a new experience h' without measurements, $\text{cov}(v_{h'}, m) = \sigma_\beta^2 z_{h'}^\top Z^\top$, so $\mathbb{E}[v_{h'} | m] = z_{h'}^\top \hat{\beta}$: its value is predicted from its content alone. Without content ($\sigma_\beta^2 = 0$), K is diagonal and

$$\hat{v}_h = \frac{\tau^2}{\tau^2 + \sigma^2/n_h} m_h, \quad (\text{C.4})$$

the James–Stein-type shrinkage of Fay III and Herriot [15]: an experience measured on few dates is shrunk towards zero, which after the centring (4) is the value of an average experience. The noise variance σ^2 is the pooled within-experience variance of the centred contributions, and (σ_β^2, τ^2) maximize (6) over a grid.

C.3 The trust test

Suppose that the learned values carry no information about future contributions, so that every forward score has mean zero. If the scores were independent and approximately normal, the statistic in Eq. (9) would follow a Student t distribution with $n - 1$ degrees of freedom, and a single test at a fixed n would pass with probability at most α . MemTrial uses the test as a decision rule: it applies it on every date with $n \geq 10$, to the scores of the learner whose mean score is higher, and scores of nearby dates are dependent when the same experiences recur. The level α is therefore the level of one test, not the rate at which the rule passes over a run. That rate is measured where every pass is a false one: in the two PlantedMem regimes in which experiences carry no signal, the rule passed on 2.3% and 3.0% of decisions (Table D.13). The minimum of 10 scores and the level $\alpha = 0.05$ were fixed on the development seeds, and Table D.9 varies both.

C.4 The anchored action

Both the reference and the average draft satisfy the mandate, which is a convex set, so every blend $x_t(q)$ does too. Because q is a single number, Eq. (11) is solved by evaluating the objective on a grid of 201 values, which finds its maximizer up to the grid spacing without any assumption on its shape; the utility of every blend is computed exactly on the realized price path, including trading costs. Before any outcome is realized, $q_t = \alpha_0$. As n grows, $\lambda_n = \lambda_0 \hat{\sigma}_n / \sqrt{n}$ tends to zero, so the weight of the prior fades and q_t approaches the blend with the highest mean utility over the past dates, as in follow-the-regularized-leader [43]. Scaling by $\hat{\sigma}_n$ makes λ_0 independent of the units of utility, so the same $\lambda_0 = 4$ is used on all benchmarks.

D More Experiments

As in the main text, *Improv.* is the gain of MemTrial over the best experience-learning agent of each setting, in percent of its utility.

D.1 Dataset statistics

Table D.1 gives the size of each benchmark and the cost of our runs. The smallest sample is the PortBench test period (20 months), which

is why we test significance across test dates; none of the PortBench differences in Table 3 is significant. The ten further InvestorBench runs of §4.4 make 8,940 executions each (the 16 subsets, ExpeL and the three sequential agents, over 149 days and three seeds) and cost USD 1.21–16.90 each; the seven further ClassAlloc runs make 4,318 executions each and cost USD 0.38–6.01 each, and the two extra memory-free draws per date and seed of Self-consistency cost USD 6.76 in all. Replaying the sequential agents of all ten ClassAlloc runs with the feedback timing of Appendix D.3 cost USD 1.26.

D.2 Baselines

Several methods are evaluated on the same logged drafts, so that they differ only in how they use memory. $1/N$ [12] holds equal weights in all assets and cash, projected to the mandate. *Minimum variance* [36] is the long-only minimum-variance portfolio estimated from the past 60 daily returns, projected to the mandate. *Memory-free* (zero-shot) [7] deploys the draft made without experiences. *Self-consistency* [51] averages memory-free drafts of the same decision: on the real benchmarks, three per seed (the seed’s own draft and two further draws, made after the main runs with the same prompt, LLM and temperature); in PlantedMem, eight. *Similarity retrieval* [32] deploys the draft made with the two most similar or with all four retrieved experiences. *FinMem* [54] ranks memories by similarity, recency and an importance score that is reinforced when the decision that used them earns a positive return. *MemRL* [56] learns a value for each memory from the rewards of the decisions that used it and selects among the most similar candidates, with ϵ -greedy exploration. *Reflexion* [45] writes a verbal reflection after each outcome and conditions later decisions on the most recent reflections. *ExpeL* [59] distils insights from successful and failed decisions; on InvestorBench the LLM writes them, and on PortBench and in PlantedMem we use its insight votes ($\dagger$): an experience gains a vote when the decision it took part in beats $1/N$ and loses one otherwise, and the two experiences with the most votes are used. *Uplift credit* [21] regresses the utility of the deployed decision on indicators of which experiences were used, with the utility of $1/N$ as a market control and a small ridge penalty, and uses the two experiences with the highest coefficients, exploring a random pair on 10% of dates. *Counterfactual selection* [50] measures the Banzhaf contribution of each retrieved experience on every date with a realized outcome and deploys the draft with the two experiences of highest mean past contribution, without value estimation, trust test or anchor. *Draft averaging* [4] deploys the average of MemTrial’s eight designed drafts. *Hedge* [16] mixes $1/N$ and the portfolios of the memory configurations available on each benchmark (memory-free, top-4 retrieval, FinMem and MemRL, plus the benchmark’s native memory on PortBench) with exponential weights, whose learning rate is scaled as in Eq. (11). On PortBench, FinMem and MemRL select experiences from the same pool with their own memory modules; where a baseline’s own module cannot run ($\dagger$ in Table 3), we re-implement its memory-scoring rule over the same experience pool. The baselines learn as follows. On PortBench, the FinMem and MemRL modules receive no feedback after July 2019, before the first decision date, and score the pool by the outcomes recorded up to then; ExpeL $\dagger$ is the vote rule above (ExpeL-style voting), and Reflexion writes a reflection

once the outcome of each decision is known. On InvestorBench and ClassAlloc, FinMem $\dagger$ and MemRL $\dagger$ update their importance and values, and Reflexion its reflections, once the outcome of each decision is known; FinMem $\dagger$ and MemRL $\dagger$ learn from the return of their own draft over the holding period, before the mandate and fees. ExpeL’s insights are written once from the warm-up lessons. In PlantedMem, every adapted agent updates after every period. Uplift credit, Counterfactual selection, Hedge and MemTrial learn from every realized outcome on all benchmarks.

D.3 Implementation details

- *Agents*. All LLM agents use gpt-4.1-mini (2025-04-14) [38] at temperature 0.7. On PortBench, a draft is one run of the benchmark’s staged agent (market interpretation, signal generation and weight optimization), which makes three LLM calls with JSON output; the retrieved experiences enter the prompt as a memory block of at most 4,096 tokens. PortBench-Raw uses the benchmark’s native prompt, which shows a price table of eight assets, and PortBench-Full extends the table to all eligible assets. On InvestorBench, a draft is one chat completion that returns a JSON object with the weights of the four stocks and cash; the prompt gives, for each stock, the close, the 1-, 5- and 20-day returns, the 20-day volatility and up to three news headlines of the day, and an invalid reply is retried once. The runs with other LLMs (§4.4) also use gpt-4.1-nano (2025-04-14) [38], gpt-5-mini (2025-08-07; minimal reasoning effort, temperature fixed by the API [39], and, through OpenRouter, Llama-3.3-70B-Instruct [20], Qwen3-235B-A22B-Instruct-2507 [53], DeepSeek-V3.1 (non-thinking mode) [11], Gemini 2.5 Flash (thinking budget 0) [8] and Claude Haiku 4.5 [1], all with unchanged prompts; on InvestorBench the lessons and their embeddings are those of the main run, and on ClassAlloc each LLM writes its own lessons; the ExpeL insights and Reflexion notes are always written by the LLM under test.
- *ClassAlloc*. Each class index averages, every day, the daily returns of PortBench’s assets in that class that have prices on both days (44 equities, 15 bond, 16 commodity, 10 real-estate and 4 cash assets); an asset listed after January 2015 enters its index only after 20 sessions of prices, because the first quotes of new listings can be wrong (one coin gains 10,200% on its third day). Decisions are made on the first session of each month from July 2015 (on 6 July, the first session with six months of history) to March 2026, the last month with 20 sessions of outcome in the data. Each decision is scored over the 20 sessions after it, and the holdings are carried to the next decision date. Every learner uses an outcome only after its 20 sessions have passed: in 9 of the 70 test months the next decision comes after 19 sessions, and that outcome is used from the decision after. The sequential agents of our runs (FinMem, MemRL and Reflexion) were replayed under this timing from the logged runs: a logged decision whose prompt is unchanged is kept, and any other decision, with the reflection written after it, is drawn anew (210 of 639 in the main run). The mandate treats equities and real estate as risky and bonds and cash

Table D.1: Benchmark statistics and cost.

	PortBench (Full / Raw)	InvestorBench	ClassAlloc	PlantedMem
Decision dates	59 monthly, Aug 2019 – Dec 2024 (a month whose 20-day window would overlap the previous one is skipped); 56 usable per configuration	63 warm-up and 149 test trading days, Jul 2020 – May 2021	129 monthly, Jul 2015 – Mar 2026: 58 warm-up and 71 test	132 monthly, Apr 2015 – Mar 2026, per episode; the first 12 are a warm-up
Development / test Assets	36 / 20 months 99 in 6 classes, including cash	none (fully held out) / 149 days HON, JNJ, MSFT, UVV and cash	none (fully held out) / 71 months equities, bonds, commodities, real estate and cash: equal-weighted indices of 89 assets	seeds 0–14 / 15–114 equities, bonds, cash, commodities
Experiences	338 records of earlier decisions (Apr 2015 – Jul 2019); 29 retrieved	63 lessons written in the warm-up	58 lessons written in the warm-up	12 or 48 planted per episode
Drafts per date and seed	19: the 16 subsets of the 4 retrieved experiences, FinMem, MemRL and the benchmark’s native memory; Reflexion runs separately	20: the 16 subsets, ExpeL, FinMem, MemRL and Reflexion	20: the 16 subsets, ExpeL, FinMem, MemRL and Reflexion	resampled from 2,828 logged LLM allocations
LLM executions	6,612 (6,518 valid) and 354 Reflexion decisions	9,003	4,318 (58 in the warm-up), and 210 redrawn (Appendix D.3)	none
API cost	about USD 43.6	USD 5.09	USD 1.61, and 0.12 to redraw	none
Retrieval ($k = 4$)	Qwen3 embeddings	text-embedding-3-small	text-embedding-3-small	regime match with noise
Trading costs (15 bp)	on the distance from $1/N$	on the amount traded	on the amount traded	on the distance from $1/N$
Seeds	3	3	3	100 (and 15 for development)
Evaluated test decisions per method	360 (20 months $\times$ 3 investors $\times$ 3 seeds $\times$ 2 configurations)	1,341 (149 days $\times$ 3 investors $\times$ 3 seeds)	639 (71 months $\times$ 3 investors $\times$ 3 seeds)	252,000 (100 seeds $\times$ 120 months $\times$ 7 regimes $\times$ 3 investors)

as defensive, as PortBench does. Each lesson is written after the agent sees the returns of the five classes over the 20 holding sessions, its own portfolio return and that of the equal-weight portfolio, and is retrieved by text-embedding-3-small [37] as on InvestorBench. FinMem’s recency decays over 4.5 months (about 90 trading days); all other settings are as on InvestorBench.

- *PlantedMem*. An episode runs over 132 monthly dates (Table D.1) with four asset classes, the equal-weighted indices of PortBench’s equities, bonds, cash and commodities, and 20-session outcomes; every decision starts from the equal-weight portfolio, and its outcome is known before the next one. Of the $M = 12$ (or 48) planted experiences, $M/4$ are useful, $M/2$ neutral and $M/4$ misleading, in random order. Each experience has a home regime, drawn independently of its type; on date t it receives the retrieval score $\mathbb{P}[\text{home} = g_t] + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5^2)$, where $g_t = 1$ if the equities rose on average over the previous 60 sessions, and the four highest scores are retrieved. On each date, a retrieved experience points to an asset class $c_{t,h}$: a useful (misleading) experience points with probability 0.7 to the class with the highest (lowest) return over the coming 20 sessions, and otherwise, like a neutral one, to a class drawn uniformly. A draft made with the experiences S is $x^S = (1 - \beta)b + \beta \frac{1}{|S|} \sum_{h \in S} e_{c_{t,h}}$, where b is an allocation drawn for each draft from 2,828 logged gpt-4.1-mini allocations of PortBench (aggregated to the four classes) and e_c is the indicator of class c ; the memory-free draft is b . The regimes set $\beta \in \{0, 0.1, 0.25, 0.5\}$, and *noise only* uses $\beta = 0.5$ with every experience neutral. The content feature is $z_h = \kappa q_h e_1 + \xi_h$, with type $q_h \in \{+1, 0, -1\}$, $\xi_h \sim \mathcal{N}(0, I_8)$, $\kappa = 1.5$ in the regime with informative content and 0 otherwise. Usefulness is thus defined by the direction in which an experience moves the draft, and its gain in utility depends on the date and the investor. The adapted baselines act

on the same retrieved experiences: FinMem† adds an importance that rises when the decision beats $1/N$, MemRL† learns values from the utilities of the decisions that used the experiences, ExpeL† votes, and Reflexion† tilts the draft towards the classes that did best in the last three periods.

- *Retrieval*. On PortBench, the query is a short JSON text with the task description, the date, the public market features, the number of assets and the risk profile of the benchmark task. Each experience is the JSON record of an earlier decision: its allocation by asset class, the public market features at the time and the realized 20-day outcome. Both are embedded with Qwen3-Embedding-0.6B [58] (1,024 dimensions, run locally), and the four records with the highest cosine similarity are retrieved. All 338 records became available between April 2015 and July 2019, before the first decision date. On InvestorBench, the query is the day’s market context described above, each experience is a lesson of at most 700 characters that the agent wrote after seeing the outcome of a warm-up decision, and both are embedded with text-embedding-3-small. Ties are broken by experience identifier.
- *Mandate*. On PortBench, P_θ is the Euclidean projection onto the long-only portfolios whose risky share (equities, real estate and cryptocurrency) is at most M and whose defensive share (bonds and cash) is at least m , computed by alternating projections. On InvestorBench, negative weights are set to zero and, if the stock share exceeds $\min(M, 1 - m)$, the stock weights are scaled down to this cap and the rest goes to cash.
- *Trading costs and utility*. Costs are 15 bp of the L1 distance moved. Each PortBench decision is an episode that starts from the equal-weight book, as in the benchmark, so the distance is measured from it; PlantedMem does the same. On InvestorBench, holdings drift with prices overnight and the distance is measured from the drifted holdings, so every

method pays for what it actually trades. The variance term of Eq. (1) is the variance of daily portfolio returns times the number of days held (20); on InvestorBench, with one day held, it is the squared daily return.

- *MemTrial*. Content features z_h are the first $d = 8$ principal components of the retrieval embeddings of all experiences, standardized; in PlantedMem, the first coordinate carries κ times the experience’s quality plus standard normal noise and the others are noise. (σ_p^2, τ^2) maximize the marginal likelihood (6) over a small grid; Eq. (11) is solved over 201 values of q . All learning uses only outcomes realized before the decision date.
- *Seeds*. A seed means different things on the two kinds of benchmark. On the real benchmarks, a seed is an independent deployment in which every LLM draft is sampled anew. There is only one market history, so additional seeds measure only the randomness of the LLM, and each one costs a full set of paid calls (about USD 14.5 per seed on PortBench, USD 43.6 for all methods and seeds); we use three seeds and test significance across test dates, where the randomness of the market lies. In PlantedMem, a seed is a newly generated world (which experiences are useful, the retrieval noise and the resampled drafts), which costs no LLM call, so we use 100 test seeds to make the estimates precise. A standard deviation shown as 0.00 is below 0.005; this happens when MemTrial stays anchored at the reference, which does not depend on the seed.
- *LLM cost and learning overhead*. MemTrial makes eight drafts per decision. The drafts are independent and run in parallel, so the latency of a decision is that of a single draft. With gpt-4.1-mini, one draft costs about USD 0.006 on PortBench and USD 0.0006 on InvestorBench, whose prompts are shorter, so a MemTrial decision costs about USD 0.05 and USD 0.005, respectively; the complete InvestorBench run, covering all 15 methods with three seeds, cost USD 5.09, and the ClassAlloc run USD 1.61 (Appendix D.1). At equal LLM spend, the relevant comparison is Draft averaging, which uses the same eight drafts and is 0.36 and 0.81 pp per month worse than MemTrial on PortBench, 1.70 bp per day worse on InvestorBench and 0.12 pp worse in PlantedMem (Table 3). Fitting both value models, scoring the trust test and solving Eq. (11) takes 3.6 ms per decision on average with 12 experiences and 6.9 ms with 48 experiences (95th percentiles 6.2 and 10.4 ms, one CPU core), and recording a realized outcome takes 0.1 ms, which is negligible next to a single LLM call.
- *Statistics*. Mean $\pm$ sd over seeds; one-sided paired t -tests, paired by test date on the real benchmarks (after averaging over seeds and investors) and by seed in PlantedMem. Every number in the paper is recomputed offline from the logged drafts; the source file of MemTrial is unchanged since it was frozen (its sha256 is recorded with the code).

D.4 Why do LLM agents trail 1/N?

Table D.2 splits the test-period utility into gross return, trading cost and risk penalty (utility = gross return – cost – risk penalty, up to

rounding; mean $\pm$ sd over seeds, each averaged over the investors, in % per month on PortBench and bp per day on InvestorBench). L1 is the mean distance from the equal-weight book, on which PortBench charges costs. On PortBench-Full, the memory-free draft, the top-4 retrieval draft and the average of the eight designed drafts trail 1/N by 0.33–0.42 pp per month, and the gap has three parts of similar size: 0.06–0.16 pp less gross return, 0.13–0.15 pp more trading cost and a 0.11–0.14 pp larger risk penalty. Each month, the LLM moves about half of the capital away from the equal-weight book (an L1 distance of 1.0–1.2, against 0.17 for 1/N projected to the mandate), which costs money and adds risk without earning more. On PortBench-Raw, whose prompt shows only eight assets, the drafts are also more defensive than 1/N: they earn 0.72–0.85 pp less gross return at a slightly smaller risk penalty. On InvestorBench, the LLM-based methods earn about the same gross return as 1/N (10.7–11.7 against 11.4 bp per day), but they change their portfolios substantially every day and pay 1.0–2.5 bp per day in trading costs, against 0.1 bp for 1/N and 0.2 bp for MemTrial. With trading costs set to zero and all methods recomputed, MemTrial included, every LLM draft still trails 1/N on PortBench, by 0.17–0.27 pp per month on Full and 0.68–0.81 pp on Raw, while MemTrial stays within 0.04 pp of it. In short, the drafts do not pick better assets than equal weighting, and their frequent reallocation costs money. Experience learning would have to close this gap to beat 1/N. Outcome credit does not close it, and MemTrial’s trust test finds that its own learned values do not close it either, so MemTrial stays near the reference.

D.5 Ablation details

This appendix completes §4.3 (Figure 4). Table D.3 gives the trust rate of every variant, the share of test decisions on which it acted on its learned values (mean $\pm$ sd over seeds: three seeds on the real benchmarks, each pooling the configurations and investors; 100 test seeds in PlantedMem and 50 for the k designs). PlantedMem (core) averages the five core regimes, and PlantedMem (informative) is the regime with 48 experiences and informative content; – marks settings that were not evaluated. A large sd on the real benchmarks means that the test passed mostly in one seed. Table D.4 gives the utilities behind Figure 4. The variant whose test never passes always takes the anchored action: it is identical to MemTrial on PortBench-Full, where the test of MemTrial never passed either, and in PlantedMem it stays at the reference in every regime (Table D.14). On ClassAlloc it earns 0.629% per month against 0.613% for MemTrial: acting on the trusted values cost 0.016 pp per month there ($p = 0.28$; Appendix E). Fixing the blend of the anchored action at its prior ($q = \alpha_0 = 0.9$, no online learning) and always keeping the reference ($q = 1$) bracket the learned blend: the first differs from MemTrial by –3.1% to +0.9%, and the second is the extreme $\alpha_0 = 1$ of the prior (Appendix D.8).

Why do some values coincide? On PortBench and InvestorBench, the three variants that change how values are learned or tested give the same numbers as MemTrial in all but two cells. This is expected and not an error. Such a change can affect a decision only when the values are acted upon, and on these two benchmarks the test (or the F-test) passes on at most 1.1% of decisions (Table D.3), so on all other decisions the four methods take the same anchored action. The two exceptions come from the few decisions on which

Table D.2: Decomposition of the test-period utility. The LLM drafts trail $1/N$ in all three parts on PortBench-Full, mostly in gross return on PortBench-Raw and mostly in trading costs on InvestorBench.

Benchmark	Method	Utility	Gross return	Trading cost	Risk penalty	L1
PortBench-Full	$1/N$	2.02 ± 0.00	2.25 ± 0.00	0.03 ± 0.00	0.21 ± 0.00	0.17 ± 0.00
	Memory-free	1.60 ± 0.05	2.12 ± 0.05	0.17 ± 0.02	0.34 ± 0.03	1.13 ± 0.11
	Sim. retrieval (top-4)	1.70 ± 0.06	2.19 ± 0.04	0.18 ± 0.02	0.31 ± 0.02	1.17 ± 0.14
	Draft averaging	1.62 ± 0.05	2.09 ± 0.06	0.16 ± 0.00	0.32 ± 0.02	1.02 ± 0.01
PortBench-Raw	$1/N$	2.02 ± 0.00	2.25 ± 0.00	0.03 ± 0.00	0.21 ± 0.00	0.17 ± 0.00
	Memory-free	1.07 ± 0.02	1.43 ± 0.03	0.18 ± 0.01	0.17 ± 0.00	1.19 ± 0.08
	Sim. retrieval (top-4)	1.05 ± 0.02	1.40 ± 0.02	0.19 ± 0.01	0.17 ± 0.00	1.22 ± 0.05
	Draft averaging	1.19 ± 0.02	1.53 ± 0.02	0.17 ± 0.00	0.17 ± 0.00	1.13 ± 0.03
InvestorBench	$1/N$	10.45 ± 0.00	11.44 ± 0.00	0.11 ± 0.00	0.88 ± 0.00	–
	Memory-free	8.43 ± 0.29	11.53 ± 0.31	2.16 ± 0.07	0.94 ± 0.01	–
	Sim. retrieval (top-4)	8.12 ± 0.61	10.74 ± 0.58	1.72 ± 0.05	0.90 ± 0.01	–
	Draft averaging	8.69 ± 0.09	11.23 ± 0.11	1.62 ± 0.03	0.91 ± 0.00	–
	ExpeL	8.93 ± 0.32	10.78 ± 0.34	0.96 ± 0.02	0.89 ± 0.00	–
	MemTrial (ours)	10.40 ± 0.04	11.44 ± 0.06	0.16 ± 0.01	0.88 ± 0.00	–

Table D.3: Trust rate (%) of every variant. With the trust test, MemTrial acts on its values on at most 0.3% of PortBench and InvestorBench decisions; without it, on 25–35%.

Variant	PortBench	InvestorBench	ClassAlloc	PlantedMem (core)	PlantedMem (informative)
<i>Ablation (Figure 4)</i>					
MemTrial (ours)	0.3 ± 0.5	0.3 ± 0.5	14.6 ± 21.7	33.7 ± 9.2	44.9 ± 22.5
per-experience only (no content)	0.0 ± 0.0	0.3 ± 0.5	14.6 ± 21.7	31.8 ± 8.9	22.0 ± 20.7
content-aware only (no learner selection)	1.1 ± 1.3	0.3 ± 0.5	14.6 ± 21.7	32.6 ± 9.0	44.5 ± 23.1
in-sample F-test	0.0 ± 0.0	0.0 ± 0.0	2.3 ± 4.1	29.3 ± 8.4	15.1 ± 17.8
no trust test	24.7 ± 21.4	34.5 ± 18.8	53.8 ± 24.7	54.3 ± 12.6	64.9 ± 18.6
test never passes (always anchored action)	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
draft average ($q = 0$)	0.3 ± 0.5	0.0 ± 0.0	14.9 ± 20.5	33.7 ± 9.2	44.9 ± 22.5
always the reference ($q = 1$)	0.3 ± 0.5	0.3 ± 0.5	14.4 ± 21.8	33.7 ± 9.2	44.9 ± 22.5
uniform prior ($\alpha_0 = 0.5$)	0.3 ± 0.5	0.0 ± 0.0	14.7 ± 20.6	33.7 ± 9.2	44.9 ± 22.5
no online learning (q fixed at α_0)	0.3 ± 0.5	0.3 ± 0.5	14.4 ± 21.8	33.7 ± 9.2	44.9 ± 22.5
<i>Trust test (Table D.9); default $\alpha = 0.05$, at least 10 forward scores</i>					
$\alpha = 0.01$	0.0 ± 0.0	0.0 ± 0.0	–	28.2 ± 8.1	32.9 ± 22.3
$\alpha = 0.02$	0.0 ± 0.0	0.0 ± 0.0	–	30.3 ± 8.5	37.8 ± 22.7
$\alpha = 0.1$	5.3 ± 5.0	0.5 ± 0.9	–	36.9 ± 10.0	50.6 ± 21.4
$\alpha = 0.2$	12.2 ± 13.0	1.6 ± 2.3	–	40.7 ± 11.0	56.3 ± 20.4
at least 5 forward scores	0.3 ± 0.5	0.3 ± 0.5	–	33.9 ± 9.2	44.6 ± 22.2
at least 20 forward scores	0.0 ± 0.0	0.0 ± 0.0	–	32.0 ± 8.5	43.4 ± 21.6
<i>Number of retrieved experiences and design (Table D.10); PlantedMem on seeds 15–64</i>					
$k = 2$, full factorial (4 drafts)	0.0 ± 0.0	2.0 ± 2.0	–	8.5 ± 7.7	7.2 ± 12.3
$k = 3$, half fraction (4 drafts)	0.8 ± 1.4	4.5 ± 5.2	–	18.5 ± 9.4	9.5 ± 13.7
$k = 3$, full factorial (8 drafts)	0.0 ± 0.0	10.8 ± 10.7	–	29.4 ± 11.0	43.4 ± 25.6
$k = 4$, half fraction (8 drafts; seeds 15–64)	0.3 ± 0.5	0.3 ± 0.5	–	35.4 ± 11.0	49.6 ± 24.3
$k = 4$, full factorial (16 drafts)	1.1 ± 1.9	8.5 ± 7.4	–	42.0 ± 8.1	69.9 ± 20.1
$k = 5$, half fraction (16 drafts)	–	–	–	43.4 ± 10.3	74.7 ± 17.0
$k = 6$, half fraction (32 drafts)	–	–	–	49.1 ± 9.8	86.4 ± 10.6

the tests disagree: the content-aware learner alone passes its test by chance on a few PortBench-Raw decisions (1.94 ± 0.12 ; observation 3), and the F-test never passes on InvestorBench, whereas MemTrial’s test passes on 0.3% of decisions (10.37 against 10.40). On ClassAlloc, the value model estimates the content term at zero on every date, so the content-aware learner coincides with the per-experience learner, and the two value-estimation variants give the same numbers as MemTrial although the test passes on 15% of decisions. Where no test passes, the anchored action stays close to the reference, which does not depend on the seed, so the sd is below 0.005; a test that passes on a few decisions of one seed raises it (0.04

for MemTrial on InvestorBench). The same steps make a difference in PlantedMem, where these variants trust their values on 15–45% of decisions and the columns differ. The anchored-action variants change what the agent does when its values are not trusted, so they matter most on PortBench and InvestorBench, where the values are almost never trusted.

We make the following observations. (1) *The trust test matters most on real markets.* Without it, MemTrial acts on its learned values whenever they favour a designed subset, on 25% of PortBench and 34% of InvestorBench test decisions, and loses 0.08 pp per month on PortBench-Full, 0.24 pp on PortBench-Raw and 0.49 bp per day

Table D.4: Utilities behind Figure 4, with the variant that always keeps the reference ($q = 1$; Appendix D.8) (% per month, bp per day on InvestorBench; mean $\pm$ sd over seeds). Without signal, the variants differ little in PlantedMem; with signal, never acting on the learned values stays at the reference.

Step changed	Variant	PortBench-Full	PortBench-Raw	InvestorBench	ClassAlloc	PlantedMem			
						no signal	signal	informative content	core
–	MemTrial (ours)	1.98 $\pm$ 0.00	<u>2.01 $\pm$ 0.00</u>	<u>10.40 $\pm$ 0.04</u>	0.613 $\pm$ 0.017	0.784 $\pm$ 0.031	0.932 $\pm$ 0.070	0.862 $\pm$ 0.068	<u>0.873 $\pm$ 0.045</u>
Value estimation	per-experience learner only	1.98 $\pm$ 0.00	2.01 $\pm$ 0.00	10.40 $\pm$ 0.04	0.613 $\pm$ 0.017	0.786 $\pm$ 0.030	0.929 $\pm$ 0.069	0.817 $\pm$ 0.048	0.872 $\pm$ 0.045
	content-aware learner only	1.98 $\pm$ 0.00	1.94 $\pm$ 0.12	<u>10.40 $\pm$ 0.04</u>	0.613 $\pm$ 0.017	0.785 $\pm$ 0.031	0.929 $\pm$ 0.072	<u>0.866 $\pm$ 0.072</u>	0.871 $\pm$ 0.046
Trust test	in-sample F-test	1.98 $\pm$ 0.00	<u>2.01 $\pm$ 0.00</u>	10.37 $\pm$ 0.00	0.628 $\pm$ 0.001	<u>0.786 $\pm$ 0.029</u>	0.924 $\pm$ 0.064	0.814 $\pm$ 0.045	0.869 $\pm$ 0.043
	none (always act on values)	1.89 $\pm$ 0.11	1.77 $\pm$ 0.41	9.91 $\pm$ 0.45	0.612 $\pm$ 0.005	0.767 $\pm$ 0.038	0.940 $\pm$ 0.080	0.882 $\pm$ 0.068	0.871 $\pm$ 0.052
	never passes (always anchored action)	1.98 $\pm$ 0.00	<u>2.01 $\pm$ 0.00</u>	10.37 $\pm$ 0.00	0.629 $\pm$ 0.001	0.787 $\pm$ 0.029	0.788 $\pm$ 0.029	0.789 $\pm$ 0.029	0.788 $\pm$ 0.028
Anchored action	draft average ($q = 0$)	1.62 $\pm$ 0.05	1.19 $\pm$ 0.02	8.69 $\pm$ 0.09	0.648 $\pm$ 0.007	0.747 $\pm$ 0.032	0.905 $\pm$ 0.080	0.837 $\pm$ 0.073	0.842 $\pm$ 0.049
	uniform prior ($\alpha_0 = 0.5$)	1.83 $\pm$ 0.02	1.92 $\pm$ 0.01	9.71 $\pm$ 0.02	<u>0.632 $\pm$ 0.008</u>	0.777 $\pm$ 0.024	0.924 $\pm$ 0.072	0.855 $\pm$ 0.066	0.865 $\pm$ 0.044
	no online learning (q fixed at α_0)	<u>1.99 $\pm$ 0.01</u>	1.94 $\pm$ 0.00	10.39 $\pm$ 0.04	0.613 $\pm$ 0.018	0.784 $\pm$ 0.031	0.931 $\pm$ 0.070	0.862 $\pm$ 0.068	0.872 $\pm$ 0.045
	always the reference ($q = 1$)	2.02 $\pm$ 0.00	2.02 $\pm$ 0.00	10.48 $\pm$ 0.05	0.605 $\pm$ 0.018	0.785 $\pm$ 0.034	<u>0.933 $\pm$ 0.070</u>	0.864 $\pm$ 0.068	0.874 $\pm$ 0.045

on InvestorBench, with a much larger spread across seeds. In PlantedMem, removing the test leaves the core average unchanged and gains 0.02 pp in the informative regime, where acting early pays off; §4.5 shows that the test trades this small delay for protection when experiences carry no signal. On ClassAlloc, where the test passes on 15% of decisions, acting on the values does not pay either way: without the test, MemTrial loses 0.001 pp, and the F-test, which passes on 2% of decisions, gains 0.016 pp. (2) *The test must look at future dates, not past ones.* An in-sample F-test of whether experiences differed in the past leaves PortBench and the core regimes nearly unchanged and never passes on InvestorBench (−0.03 bp per day), but it loses 0.05 pp in the informative regime: with 48 experiences, each retrieved only occasionally, the F-test, which needs repeated measurements of the same experience, passes on only 15% of decisions there, against 45% for the trust test. (3) *Content-aware value estimation helps when content is informative and costs nothing otherwise.* Without it, MemTrial trusts its values on only 22% of decisions in the informative regime and loses 0.05 pp. Using the content-aware learner alone, without learner selection, is as good there but costs 0.07 pp on PortBench-Raw, with a larger spread across seeds, because its chance forward scores pass the test more often (1.1% against 0.3% of PortBench decisions); selecting the learner by its forward score keeps the benefit and avoids the cost. (4) *When the values are not trusted, the agent must stay anchored to the reference.* Replacing the anchored blend by the draft average costs 0.36 pp (Full), 0.82 pp (Raw), 1.70 bp per day (InvestorBench) and 0.03 pp (PlantedMem, core), and learning the blend from a uniform prior ($\alpha_0 = 0.5$) costs 0.14 and 0.09 pp on PortBench and 0.68 bp per day on InvestorBench. Always keeping the reference ($q = 1$) is as good as MemTrial in PlantedMem and slightly better on the public benchmarks (by up to 0.04 pp per month and 0.08 bp per day), because there the average draft never beats the reference; online learning matters when the reference is weak (Appendix D.8). ClassAlloc is such a case: there the drafts beat $1/N$, so the draft average would gain 0.035 pp per month and the uniform prior 0.019 pp, whereas always keeping the reference loses 0.008 pp.

Baselines with the same anchor. Table D.5 gives every LLM-based baseline the anchored action of MemTrial: on every date,

it deploys $q_t x_t^{\text{ref}} + (1 - q_t) x_t$, where x_t is the portfolio that the baseline would deploy and q_t follows Eq. (11) on the baseline’s own realized outcomes ($\alpha_0 = 0.9$, $\lambda_0 = 4$); holdings, fees and the timing of outcomes are those of each benchmark. For Draft averaging, this is exactly the variant whose test never passes. On the real benchmarks, the anchor helps every baseline: all of them come within 1% of MemTrial on PortBench and InvestorBench, and on ClassAlloc, where the trusted decisions of MemTrial lost, all of them are ahead of it (0.621–0.635% against 0.613% per month). No difference between MemTrial and the best anchored baseline on a real benchmark is significant ($p \geq 0.13$, two-sided paired t -test over test dates), and Hedge with a prior weight of 0.9 on the reference does as well. In PlantedMem, the anchor keeps the baselines close to the reference: with signal, they earn 0.787–0.838% per month against 0.788% for the reference, whereas MemTrial, which acts fully on its values once they have predicted unseen dates, earns 0.932% per month, 11% more than the best of them (Counterfactual selection; $p < 0.001$ over seeds), and 7.5% more with informative content. Without signal, the anchored baselines stay at the reference and MemTrial is 0.6% below them, the cost of its false passes.

D.6 Wealth over time, market phases and survival

Wealth. For every method, seed and setting, the wealth path starts at 1 and follows the net returns after the 15 bp trading fee, before the risk penalty of the utility; we divide it by the wealth path of $1/N$ in the same setting. On InvestorBench and ClassAlloc, which carry the holdings from one decision to the next, it is the value of the account from each decision date to the next (on ClassAlloc, a decision is scored over the 20 sessions after it and held until the next decision date). On PortBench, whose decisions each start from the equal-weight book, it compounds the net returns of the 20-day decision periods. The net returns are those of the portfolios deployed in the published runs: re-running the evaluation reproduces every published utility exactly, so the wealth paths need no new LLM calls. **Market phases.** For each benchmark we form the equal-weighted index of its equities (PortBench: the equities in the date’s universe; InvestorBench: the four stocks; ClassAlloc: the equities class) and

Table D.5: Every LLM-based baseline with the anchored action of MemTrial (Eq. (11); % per month, bp per day on InvestorBench; mean $\pm$ sd over seeds). With the same anchor, the baselines do as well as MemTrial on the real benchmarks; with signal in PlantedMem, MemTrial earns 11% more than the best of them.

Method	PortBench-Full	PortBench-Raw	InvestorBench	ClassAlloc	PlantedMem		
					no signal	signal	informative content
1/N	2.021 $\pm$ 0.000	2.021 $\pm$ 0.000	10.45 $\pm$ 0.00	0.623 $\pm$ 0.000	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000
MemTrial (ours)	1.976 $\pm$ 0.005	2.007 $\pm$ 0.002	10.40 $\pm$ 0.04	0.613 $\pm$ 0.017	0.784 $\pm$ 0.031	0.932 $\pm$ 0.070	0.862 $\pm$ 0.068
Hedge, prior 0.9 on the reference	<u>1.992 $\pm$ 0.009</u>	2.003 $\pm$ 0.003	<u>10.41 $\pm$ 0.01</u>	0.627 $\pm$ 0.001	0.786 $\pm$ 0.029	0.787 $\pm$ 0.029	0.787 $\pm$ 0.030
<i>With the anchored action of MemTrial</i>							
Memory-free	1.985 $\pm$ 0.008	1.998 $\pm$ 0.004	10.34 $\pm$ 0.02	<u>0.630 $\pm$ 0.001</u>	0.789 $\pm$ 0.030	0.789 $\pm$ 0.032	0.788 $\pm$ 0.030
Self-consistency	1.980 $\pm$ 0.022	2.000 $\pm$ 0.005	10.32 $\pm$ 0.01	0.635 $\pm$ 0.002	<u>0.788 $\pm$ 0.032</u>	0.788 $\pm$ 0.032	0.788 $\pm$ 0.032
Similarity retrieval (top-2)	1.961 $\pm$ 0.021	2.003 $\pm$ 0.010	10.36 $\pm$ 0.01	0.629 $\pm$ 0.006	0.787 $\pm$ 0.029	0.788 $\pm$ 0.030	0.790 $\pm$ 0.031
Similarity retrieval (top-4)	1.984 $\pm$ 0.011	2.006 $\pm$ 0.003	10.34 $\pm$ 0.03	0.627 $\pm$ 0.006	0.787 $\pm$ 0.029	0.787 $\pm$ 0.028	0.789 $\pm$ 0.031
FinMem	1.981 $\pm$ 0.041	2.005 $\pm$ 0.001	10.35 $\pm$ 0.00	0.621 $\pm$ 0.002	0.787 $\pm$ 0.030	0.796 $\pm$ 0.030	0.789 $\pm$ 0.031
MemRL	1.963 $\pm$ 0.044	1.998 $\pm$ 0.007	10.36 $\pm$ 0.03	0.625 $\pm$ 0.002	0.787 $\pm$ 0.030	0.792 $\pm$ 0.031	0.790 $\pm$ 0.030
Reflexion	1.964 $\pm$ 0.042	1.999 $\pm$ 0.001	10.30 $\pm$ 0.01	0.621 $\pm$ 0.006	0.789 $\pm$ 0.029	0.790 $\pm$ 0.029	0.790 $\pm$ 0.029
ExpeL	1.971 $\pm$ 0.015	2.007 $\pm$ 0.003	10.34 $\pm$ 0.01	0.622 $\pm$ 0.001	0.787 $\pm$ 0.029	0.803 $\pm$ 0.032	0.791 $\pm$ 0.029
Uplift credit	1.969 $\pm$ 0.012	2.004 $\pm$ 0.003	10.35 $\pm$ 0.01	0.628 $\pm$ 0.003	0.787 $\pm$ 0.030	0.804 $\pm$ 0.034	0.792 $\pm$ 0.031
Counterfactual selection	1.970 $\pm$ 0.026	2.006 $\pm$ 0.009	10.34 $\pm$ 0.01	0.629 $\pm$ 0.005	<u>0.788 $\pm$ 0.031</u>	<u>0.838 $\pm$ 0.040</u>	<u>0.802 $\pm$ 0.032</u>
Draft averaging (= test never passes)	1.976 $\pm$ 0.005	2.007 $\pm$ 0.001	10.37 $\pm$ 0.00	0.629 $\pm$ 0.001	0.787 $\pm$ 0.029	0.788 $\pm$ 0.029	0.789 $\pm$ 0.029
Hedge	1.984 $\pm$ 0.006	<u>2.013 $\pm$ 0.000</u>	10.39 $\pm$ 0.01	0.626 $\pm$ 0.001	0.787 $\pm$ 0.030	0.788 $\pm$ 0.030	0.788 $\pm$ 0.030

call *falling* every phase from a peak of this index to the following trough in which the index lost at least 5%; all other periods are *rising*. This gives one falling phase on PortBench (August to October 2023, -8.7%), one on InvestorBench (October 2020, -8.7%) and four on ClassAlloc, the largest of which is the 2022 bear market (-20.6%). **Survival.** A run survives at a given shortfall if its wealth never fell further behind the wealth of 1/N than this shortfall at any time in the test period.

Figure D.1 shows the wealth relative to 1/N in every setting, Figure D.2 the wealth itself, Figure D.3 InvestorBench with the other LLMs and temperatures, and Figure D.5 ClassAlloc with the other LLMs. All use gpt-4.1-mini unless stated and show the mean over three seeds, with a band of $\pm$ sd for MemTrial; shading marks the market phases and the dotted line the LLM’s knowledge cutoff, as in Figure 3. Figures D.3 and D.5 show the balanced investor; on InvestorBench, the experience pool, retrieval and test days are those of the main run. Table D.6 gives the relative wealth at the end of the test period and accumulated over the falling and the rising phases, Table D.7 the share of runs whose wealth never fell further behind that of 1/N than a given shortfall, with the median over runs of the largest shortfall, and Table D.8 the cumulative net return, annualized Sharpe ratio and maximum drawdown over the test period, computed from the wealth paths of Figure D.2. In Tables D.6 and D.8, each seed averages the settings of a benchmark (PortBench: two configurations and three investors; InvestorBench and ClassAlloc: eight LLMs and three investors); Table D.7 pools the 54 runs of each seed over the three benchmarks. All three report the mean $\pm$ sd over the three seeds. On PortBench, the experience-learning agents were ahead of 1/N in mid-2024 in the Full configuration; the last test month (November 2024) reverses this: the equal-weighted cryptocurrency class of the universe gained 129%, and 1/N, which holds 10% in it for the balanced investor, gained 15%, whereas the agents’ drafts held 0–7%. Without this month, Similarity retrieval (top-4) (1.50% per month) and Hedge (1.49%) come out ahead of 1/N and MemTrial (1.46% each) on PortBench-Full; dropping any other

single test month leaves MemTrial the best LLM-based method on both PortBench variants. On ClassAlloc, the memory-free LLM and most memory-based methods end above 1/N for the balanced and aggressive investors, with their gains made before the knowledge cutoff of gpt-4.1-mini; after it, every LLM-based method trails 1/N, MemTrial the least for every investor.

Summary over all settings. Every method gains over the test periods (for the balanced investor, 1/N ends at 1.61, 1.19 and 1.82 times its initial wealth on PortBench-Full, InvestorBench and ClassAlloc), so the methods differ mainly relative to 1/N. Averaged over all settings, including every LLM, the experience-learning agents fall behind 1/N on PortBench and InvestorBench mostly while markets rise (by 9.8–11.5% and 2.0–4.1% over the rising phases), and on ClassAlloc they lose 1.5–3.5% against 1/N over the falling phases, which include the 2022 bear market (Table D.6). Over the falling phases of InvestorBench and ClassAlloc, MemTrial loses less than the memory-free LLM and every experience-learning agent (0.06% and 0.9%, against 0.38–0.65% and 1.5–3.5%). Its terminal wealth is the highest of all LLM-based methods in 38 of the 39 PortBench and InvestorBench settings and within 1.4% of 1/N, and its Sharpe ratio and maximum drawdown are the closest to those of 1/N (Table D.8). On ClassAlloc, the memory-free LLM ends 0.3% above 1/N.

D.7 Hyperparameter analysis

We vary one hyperparameter at a time around the frozen setting: the trust test (Figure D.4 and Table D.9), the number of retrieved experiences k and the design (Table D.10), and the strength of the anchor (Appendix D.8, Table D.11). Tables report the mean $\pm$ sd over seeds, in % per month (bp per day on InvestorBench). In Table D.10, *Drafts* is the number of drafts per decision; a draft is one LLM call on InvestorBench and one run of the three-call agent on PortBench, so the LLM cost grows in proportion. On the real benchmarks, drafts were generated for all 16 subsets of the four retrieved experiences, so every design with $k \leq 4$ is evaluated offline from these drafts, without new LLM calls. In PlantedMem, drafts are resampled, which

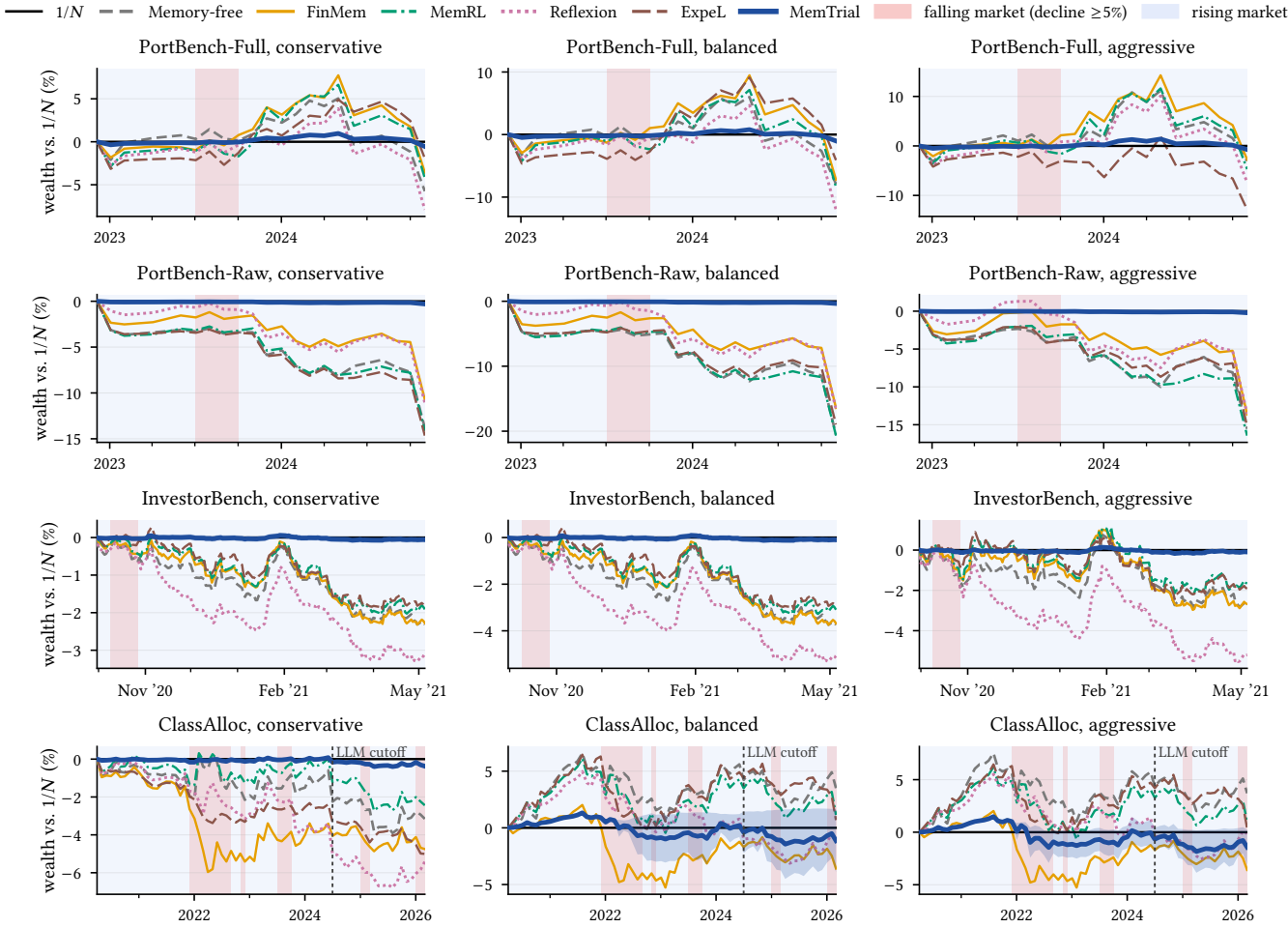


Figure D.1: Wealth relative to $1/N$ in every setting. MemTrial stays within 2% of $1/N$ throughout, whereas the other agents fall up to 21% behind.

Table D.6: Wealth relative to $1/N$ (%) at the end and by market phase. MemTrial stays within 1% of $1/N$ in every phase; the other agents lose most in the rising phases of PortBench and InvestorBench.

Method	PortBench			InvestorBench			ClassAlloc		
	End	Falling	Rising	End	Falling	Rising	End	Falling	Rising
Memory-free	-11.10 ± 0.46	-0.65 ± 0.58	-10.50 ± 0.61	-3.15 ± 0.36	-0.65 ± 0.06	-2.52 ± 0.39	$+0.31 \pm 0.97$	-1.59 ± 0.25	$+2.08 \pm 0.89$
FinMem	-9.05 ± 1.93	$+0.75 \pm 1.48$	-9.80 ± 0.96	-2.54 ± 0.20	-0.41 ± 0.03	-2.14 ± 0.22	-4.62 ± 0.25	-3.17 ± 0.10	-1.33 ± 0.35
MemRL	-11.44 ± 2.82	-0.93 ± 0.38	-10.59 ± 2.64	-2.46 ± 0.10	-0.50 ± 0.11	-1.97 ± 0.11	-2.50 ± 0.22	-3.53 ± 0.20	$+1.35 \pm 0.07$
Reflexion	-11.42 ± 1.16	$+0.11 \pm 1.31$	-11.50 ± 1.78	-4.60 ± 0.10	-0.52 ± 0.06	-4.09 ± 0.11	-2.53 ± 0.87	-1.45 ± 0.67	-0.98 ± 0.71
ExpeL	-11.29 ± 0.88	-0.05 ± 1.03	-11.27 ± 0.83	-2.75 ± 0.16	-0.38 ± 0.03	-2.37 ± 0.18	-3.88 ± 0.39	-3.44 ± 0.26	$+0.17 \pm 0.33$
MemTrial (ours)	-0.53 ± 0.04	$+0.03 \pm 0.03$	-0.55 ± 0.03	-0.21 ± 0.06	-0.06 ± 0.02	-0.15 ± 0.04	-0.33 ± 0.30	-0.94 ± 0.15	$+0.64 \pm 0.18$
Falling / rising periods	3 / 17 months			12 / 137 days			17 / 54 months		

also allows $k = 5$ and 6; these results come from a separate run on 50 test seeds (15–64) with common random numbers across designs, in which the default scores 0.868% per month (0.87 on 100 seeds in Table 3), and are shown with three decimals. Trust rates, with sd, are in Table D.3.

We make the following observations. (1) *The test level trades protection against speed.* Levels up to 0.05 give identical results on

PortBench, because at these levels the test passes on at most 0.3% of PortBench decisions. At $\alpha = 0.1$ and 0.2, chance forward scores pass the test on 5% and 12% of PortBench decisions; $\alpha = 0.1$ gains 0.03 pp on PortBench-Full but costs 0.03 pp on PortBench-Raw, and $\alpha = 0.2$ costs 0.03 and 0.08 pp, with a larger spread across seeds. In PlantedMem, a looser test trusts the values earlier and more often (on 28% of core decisions at $\alpha = 0.01$ against 41% at

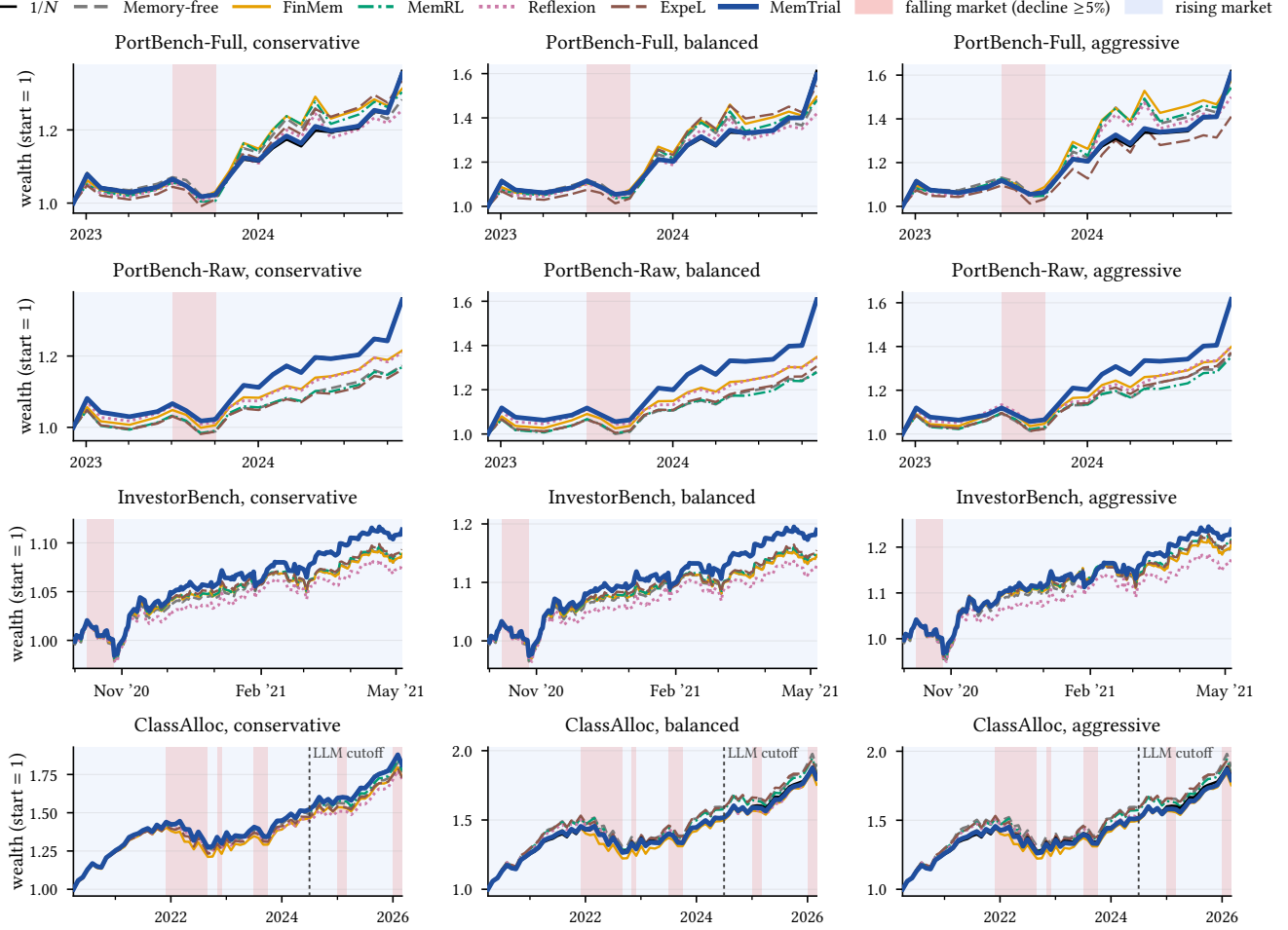


Figure D.2: Wealth in every setting (start = 1). MemTrial tracks 1/N; on PortBench and InvestorBench, the others end below it.

Table D.7: Survival against 1/N (% of runs). MemTrial never falls 5% behind 1/N; the others do in 22–44% of runs.

Method	Shortfall < 2%	Shortfall < 5%	Shortfall < 10%	Median largest shortfall (%)
Memory-free	27.8 ± 3.7	77.8 ± 3.7	94.4 ± 0.0	3.15 ± 0.19
FinMem	19.1 ± 3.9	67.9 ± 3.9	90.7 ± 3.2	3.59 ± 0.12
MemRL	31.5 ± 4.9	74.7 ± 1.1	93.2 ± 1.1	3.04 ± 0.26
Reflexion	14.2 ± 4.3	55.6 ± 1.9	88.9 ± 3.7	4.60 ± 0.24
ExpeL	26.5 ± 2.1	69.8 ± 3.9	83.3 ± 3.2	3.01 ± 0.36
MemTrial (ours)	93.2 ± 1.1	100.0 ± 0.0	100.0 ± 0.0	0.20 ± 0.01

$\alpha = 0.2$), which helps in the informative regime (0.84% against 0.88%). On InvestorBench, all levels stay within 0.05 bp per day of each other. The frozen level $\alpha = 0.05$ sits between these regimes. (2) *The minimum number of forward scores (5, 10 or 20) changes the InvestorBench result by at most 0.03 bp per day and no other result at two decimals.* (3) *More retrieved experiences help when experiences matter, with diminishing returns per draft.* In PlantedMem, the core utility rises from 0.812% per month with $k = 2$ (4 drafts) to 0.868 with $k = 4$ (8 drafts), 0.896 with $k = 5$ (16 drafts) and 0.909 with

$k = 6$ (32 drafts): going from 8 to 16 drafts adds 0.03 pp, and doubling again to 32 drafts adds 0.01 pp. (4) *At equal cost, the half fraction is the better design*, because it measures one more experience with the same number of drafts. With 4 drafts, $k = 3$ in a half fraction beats $k = 2$ in a full factorial (0.829% against 0.812%); with 8 drafts, $k = 4$ in a half fraction matches $k = 3$ in a full factorial (0.868% against 0.865%) and trusts its values more often (35% against 29%); with 16 drafts, $k = 5$ in a half fraction beats $k = 4$ in a full factorial (0.896% against 0.889%). (5) *On the real benchmarks*, where the values are rarely trusted, k and the design barely matter on PortBench (1.95–2.01% per month). On InvestorBench, every other design passes the test more often (on 2–11% of decisions), on values that do not carry over to the next day, and loses 0.13–0.44 bp per day against the default; the full factorial at $k = 4$, for example, doubles the drafts and loses 0.41 bp per day. We therefore keep $k = 4$ with eight drafts: in PlantedMem its gain over the reference per draft (0.011 pp) is close to the largest (0.012 pp, for $k = 3$ in a half fraction), and on InvestorBench it is the best design. A larger k is worth its cost only when experiences are known to matter. (6) *The anchor matters more than its exact strength* (Table D.11). A uniform prior ($\alpha_0 = 0.5$) costs up to 0.18 pp on PortBench-Full, up to 0.29 pp on PortBench-Raw

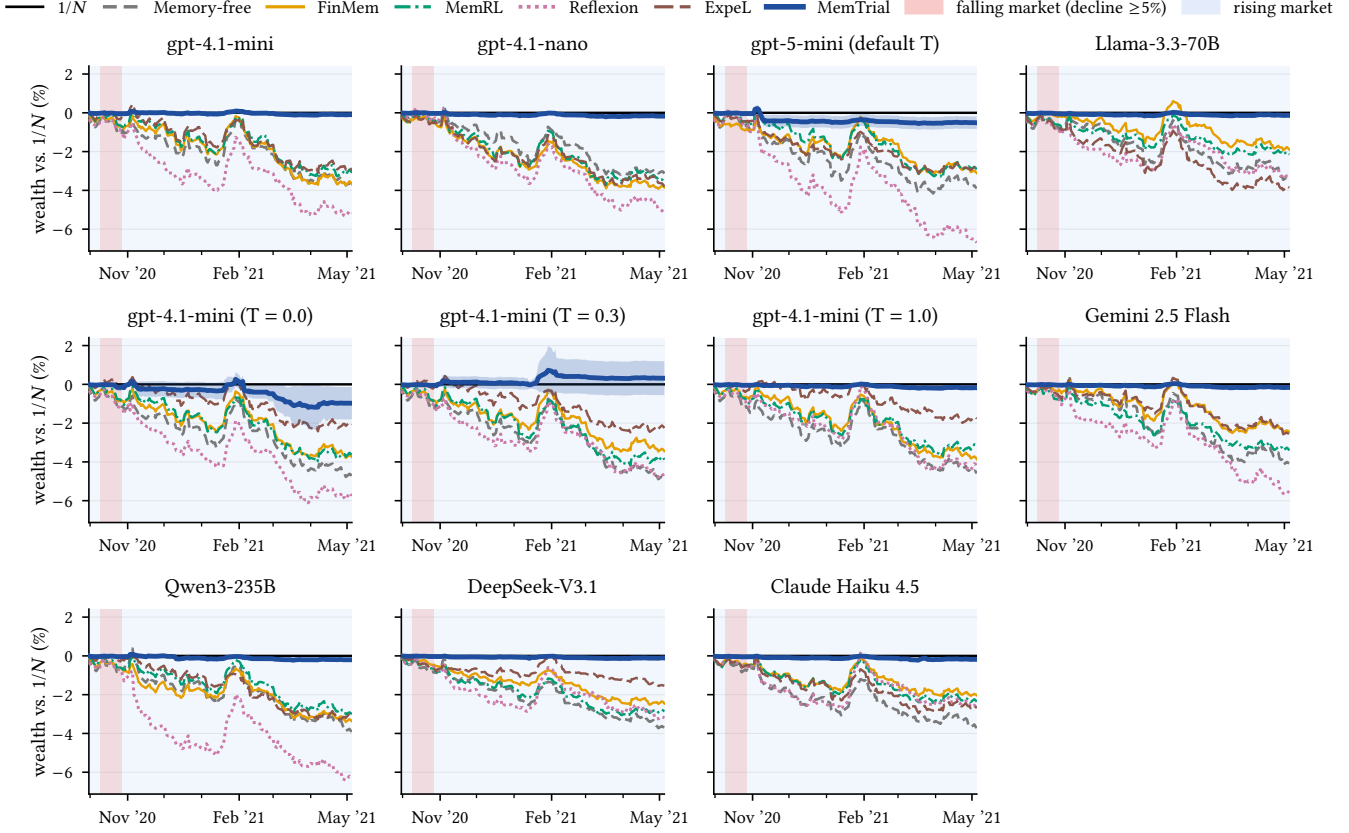


Figure D.3: InvestorBench with other LLMs and temperatures: wealth relative to $1/N$. MemTrial stays within 1.2% of $1/N$ in every configuration; the other agents end 1.5–6.7% behind.

Table D.8: Cumulative net return (CR, %), annualized Sharpe ratio (SR) and maximum drawdown (MDD, %). On every benchmark, MemTrial has the highest Sharpe ratio and the smallest drawdown of the LLM-based agents.

Method	PortBench			InvestorBench			ClassAlloc		
	CR	SR	MDD	CR	SR	MDD	CR	SR	MDD
$1/N$	53.1 ± 0.0	1.72 ± 0.00	5.7 ± 0.0	18.1 ± 0.0	2.87 ± 0.00	5.4 ± 0.0	82.0 ± 0.0	1.22 ± 0.00	12.0 ± 0.0
Memory-free	36.0 ± 0.8	1.43 ± 0.04	6.8 ± 0.8	14.4 ± 0.4	2.26 ± 0.06	6.0 ± 0.1	82.5 ± 1.8	1.14 ± 0.02	12.7 ± 0.3
FinMem	39.1 ± 3.0	<u>1.53 ± 0.08</u>	<u>6.2 ± 0.6</u>	15.1 ± 0.2	2.42 ± 0.03	5.8 ± 0.0	73.5 ± 0.5	1.12 ± 0.00	13.4 ± 0.0
MemRL	35.4 ± 4.4	1.39 ± 0.05	6.5 ± 0.4	15.2 ± 0.1	2.39 ± 0.03	5.9 ± 0.1	77.4 ± 0.4	1.11 ± 0.00	13.5 ± 0.3
Reflexion	35.4 ± 1.8	1.43 ± 0.09	6.8 ± 0.8	12.6 ± 0.1	2.02 ± 0.01	5.9 ± 0.0	77.4 ± 1.6	1.14 ± 0.01	13.0 ± 0.4
ExpeL	35.5 ± 1.2	1.47 ± 0.11	6.7 ± 0.7	14.8 ± 0.2	2.41 ± 0.03	5.8 ± 0.0	74.9 ± 0.7	1.11 ± 0.01	13.4 ± 0.3
MemTrial (ours)	<u>52.3 ± 0.1</u>	1.72 ± 0.01	5.7 ± 0.0	<u>17.9 ± 0.1</u>	<u>2.84 ± 0.01</u>	<u>5.5 ± 0.0</u>	81.4 ± 0.5	<u>1.19 ± 0.01</u>	<u>12.3 ± 0.2</u>

and 0.40–0.74 bp per day on InvestorBench, whereas with $\alpha_0 \geq 0.9$ every λ_0 stays within 0.06 pp per month and 0.06 bp per day of the default; in PlantedMem, all twelve settings give 0.86–0.87% per month on the core regimes.

D.8 Strength of the anchor

Table D.11 varies the prior weight α_0 on the reference and the regularization strength λ_0 of Eq. (11) (mean ± sd over seeds; % per month, bp per day on InvestorBench; default $\alpha_0 = 0.9$ and $\lambda_0 = 4$). Its last row is the extreme $\alpha_0 = 1$, which always keeps the reference when the values are not trusted. It is the best choice where $1/N$

cannot be beaten, 2.2%, 0.7% and 0.8% better than MemTrial on PortBench-Full, PortBench-Raw and InvestorBench, but it cannot move towards better drafts: it is 1.2% worse on ClassAlloc and 4.4% worse in PlantedMem when $1/N$ is the reference (Table D.15). The default $\alpha_0 = 0.9$ gives up at most 2.2% where the reference is strong and keeps the ability to follow better drafts.

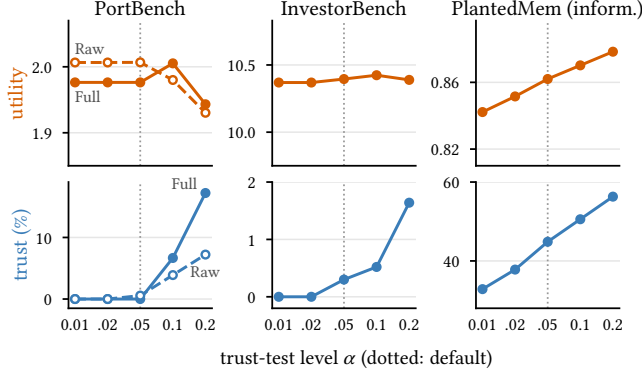
The best λ_0 differs between the two PortBench configurations (16 on Full, 1 on Raw), and the default is within 0.03 pp of the best setting on both. A higher prior ($\alpha_0 = 0.95$) would have done slightly better on the real benchmarks, where the reference is strong, but by the trade-off shown in §4.5 it would adapt more slowly when the reference is weak.

Table D.9: Sensitivity to the trust test. Test levels up to 0.05 give identical PortBench results; looser levels help only in the informative PlantedMem regime.

Hyperparameter	Value	PortBench-Full	PortBench-Raw	InvestorBench	PlantedMem (core)	PlantedMem (informative)
Test level α	0.01	<u>1.98 $\pm$ 0.00</u>	2.01 $\pm$ 0.00	10.37 $\pm$ 0.00	0.87 $\pm$ 0.04	0.84 $\pm$ 0.07
	0.02	<u>1.98 $\pm$ 0.00</u>	2.01 $\pm$ 0.00	10.37 $\pm$ 0.00	0.87 $\pm$ 0.04	0.85 $\pm$ 0.07
	0.05 (default)	<u>1.98 $\pm$ 0.00</u>	2.01 $\pm$ 0.00	<u>10.40 $\pm$ 0.04</u>	0.87 $\pm$ 0.04	0.86 $\pm$ 0.07
	0.1	2.01 $\pm$ 0.05	<u>1.98 $\pm$ 0.04</u>	10.42 $\pm$ 0.09	0.87 $\pm$ 0.05	<u>0.87 $\pm$ 0.07</u>
	0.2	1.94 $\pm$ 0.05	1.93 $\pm$ 0.13	10.39 $\pm$ 0.07	0.87 $\pm$ 0.05	0.88 $\pm$ 0.07
Minimum forward scores	5	1.98 $\pm$ 0.00	2.01 $\pm$ 0.00	10.40 $\pm$ 0.04	0.87 $\pm$ 0.05	0.86 $\pm$ 0.07
	10 (default)	1.98 $\pm$ 0.00	2.01 $\pm$ 0.00	10.40 $\pm$ 0.04	0.87 $\pm$ 0.04	0.86 $\pm$ 0.07
	20	1.98 $\pm$ 0.00	2.01 $\pm$ 0.00	10.37 $\pm$ 0.00	0.87 $\pm$ 0.04	0.86 $\pm$ 0.06

Table D.10: Number of retrieved experiences k and the design. More experiences help when experiences matter, with diminishing returns per draft; at equal cost, the half fraction is the better design.

k	Design	Drafts	PortBench-Full	PortBench-Raw	InvestorBench	PlantedMem (core)	PlantedMem (informative)
2	full factorial	4	1.98 $\pm$ 0.01	2.01 $\pm$ 0.00	<u>10.27 $\pm$ 0.11</u>	0.812 $\pm$ 0.043	0.798 $\pm$ 0.042
3	half fraction	4	1.97 $\pm$ 0.02	2.01 $\pm$ 0.00	10.14 $\pm$ 0.21	0.829 $\pm$ 0.041	0.799 $\pm$ 0.047
3	full factorial	8	1.98 $\pm$ 0.01	2.01 $\pm$ 0.00	9.96 $\pm$ 0.36	0.865 $\pm$ 0.051	0.850 $\pm$ 0.070
4	half fraction (default)	8	1.98 $\pm$ 0.00	2.01 $\pm$ 0.00	10.40 $\pm$ 0.04	0.868 $\pm$ 0.045	0.862 $\pm$ 0.072
4	full factorial	16	1.97 $\pm$ 0.00	1.95 $\pm$ 0.11	9.99 $\pm$ 0.34	0.889 $\pm$ 0.047	0.903 $\pm$ 0.085
5	half fraction	16	—	—	—	<u>0.896 $\pm$ 0.044</u>	<u>0.914 $\pm$ 0.078</u>
6	half fraction	32	—	—	—	0.909 $\pm$ 0.048	0.932 $\pm$ 0.070

**Figure D.4: Effect of the trust-test level α (top: utility; bottom: trust rate). On real markets, a looser test acts on chance and loses; in PlantedMem, it acts earlier on values that hold.**

D.9 Number of retrieved experiences by regime

Table D.12 breaks the PlantedMem part of Table D.10 down by regime (% per month; mean $\pm$ sd over test seeds 15–64, each averaged over the three investors; trust rates in Table D.3). When experiences carry no signal (first two columns), every design stays within 0.01 pp of the reference, so a larger k adds no risk there; its gains come entirely from the regimes in which experiences matter.

Table D.11: Strength of the anchor ($\alpha_0 = 1$: always the reference). A uniform prior ($\alpha_0 = 0.5$) costs the most; with $\alpha_0 \geq 0.9$, every λ_0 stays close to the default.

α_0	λ_0	PortBench-Full	PortBench-Raw	InvestorBench	PlantedMem (core)
0.5	1	1.80 $\pm$ 0.02	2.02 $\pm$ 0.00	9.99 $\pm$ 0.03	0.87 $\pm$ 0.04
0.5	4	1.83 $\pm$ 0.02	1.92 $\pm$ 0.01	9.71 $\pm$ 0.02	0.87 $\pm$ 0.04
0.5	16	1.85 $\pm$ 0.02	1.72 $\pm$ 0.01	9.65 $\pm$ 0.04	<u>0.86 $\pm$ 0.04</u>
0.75	1	1.86 $\pm$ 0.03	2.02 $\pm$ 0.00	10.29 $\pm$ 0.05	0.87 $\pm$ 0.04
0.75	4	1.92 $\pm$ 0.01	1.98 $\pm$ 0.00	10.16 $\pm$ 0.02	0.87 $\pm$ 0.04
0.75	16	1.94 $\pm$ 0.01	1.89 $\pm$ 0.00	10.12 $\pm$ 0.01	0.87 $\pm$ 0.04
0.9	1	1.92 $\pm$ 0.03	2.02 $\pm$ 0.00	10.42 $\pm$ 0.05	0.87 $\pm$ 0.04
0.9	4 (default)	1.98 $\pm$ 0.00	<u>2.01 $\pm$ 0.00</u>	10.40 $\pm$ 0.04	0.87 $\pm$ 0.04
0.9	16	1.99 $\pm$ 0.01	1.97 $\pm$ 0.00	10.39 $\pm$ 0.04	0.87 $\pm$ 0.04
0.95	1	1.95 $\pm$ 0.02	2.02 $\pm$ 0.00	10.45 $\pm$ 0.05	0.87 $\pm$ 0.04
0.95	4	2.00 $\pm$ 0.00	<u>2.01 $\pm$ 0.00</u>	10.45 $\pm$ 0.04	0.87 $\pm$ 0.05
0.95	16	<u>2.01 $\pm$ 0.00</u>	2.00 $\pm$ 0.00	<u>10.46 $\pm$ 0.04</u>	0.87 $\pm$ 0.05
1	—	2.02 $\pm$ 0.00	2.02 $\pm$ 0.00	10.48 $\pm$ 0.05	0.87 $\pm$ 0.05

D.10 PlantedMem by regime

Tables D.13–D.15 break the PlantedMem results down by regime (mean $\pm$ sd over the 100 test seeds, each averaged over the three investors). The trust rate is the share of decisions on which a method acts on its learned values, and the reference is the average of eight memory-free drafts, except in Table D.15, where it is $1/N$.

When does MemTrial trust its values? Table D.13 shows how often the trust test passes in each regime of PlantedMem. When experiences carry no signal, MemTrial acts on its values on only 2–3% of decisions, below the nominal 5% level of a single test. The

Table D.12: PlantedMem utility by regime for each k and design (% per month). A larger k gains only where experiences matter.

Design	Drafts	No influence	Noise only	$\beta = 0.1$	$\beta = 0.25$	$\beta = 0.5$	Many exp.	Informative	Core avg.
Reference	8	0.782 ± 0.031	<u>0.782 ± 0.031</u>	0.782 ± 0.031	0.782 ± 0.031	0.782 ± 0.031	0.782 ± 0.031	0.782 ± 0.031	0.782 ± 0.031
$k = 2$, full factorial	4	0.784 ± 0.029	<u>0.782 ± 0.038</u>	0.785 ± 0.029	0.804 ± 0.048	0.904 ± 0.130	0.782 ± 0.029	0.798 ± 0.042	0.812 ± 0.043
$k = 3$, half fraction	4	0.784 ± 0.032	0.784 ± 0.042	0.787 ± 0.038	0.821 ± 0.061	0.970 ± 0.123	0.789 ± 0.036	0.799 ± 0.047	0.829 ± 0.041
$k = 3$, full factorial	8	0.782 ± 0.028	0.778 ± 0.032	0.784 ± 0.031	0.904 ± 0.089	1.078 ± 0.154	0.812 ± 0.048	0.850 ± 0.070	0.865 ± 0.051
$k = 4$, half fraction (default)	8	<u>0.783 ± 0.028</u>	0.777 ± 0.037	0.794 ± 0.045	0.901 ± 0.084	1.085 ± 0.114	0.836 ± 0.079	0.862 ± 0.072	0.868 ± 0.045
$k = 4$, full factorial	16	0.780 ± 0.030	0.778 ± 0.029	0.792 ± 0.049	0.950 ± 0.080	1.143 ± 0.134	0.869 ± 0.094	0.903 ± 0.085	0.889 ± 0.047
$k = 5$, half fraction	16	0.780 ± 0.032	0.775 ± 0.034	<u>0.800 ± 0.054</u>	<u>0.956 ± 0.081</u>	<u>1.170 ± 0.110</u>	<u>0.873 ± 0.075</u>	<u>0.914 ± 0.078</u>	<u>0.896 ± 0.044</u>
$k = 6$, half fraction	32	<u>0.783 ± 0.029</u>	0.776 ± 0.034	0.822 ± 0.073	0.981 ± 0.082	1.184 ± 0.126	0.921 ± 0.066	0.932 ± 0.070	0.909 ± 0.048

trust rate rises with the influence of experiences, from 14% at $\beta = 0.1$ to 65% at $\beta = 0.25$ and 85% at $\beta = 0.5$, and with 48 experiences it rises from 26% to 45% when content becomes informative, because the content-aware learner then predicts the contributions of rarely seen experiences. On PortBench, the test passed on 1 of 360 test decisions and on 14 of 1,008 decisions over all months, consistent with Table 1, where the effect of experiences on real decisions is rarely detectable. On InvestorBench, it passed on only 0.3% of decisions, even though experiences change most daily decisions detectably: the values learned from past days do not carry over to the following days, and the mean forward score of the learned values, averaged over investors and seeds, is -0.07 . The test thus responds to whether learned values transfer to unseen dates, not to whether experiences matter on a given date.

Where does MemTrial gain and where does it lose? Table D.14 breaks the PlantedMem results down by regime. When experiences carry no signal, acting on learned values is costly: without its trust test, MemTrial falls 0.01–0.03 pp below the reference, and Counterfactual selection falls up to 0.13 pp below it, whereas MemTrial stays at the reference. When experiences matter, MemTrial gains up to 0.30 pp over the reference, but it gives up 0.01–0.03 pp to Counterfactual selection, because it waits for forward evidence before acting; the exception is the informative regime, where content-aware value estimation puts it 0.01 pp ahead. Averaged over the core regimes, protection outweighs delay (+0.02 pp, $p = 0.001$; Table 3). Outcome credit (MemRL+) never gains from experiences, even at $\beta = 0.5$, because its credit follows the market.

What if the reference is weak? On PortBench, InvestorBench and PlantedMem, the reference is never beaten by the average draft, so always keeping it ($q = 1$) is as good as MemTrial; on ClassAlloc, where the average draft beats $1/N$ by 0.03 pp per month, it is 0.008 pp worse (Figure 4). To test the online learning of the anchored action, we replace the reference in PlantedMem with the $1/N$ portfolio, which ignores the investor’s risk aversion and is about 0.18 pp per month worse than the average draft there. Table D.15 shows that, if it always kept $1/N$ when its values are not trusted, MemTrial would stay near $1/N$ (0.71% per month on the core regimes). The anchored action instead learns to move towards the drafts (0.75%), and a uniform prior moves faster (0.81%). In every regime, the learned blend is at least as good as always keeping $1/N$. Never acting on its values, MemTrial would earn only 0.62%, so with a weak reference the learned values matter even more. The prior weight α_0 therefore sets a trade-off: a high prior protects when the reference is strong (Table D.11), and a low prior adapts faster when it is weak.

D.11 ClassAlloc before and after the knowledge cutoff

Table D.16 splits the ClassAlloc results at the knowledge cutoff of gpt-4.1-mini (June 2024; % per month, mean $\pm$ sd over seeds, averaged over the three investors). Before the cutoff, every LLM-based method except FinMem and Reflexion beats $1/N$, by up to 0.12 pp per month (Self-consistency); after it, every LLM-based method trails $1/N$, MemTrial the least (0.746% against 0.787%) and the memory-free draft by 0.08 pp. As on the other benchmarks, outcome credit tracks the market (Spearman correlation 0.97, Table 1), and the retrieved experiences change the decision detectably in 53–57 of the 71 months. Over all 71 months, the memory-free draft beats $1/N$ by 0.03 pp per month, but no form of memory improves on it: retrieval, the experience-learning agents and the credit-based learners earn 0.59–0.65%, against 0.66% for the memory-free draft and 0.69% for Self-consistency (Table 3). The difference between the periods before and after the cutoff is not significant ($p = 0.36$, Welch t -test). The trust test of MemTrial passed on none of the decisions of 2020, on 3% of those of 2021 and on 13–33% of those of each later year, and on 0%, 39% and 4% of the decisions of the three seeds. On the 93 decisions on which it passed, the trusted draft earned 0.05 pp less (s.e. 0.07) than the anchored blend it replaced, and more on 48% of them.

D.12 Other LLMs and temperatures

Table D.17 summarizes every configuration of §4.4: on InvestorBench, in bp per day ($1/N$ earns 10.45 in every configuration), with the experience pool, retrieval and test days of the main run; on ClassAlloc, in % per month ($1/N$ earns 0.623), where each LLM writes its own warm-up lessons. The temperature is 0.7 unless stated, and gpt-5-mini runs at the temperature fixed by its API. *Best other LLM-based method* is the best of the 12 LLM-based baselines; Improv. and the trust rate are defined as in Table 3, and p is the one-sided paired t -test over test dates. Tables D.18 and D.19 give every method on InvestorBench (the column for temperature 0.7 is the main run), and Table D.20 every method on ClassAlloc; all report the mean $\pm$ sd over seeds, averaged over the three investors.

In every InvestorBench configuration, the best LLM-based baseline earns 8.97–9.71 bp per day, $1/N$ 10.45 and MemTrial 9.84–10.54.

D.13 ClassAlloc with other LLMs

Table D.20 gives every method of Table D.17b on ClassAlloc, Figure D.5 the wealth relative to $1/N$, and Table D.21 the results before and after each LLM’s own knowledge cutoff (utility minus that of $1/N$, mean over seeds; *Months* counts the test months before

Table D.13: Trust rate (%) by PlantedMem regime. MemTrial rarely trusts its values when experiences carry no signal and mostly trusts them when experiences matter.

Trust rule	No influence	Noise only	$\beta = 0.1$	$\beta = 0.25$	$\beta = 0.5$	Many exp.	Informative
MemTrial (trust test)	2.3 $\pm$ 6.6	3.0 $\pm$ 10.1	13.7 $\pm$ 20.7	64.5 $\pm$ 23.3	84.8 $\pm$ 8.9	26.1 $\pm$ 22.6	44.9 $\pm$ 22.5
Per-experience learner only	1.5 $\pm$ 5.1	2.0 $\pm$ 8.3	10.5 $\pm$ 18.0	61.4 $\pm$ 24.0	83.5 $\pm$ 10.1	22.0 $\pm$ 20.7	22.0 $\pm$ 20.7
In-sample F-test	2.4 $\pm$ 6.7	2.1 $\pm$ 7.1	8.1 $\pm$ 13.7	52.4 $\pm$ 22.9	81.7 $\pm$ 10.5	14.8 $\pm$ 17.7	15.1 $\pm$ 17.8

Table D.14: Utility by PlantedMem regime (% per month). MemTrial stays at the reference when experiences carry no signal and gains when they matter.

Method	No influence	Noise only	$\beta = 0.1$	$\beta = 0.25$	$\beta = 0.5$	Many exp.	Informative
Reference (memory-free average)	0.788 $\pm$ 0.032	0.788 $\pm$ 0.032	0.788 $\pm$ 0.032	0.788 $\pm$ 0.032	0.788 $\pm$ 0.032	0.788 $\pm$ 0.032	0.788 $\pm$ 0.032
Outcome credit (MemRL $\uparrow$)	0.759 $\pm$ 0.097	0.658 $\pm$ 0.089	0.758 $\pm$ 0.090	0.755 $\pm$ 0.098	0.749 $\pm$ 0.144	0.760 $\pm$ 0.085	0.760 $\pm$ 0.085
Counterfactual selection	0.768 $\pm$ 0.097	0.659 $\pm$ 0.087	0.805 $\pm$ 0.093	0.926 $\pm$ 0.100	1.101 $\pm$ 0.138	0.852 $\pm$ 0.078	0.852 $\pm$ 0.078
MemTrial w/o trust test	0.776 $\pm$ 0.047	0.758 $\pm$ 0.056	0.803 $\pm$ 0.059	0.924 $\pm$ 0.092	1.094 $\pm$ 0.131	0.843 $\pm$ 0.061	0.882 $\pm$ 0.068
MemTrial, never acting on its values	0.788 $\pm$ 0.028	0.787 $\pm$ 0.029	0.788 $\pm$ 0.028	0.788 $\pm$ 0.029	0.788 $\pm$ 0.029	0.789 $\pm$ 0.029	0.789 $\pm$ 0.029
MemTrial (ours)	<u>0.787 $\pm$ 0.030</u>	0.782 $\pm$ 0.035	0.793 $\pm$ 0.043	0.912 $\pm$ 0.088	1.090 $\pm$ 0.124	0.821 $\pm$ 0.050	<u>0.862 $\pm$ 0.068</u>

Table D.15: PlantedMem with 1/N as the reference (% per month). The anchored blend learns to leave a weak reference.

Method	No influence	Noise only	$\beta = 0.1$	$\beta = 0.25$	$\beta = 0.5$	Many exp.	Informative	Core avg.
1/N	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000	0.575 $\pm$ 0.000	0.57 $\pm$ 0.00
Draft averaging	0.782 $\pm$ 0.032	0.715 $\pm$ 0.052	0.777 $\pm$ 0.031	0.764 $\pm$ 0.044	0.719 $\pm$ 0.080	0.775 $\pm$ 0.041	<u>0.775 $\pm$ 0.041</u>	<u>0.75 $\pm$ 0.03</u>
MemTrial, always 1/N when not trusted ($q = 1$)	0.578 $\pm$ 0.016	0.580 $\pm$ 0.021	0.596 $\pm$ 0.055	0.783 $\pm$ 0.124	1.030 $\pm$ 0.136	0.626 $\pm$ 0.073	0.700 $\pm$ 0.096	0.71 $\pm$ 0.05
MemTrial, never acting on its values	0.633 $\pm$ 0.009	0.608 $\pm$ 0.017	0.630 $\pm$ 0.009	0.624 $\pm$ 0.014	0.611 $\pm$ 0.026	0.626 $\pm$ 0.013	0.626 $\pm$ 0.013	0.62 $\pm$ 0.01
MemTrial, anchored ($\alpha_0 = 0.9$)	0.635 $\pm$ 0.015	0.612 $\pm$ 0.024	0.646 $\pm$ 0.047	<u>0.805 $\pm$ 0.114</u>	<u>1.033 $\pm$ 0.133</u>	0.669 $\pm$ 0.063	0.733 $\pm$ 0.085	<u>0.75 $\pm$ 0.05</u>
MemTrial, uniform prior ($\alpha_0 = 0.5$)	<u>0.735 $\pm$ 0.022</u>	<u>0.675 $\pm$ 0.041</u>	<u>0.736 $\pm$ 0.039</u>	0.852 $\pm$ 0.100	1.043 $\pm$ 0.130	<u>0.751 $\pm$ 0.055</u>	0.797 $\pm$ 0.073	0.81 $\pm$ 0.05

Table D.16: ClassAlloc before and after the knowledge cutoff of gpt-4.1-mini (% per month). Most LLM-based methods beat 1/N before the cutoff; after it, all trail it, MemTrial the least.

Method	All 71 months	50 months before cutoff	21 months after cutoff
1/N	0.623 $\pm$ 0.000	0.554 $\pm$ 0.000	0.787 $\pm$ 0.000
Minimum variance	0.230 $\pm$ 0.000	0.178 $\pm$ 0.000	0.353 $\pm$ 0.000
Memory-free	<u>0.657 $\pm$ 0.001</u>	<u>0.636 $\pm$ 0.008</u>	0.705 $\pm$ 0.017
Self-consistency	0.686 $\pm$ 0.007	0.672 $\pm$ 0.010	0.721 $\pm$ 0.013
Similarity retr. (top-2)	0.649 $\pm$ 0.033	0.616 $\pm$ 0.043	0.728 $\pm$ 0.008
Similarity retr. (top-4)	0.643 $\pm$ 0.039	0.621 $\pm$ 0.047	0.696 $\pm$ 0.027
FinMem $\uparrow$	0.594 $\pm$ 0.017	0.538 $\pm$ 0.021	0.725 $\pm$ 0.026
MemRL $\uparrow$	0.633 $\pm$ 0.012	0.612 $\pm$ 0.030	0.683 $\pm$ 0.033
Reflexion	0.590 $\pm$ 0.061	0.545 $\pm$ 0.072	0.697 $\pm$ 0.064
ExpeL	0.609 $\pm$ 0.006	0.593 $\pm$ 0.005	0.648 $\pm$ 0.012
Uplift credit $\uparrow$	0.645 $\pm$ 0.022	0.617 $\pm$ 0.031	0.711 $\pm$ 0.001
Counterfactual sel.	0.650 $\pm$ 0.033	0.632 $\pm$ 0.039	0.691 $\pm$ 0.047
Draft averaging	0.651 $\pm$ 0.008	0.632 $\pm$ 0.007	0.696 $\pm$ 0.012
Hedge	0.634 $\pm$ 0.008	0.601 $\pm$ 0.009	0.714 $\pm$ 0.012
MemTrial (ours)	0.613 $\pm$ 0.017	0.557 $\pm$ 0.009	<u>0.746 $\pm$ 0.050</u>
MemTrial w/o trust test	0.612 $\pm$ 0.005	0.572 $\pm$ 0.030	0.707 $\pm$ 0.061

and after the cutoff). For Claude Haiku 4.5 we use the end of its training data, July 2025 (its reliable-knowledge cutoff is February 2025); Qwen3-235B and DeepSeek-V3.1 have no published knowledge cutoff. Each LLM writes its own warm-up lessons, which are embedded and retrieved as in the main run. The memory-free draft beats 1/N over the whole test period with four of the eight LLMs (gpt-4.1-mini, gpt-5-mini, Qwen3-235B and Claude Haiku 4.5). Of the six LLMs whose knowledge cutoff is published, it trails 1/N after the cutoff with five (by 0.06–0.09 pp per month), which is

what one would expect if the LLMs had seen the earlier outcomes during pretraining; with Claude Haiku 4.5, it beats 1/N also in the eight months after the cutoff (by 0.08 pp). MemTrial stays within 0.021 pp per month of 1/N with every LLM; with gpt-5-mini and Qwen3-235B, whose memory-free drafts beat 1/N by the widest margins, its anchor gives up most of their advantage.

D.14 Look-ahead checks

The LLMs were trained on text that may describe the test periods, a source of look-ahead bias. We ran two checks on ClassAlloc, at a total cost of USD 3.87.

Memorization probe. For each of the 129 ClassAlloc decision dates (58 warm-up and 71 test months, July 2015 to March 2026), we gave each LLM only the date and asked for the return of each asset class over the 20 trading days for which a decision is held, with no market data and no memory (prompt in Appendix G.2); three draws per date, at the temperature of the runs. We compare the answers with the realized returns before and after each LLM’s knowledge cutoff in two ways. *Absolute recall* is the correlation over months between recalled and realized returns, averaged over the four risky classes. *Relative recall* is the correlation after removing each class’s mean and each month’s mean, that is, of which class did better than usual in which month, which is what matters for beating 1/N. Table D.22 gives the mean $\pm$ sd over the three draws; each LLM has 102–121 dates before its cutoff, warm-up months included, and 8–27 after it. Every LLM whose cutoff is published recalls the outcomes before its cutoff (absolute recall 0.10–0.63, relative recall 0.06–0.50) and hardly at all after it (−0.16 to +0.09 and −0.10 to +0.01). Gemini 2.5 Flash recalls the most, and Qwen3-235B and DeepSeek-V3.1,

Table D.17: Robustness to the LLM and its temperature, all configurations. MemTrial beats every LLM-based baseline in all eleven InvestorBench configurations and stays within 0.021 pp of 1/N with all eight LLMs on ClassAlloc.

LLM, temperature	Memory-free	Best experience-learning agent	Best other LLM-based method	MemTrial (ours)	MemTrial w/o trust test	Improv. (% , p)	Trust rate (%)
<i>(a) InvestorBench (bp per day)</i>							
gpt-4.1-mini, 0	7.74 $\pm$ 0.35	ExpeL 9.46 $\pm$ 0.09	ExpeL 9.46 $\pm$ 0.09	9.84 $\pm$ 0.51	9.69 $\pm$ 0.22	+4.0 $\pm$ 6.0 (0.32)	23.2 $\pm$ 7.9
gpt-4.1-mini, 0.3	7.63 $\pm$ 0.25	ExpeL 9.40 $\pm$ 0.15	ExpeL 9.40 $\pm$ 0.15	10.54 $\pm$ 0.32	9.33 $\pm$ 0.28	+12.1 $\pm$ 4.0 (0.09)	7.0 $\pm$ 12.1
gpt-4.1-mini, 0.7 (main)	8.43 $\pm$ 0.29	ExpeL 8.93 $\pm$ 0.32	Hedge 8.97 $\pm$ 0.23	10.40 $\pm$ 0.04	9.91 $\pm$ 0.45	+16.4 $\pm$ 3.2 (0.12)	0.3 $\pm$ 0.5
gpt-4.1-mini, 1.0	7.83 $\pm$ 0.47	ExpeL 9.51 $\pm$ 0.45	ExpeL 9.51 $\pm$ 0.45	10.36 $\pm$ 0.01	10.10 $\pm$ 0.54	+9.0 $\pm$ 4.8 (0.15)	0.0 $\pm$ 0.0
gpt-4.1-nano, 0.7	8.45 $\pm$ 0.74	MemRL 7.75 $\pm$ 0.55	Self-consistency 9.07 $\pm$ 0.39	10.30 $\pm$ 0.06	9.22 $\pm$ 1.25	+33.0 $\pm$ 7.4 (0.02)	2.1 $\pm$ 3.6
gpt-5-mini, default	8.36 $\pm$ 0.54	MemRL 8.98 $\pm$ 0.07	Hedge 9.19 $\pm$ 0.11	10.29 $\pm$ 0.12	9.48 $\pm$ 0.79	+14.7 $\pm$ 2.1 (0.14)	2.3 $\pm$ 0.9
Llama-3.3-70B, 0.7	8.34 $\pm$ 0.39	FinMem 9.18 $\pm$ 0.37	Counterfactual sel. 9.39 $\pm$ 0.44	10.38 $\pm$ 0.04	10.41 $\pm$ 0.16	+13.1 $\pm$ 4.0 (0.10)	0.8 $\pm$ 1.1
Qwen3-235B, 0.7	8.31 $\pm$ 0.21	MemRL 8.86 $\pm$ 0.64	Hedge 8.97 $\pm$ 0.21	10.07 $\pm$ 0.25	8.94 $\pm$ 0.20	+13.6 $\pm$ 8.6 (0.12)	9.2 $\pm$ 10.3
DeepSeek-V3.1, 0.7	8.37 $\pm$ 0.55	ExpeL 9.71 $\pm$ 0.14	ExpeL 9.71 $\pm$ 0.14	10.33 $\pm$ 0.12	9.57 $\pm$ 0.27	+6.4 $\pm$ 2.3 (0.13)	1.1 $\pm$ 1.0
Gemini 2.5 Flash, 0.7	7.66 $\pm$ 0.23	FinMem 8.97 $\pm$ 0.27	FinMem 8.97 $\pm$ 0.27	10.36 $\pm$ 0.00	9.53 $\pm$ 0.47	+15.5 $\pm$ 3.0 (0.09)	0.0 $\pm$ 0.0
Claude Haiku 4.5, 0.7	8.42 $\pm$ 0.10	FinMem 9.40 $\pm$ 0.11	FinMem 9.40 $\pm$ 0.11	10.35 $\pm$ 0.03	9.48 $\pm$ 0.69	+10.1 $\pm$ 1.6 (0.14)	1.3 $\pm$ 1.8
<i>(b) ClassAlloc (% per month)</i>							
gpt-4.1-mini (main)	0.657 $\pm$ 0.001	MemRL 0.633 $\pm$ 0.012	Self-consistency 0.686 $\pm$ 0.007	0.613 $\pm$ 0.017	0.612 $\pm$ 0.005	-3.2 $\pm$ 1.4 (0.65)	14.6 $\pm$ 21.7
gpt-4.1-nano	0.587 $\pm$ 0.007	Reflexion 0.575 $\pm$ 0.040	Counterfactual sel. 0.617 $\pm$ 0.016	0.609 $\pm$ 0.005	0.616 $\pm$ 0.007	+5.9 $\pm$ 7.8 (0.26)	8.5 $\pm$ 6.0
gpt-5-mini	0.713 $\pm$ 0.013	ExpeL 0.690 $\pm$ 0.013	Self-consistency 0.719 $\pm$ 0.012	0.643 $\pm$ 0.017	0.642 $\pm$ 0.015	-6.7 $\pm$ 0.7 (0.76)	16.3 $\pm$ 27.8
Llama-3.3-70B	0.619 $\pm$ 0.059	FinMem 0.589 $\pm$ 0.015	Memory-free 0.619 $\pm$ 0.059	0.622 $\pm$ 0.000	0.622 $\pm$ 0.013	+5.6 $\pm$ 2.5 (0.26)	0.9 $\pm$ 1.6
Qwen3-235B	0.688 $\pm$ 0.022	MemRL 0.609 $\pm$ 0.016	Self-consistency 0.698 $\pm$ 0.021	0.626 $\pm$ 0.001	0.640 $\pm$ 0.015	+2.8 $\pm$ 2.7 (0.38)	3.4 $\pm$ 2.4
DeepSeek-V3.1	0.597 $\pm$ 0.018	Reflexion 0.624 $\pm$ 0.049	Sim. retrieval (top-4) 0.625 $\pm$ 0.016	0.607 $\pm$ 0.013	0.604 $\pm$ 0.014	-2.7 $\pm$ 10.0 (0.65)	12.8 $\pm$ 11.7
Gemini 2.5 Flash	0.616 $\pm$ 0.012	ExpeL 0.613 $\pm$ 0.010	Sim. retrieval (top-2) 0.656 $\pm$ 0.026	0.621 $\pm$ 0.010	0.625 $\pm$ 0.039	+1.3 $\pm$ 0.7 (0.44)	21.9 $\pm$ 26.4
Claude Haiku 4.5	0.668 $\pm$ 0.020	MemRL 0.637 $\pm$ 0.006	Memory-free 0.668 $\pm$ 0.020	0.629 $\pm$ 0.000	0.618 $\pm$ 0.004	-1.2 $\pm$ 0.9 (0.56)	0.0 $\pm$ 0.0

Table D.18: InvestorBench with gpt-4.1-mini at four temperatures, all methods (bp per day). At every temperature, no LLM-based baseline beats 1/N and MemTrial is the best LLM-based method.

Method	gpt-4.1-mini, temperature			
	0	0.3	0.7 (main)	1.0
1/N	10.45 $\pm$ 0.00	10.45 $\pm$ 0.00	10.45 $\pm$ 0.00	10.45 $\pm$ 0.00
Minimum variance	5.53 $\pm$ 0.00	5.53 $\pm$ 0.00	5.53 $\pm$ 0.00	5.53 $\pm$ 0.00
Memory-free	7.74 $\pm$ 0.35	7.63 $\pm$ 0.25	8.43 $\pm$ 0.29	7.83 $\pm$ 0.47
Self-consistency	7.86 $\pm$ 0.17	7.81 $\pm$ 0.08	7.89 $\pm$ 0.21	7.83 $\pm$ 0.24
Similarity retrieval (top-2)	8.36 $\pm$ 0.14	8.73 $\pm$ 0.28	8.67 $\pm$ 0.19	8.60 $\pm$ 0.45
Similarity retrieval (top-4)	8.06 $\pm$ 0.10	7.91 $\pm$ 0.27	8.12 $\pm$ 0.61	8.65 $\pm$ 0.40
FinMem [†]	8.45 $\pm$ 0.17	8.66 $\pm$ 0.37	8.47 $\pm$ 0.17	8.36 $\pm$ 0.14
MemRL [†]	8.46 $\pm$ 0.25	8.32 $\pm$ 0.39	8.90 $\pm$ 0.39	8.77 $\pm$ 0.25
Reflexion	6.87 $\pm$ 0.15	7.59 $\pm$ 0.54	7.29 $\pm$ 0.66	7.93 $\pm$ 0.36
ExpeL	9.46 $\pm$ 0.09	9.40 $\pm$ 0.15	8.93 $\pm$ 0.32	9.51 $\pm$ 0.45
Uplift credit [†]	8.52 $\pm$ 0.28	8.35 $\pm$ 0.24	8.44 $\pm$ 0.25	8.44 $\pm$ 0.20
Counterfactual selection	8.50 $\pm$ 0.07	8.39 $\pm$ 0.12	8.54 $\pm$ 0.26	8.60 $\pm$ 0.65
Draft averaging	8.55 $\pm$ 0.06	8.51 $\pm$ 0.06	8.69 $\pm$ 0.09	8.69 $\pm$ 0.16
Hedge	8.70 $\pm$ 0.07	8.69 $\pm$ 0.12	8.97 $\pm$ 0.23	8.96 $\pm$ 0.10
MemTrial (ours)	9.84 $\pm$ 0.51	10.54 $\pm$ 0.32	10.40 $\pm$ 0.04	10.36 $\pm$ 0.01
MemTrial w/o trust test	9.69 $\pm$ 0.22	9.33 $\pm$ 0.28	9.91 $\pm$ 0.45	10.10 $\pm$ 0.54

whose cutoffs are not published, have an absolute recall of 0.27 and 0.22 over all months. The advantage of the memory-free draft before the cutoff does not grow with recall: with Gemini 2.5 Flash, which recalls the most, it beats 1/N by only 0.008 pp per month before the cutoff, against 0.157 pp with gpt-5-mini (Table D.21).

Dates hidden. We repeated the ClassAlloc runs of gpt-4.1-mini and gpt-5-mini (three seeds each, with the two extra draws of Self-consistency) with no calendar date shown to the LLM: the market context omits its date line, the warm-up lessons are stored and retrieved without their date, and the ExpeL insight prompt lists the lessons without their dates. Everything else is unchanged. The lessons and reflections are written by the LLM from the dateless context, and none of the 58 warm-up lessons of either run mentions a year or a month. Table D.23 compares these runs with those that

show the dates (mean $\pm$ sd over seeds, averaged over the three investors). Hiding the dates leaves the pattern of the memory-free draft intact: with gpt-4.1-mini, it beats 1/N by 0.13 pp per month before the cutoff and trails it by 0.10 pp after (0.08 and 0.08 with the dates shown); with gpt-5-mini, by 0.14 and 0.07 pp (0.16 and 0.06). Over all 71 test months, it beats 1/N by 0.06 and 0.07 pp (0.03 and 0.09 with the dates shown). MemTrial stays within 0.02 pp per month of 1/N (+0.006 with gpt-4.1-mini and -0.017 with gpt-5-mini; trust rates 0.0% and 6.6%) and, after the cutoff, trails it by 0.02 and 0.03 pp, against 0.10 and 0.07 pp for the memory-free draft.

Figure D.6 brings these checks together with Table D.21. The LLMs thus recall dated market outcomes, but the advantage of the memory-free draft before the cutoff does not depend on being told the date. It may come from recognizing a period from the returns

Table D.19: InvestorBench with eight LLMs, all methods (bp per day). With every LLM, no LLM-based baseline beats 1/N and MemTrial beats every LLM-based baseline.

Method	gpt-4.1-mini	gpt-4.1-nano	gpt-5-mini	Llama-3.3-70B	Qwen3-235B	DeepSeek-V3.1	Gemini 2.5 Flash	Claude Haiku 4.5
1/N	10.45 ± 0.00	10.45 ± 0.00	10.45 ± 0.00	10.45 ± 0.00	10.45 ± 0.00	10.45 ± 0.00	10.45 ± 0.00	10.45 ± 0.00
Minimum variance	5.53 ± 0.00	5.53 ± 0.00	5.53 ± 0.00	5.53 ± 0.00	5.53 ± 0.00	5.53 ± 0.00	5.53 ± 0.00	5.53 ± 0.00
Memory-free	8.43 ± 0.29	8.45 ± 0.74	8.36 ± 0.54	8.34 ± 0.39	8.31 ± 0.21	8.37 ± 0.55	7.66 ± 0.23	8.42 ± 0.10
Self-consistency	7.89 ± 0.21	9.07 ± 0.39	8.31 ± 0.31	8.26 ± 0.22	8.37 ± 0.17	8.63 ± 0.51	7.92 ± 0.23	8.48 ± 0.07
Similarity retrieval (top-2)	8.67 ± 0.19	6.89 ± 0.92	8.41 ± 0.17	8.90 ± 0.28	8.15 ± 0.44	8.66 ± 0.47	8.50 ± 0.41	8.98 ± 0.12
Similarity retrieval (top-4)	8.12 ± 0.61	7.71 ± 0.66	8.46 ± 0.24	8.51 ± 0.14	8.00 ± 0.32	8.55 ± 0.45	8.22 ± 0.08	8.83 ± 0.15
FinMem [†]	8.47 ± 0.17	7.47 ± 0.72	8.87 ± 0.22	9.18 ± 0.37	8.59 ± 0.48	8.94 ± 0.35	8.97 ± 0.27	9.40 ± 0.11
MemRL [†]	8.90 ± 0.39	7.75 ± 0.55	8.98 ± 0.07	9.10 ± 0.34	8.86 ± 0.64	8.84 ± 0.55	8.35 ± 0.22	9.38 ± 0.31
Reflexion	7.29 ± 0.66	6.58 ± 0.94	5.91 ± 1.45	8.33 ± 0.61	6.10 ± 1.13	8.24 ± 0.22	6.57 ± 0.82	9.05 ± 0.68
ExpeL	8.93 ± 0.32	7.38 ± 0.41	8.82 ± 0.51	7.79 ± 0.19	8.29 ± 0.27	9.71 ± 0.14	8.72 ± 0.21	9.11 ± 0.14
Uplift credit [†]	8.44 ± 0.25	6.62 ± 0.55	8.37 ± 0.33	9.04 ± 0.29	8.38 ± 0.32	8.82 ± 0.30	8.44 ± 0.18	8.71 ± 0.20
Counterfactual selection	8.54 ± 0.26	6.79 ± 0.44	9.01 ± 0.09	9.39 ± 0.44	8.68 ± 0.53	9.25 ± 0.31	8.61 ± 0.37	8.80 ± 0.26
Draft averaging	8.69 ± 0.09	8.42 ± 0.10	8.80 ± 0.11	9.25 ± 0.04	8.56 ± 0.09	9.14 ± 0.13	8.57 ± 0.07	8.93 ± 0.04
Hedge	8.97 ± 0.23	8.98 ± 0.21	9.19 ± 0.11	9.29 ± 0.16	8.97 ± 0.21	9.33 ± 0.23	8.96 ± 0.07	9.39 ± 0.05
MemTrial (ours)	10.40 ± 0.04	10.30 ± 0.06	10.29 ± 0.12	10.38 ± 0.04	10.07 ± 0.25	10.33 ± 0.12	10.36 ± 0.00	10.35 ± 0.03
MemTrial w/o trust test	9.91 ± 0.45	9.22 ± 1.25	9.48 ± 0.79	<u>10.41 ± 0.16</u>	8.94 ± 0.20	9.57 ± 0.27	9.53 ± 0.47	9.48 ± 0.69

Table D.20: ClassAlloc with eight LLMs, all methods (% per month). MemTrial stays within 0.021 pp of 1/N with every LLM; the memory-free LLM beats 1/N with four of them.

Method	gpt-4.1-mini	gpt-4.1-nano	gpt-5-mini	Llama-3.3-70B	Qwen3-235B	DeepSeek-V3.1	Gemini 2.5 Flash	Claude Haiku 4.5
1/N	0.623 ± 0.000	0.623 ± 0.000	0.623 ± 0.000	0.623 ± 0.000	0.623 ± 0.000	0.623 ± 0.000	0.623 ± 0.000	0.623 ± 0.000
Minimum variance	0.230 ± 0.000	0.230 ± 0.000	0.230 ± 0.000	0.230 ± 0.000	0.230 ± 0.000	0.230 ± 0.000	0.230 ± 0.000	0.230 ± 0.000
Memory-free	<u>0.657 ± 0.001</u>	0.587 ± 0.007	<u>0.713 ± 0.013</u>	0.619 ± 0.059	<u>0.688 ± 0.022</u>	0.597 ± 0.018	0.616 ± 0.012	0.668 ± 0.020
Self-consistency	0.686 ± 0.007	0.588 ± 0.014	0.719 ± 0.012	0.614 ± 0.017	0.698 ± 0.021	0.615 ± 0.024	0.618 ± 0.003	<u>0.656 ± 0.004</u>
Similarity retrieval (top-2)	0.649 ± 0.033	0.557 ± 0.000	0.701 ± 0.019	0.602 ± 0.009	0.599 ± 0.006	0.603 ± 0.009	0.656 ± 0.026	0.640 ± 0.006
Similarity retrieval (top-4)	0.643 ± 0.039	0.554 ± 0.019	0.655 ± 0.006	0.598 ± 0.005	0.631 ± 0.019	0.625 ± 0.016	0.586 ± 0.016	0.641 ± 0.009
FinMem [†]	0.594 ± 0.017	0.532 ± 0.017	0.632 ± 0.025	0.589 ± 0.015	0.564 ± 0.006	0.570 ± 0.008	0.567 ± 0.026	0.608 ± 0.008
MemRL [†]	0.633 ± 0.012	0.569 ± 0.030	0.661 ± 0.022	0.579 ± 0.024	0.609 ± 0.016	0.553 ± 0.015	0.598 ± 0.053	0.637 ± 0.006
Reflexion	0.590 ± 0.061	0.575 ± 0.040	0.659 ± 0.013	0.560 ± 0.003	0.550 ± 0.069	<u>0.624 ± 0.049</u>	0.585 ± 0.052	0.633 ± 0.031
ExpeL	0.609 ± 0.006	0.475 ± 0.025	0.690 ± 0.013	0.565 ± 0.021	0.606 ± 0.012	0.597 ± 0.007	0.613 ± 0.010	0.524 ± 0.017
Uplift credit [†]	0.645 ± 0.022	0.558 ± 0.002	0.677 ± 0.028	0.582 ± 0.007	0.640 ± 0.005	0.620 ± 0.022	0.617 ± 0.023	0.638 ± 0.011
Counterfactual selection	0.650 ± 0.033	<u>0.617 ± 0.016</u>	0.685 ± 0.004	0.604 ± 0.013	0.646 ± 0.007	0.622 ± 0.020	<u>0.640 ± 0.011</u>	0.624 ± 0.004
Draft averaging	0.651 ± 0.008	0.587 ± 0.003	0.677 ± 0.005	0.604 ± 0.011	0.652 ± 0.009	0.620 ± 0.011	0.622 ± 0.003	0.650 ± 0.002
Hedge	0.634 ± 0.008	0.583 ± 0.002	0.665 ± 0.011	0.602 ± 0.010	0.632 ± 0.009	0.602 ± 0.007	0.604 ± 0.012	0.640 ± 0.008
MemTrial (ours)	0.613 ± 0.017	0.609 ± 0.005	0.643 ± 0.017	<u>0.622 ± 0.000</u>	0.626 ± 0.001	0.607 ± 0.013	0.621 ± 0.010	0.629 ± 0.000
MemTrial w/o trust test	0.612 ± 0.005	0.616 ± 0.007	0.642 ± 0.015	<u>0.622 ± 0.013</u>	0.640 ± 0.015	0.604 ± 0.014	0.625 ± 0.039	0.618 ± 0.004

Table D.21: ClassAlloc before and after each LLM’s knowledge cutoff (utility minus that of 1/N, pp per month). With five of the six LLMs whose cutoff is published, the memory-free draft trails 1/N after the cutoff.

LLM	Cutoff	Months	Memory-free		MemTrial (ours)	
			before	after	before	after
gpt-4.1-mini	Jun 2024	50/21	+0.082	−0.082	+0.003	−0.041
gpt-4.1-nano	Jun 2024	50/21	−0.018	−0.078	−0.018	−0.007
gpt-5-mini	May 2024	49/22	+0.157	−0.059	+0.040	−0.024
Llama-3.3-70B	Dec 2023	44/27	+0.047	−0.087	+0.005	−0.009
Qwen3-235B	–	–	–	–	–	–
DeepSeek-V3.1	–	–	–	–	–	–
Gemini 2.5 Flash	Jan 2025	57/14	+0.008	−0.069	+0.001	−0.014
Claude Haiku 4.5	Jul 2025	63/8	+0.040	+0.079	+0.005	+0.013

shown, or from allocations that happened to suit the markets of 2020–2024. Either way, it applies to all LLM-based methods alike, and PlantedMem is free of it.

Table D.22: Memorization probe: correlation of the asset-class returns that each LLM recalls from a date alone with the realized ClassAlloc returns. Every LLM recalls the outcomes before its knowledge cutoff, and hardly any after it.

LLM	Cutoff	Absolute recall		Relative recall	
		before	after	before	after
gpt-4.1-mini	Jun 2024	+0.21 ± 0.02	−0.05 ± 0.12	+0.19 ± 0.01	−0.02 ± 0.03
gpt-4.1-nano	Jun 2024	+0.10 ± 0.04	0.00 ± 0.04	+0.06 ± 0.06	−0.10 ± 0.09
gpt-5-mini	May 2024	+0.22 ± 0.06	−0.04 ± 0.09	+0.20 ± 0.04	−0.10 ± 0.13
Llama-3.3-70B	Dec 2023	+0.40 ± 0.09	+0.09 ± 0.15	+0.36 ± 0.08	+0.01 ± 0.12
Qwen3-235B	–	+0.27 ± 0.03	–	+0.21 ± 0.05	–
DeepSeek-V3.1	–	+0.22 ± 0.09	–	+0.19 ± 0.05	–
Gemini 2.5 Flash	Jan 2025	+0.63 ± 0.03	−0.07 ± 0.02	+0.50 ± 0.02	−0.09 ± 0.06
Claude Haiku 4.5	Jul 2025	+0.27 ± 0.01	−0.16 ± 0.07	+0.27 ± 0.01	−0.07 ± 0.10

D.15 PortBench after the knowledge cutoff

To test on months that gpt-4.1-mini, the LLM of the PortBench runs, cannot have seen, we extended the monthly PortBench stream with

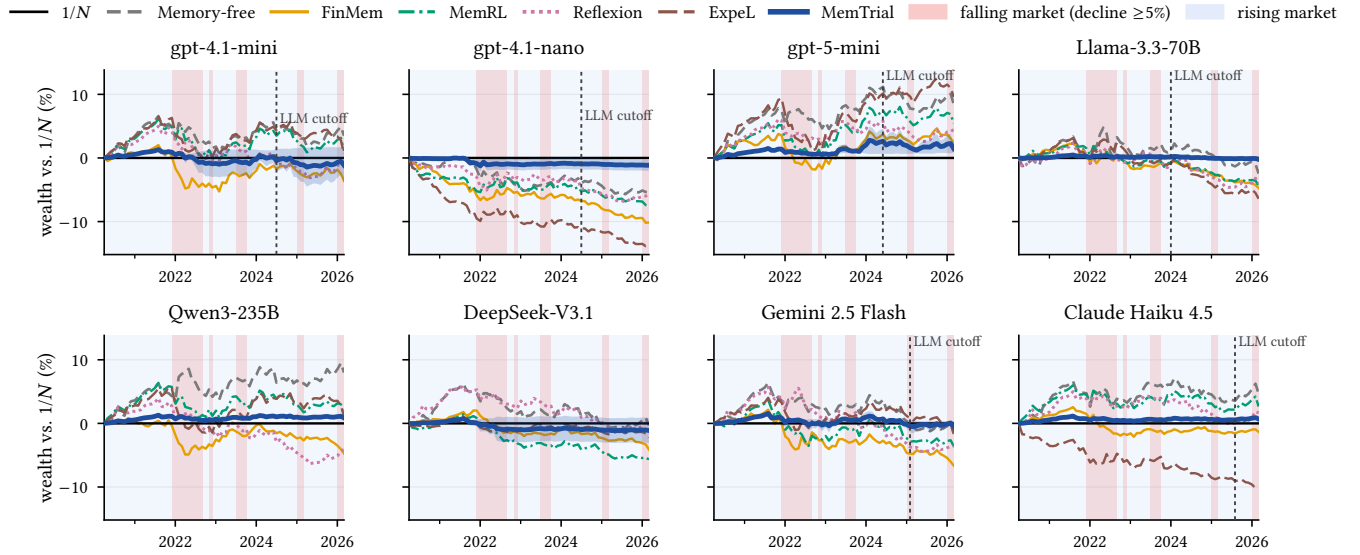


Figure D.5: ClassAlloc with other LLMs: wealth relative to $1/N$. MemTrial stays within 3% of $1/N$ with every LLM, whereas the memory-free LLM and the other agents move up to 14% away.

Table D.23: ClassAlloc with every calendar date hidden from the LLM (utility minus that of $1/N$, pp per month). Hiding the dates does not remove the memory-free draft’s advantage before the knowledge cutoff.

Method	gpt-4.1-mini				gpt-5-mini			
	dates shown		dates hidden		dates shown		dates hidden	
	before	after	before	after	before	after	before	after
Memory-free	$+0.082 \pm 0.008$	-0.082 ± 0.017	$+0.127 \pm \mathbf{0.017}$	-0.099 ± 0.027	$+0.157 \pm 0.020$	-0.059 ± 0.025	$+0.136 \pm 0.038$	-0.066 ± 0.006
Self-consistency	$+0.118 \pm \mathbf{0.010}$	-0.066 ± 0.013	$+0.125 \pm 0.030$	-0.104 ± 0.012	$+0.160 \pm \mathbf{0.013}$	-0.048 ± 0.016	$+0.118 \pm 0.028$	-0.068 ± 0.015
FinMem [†]	-0.016 ± 0.021	-0.062 ± 0.026	-0.047 ± 0.029	-0.076 ± 0.003	$+0.019 \pm 0.025$	-0.013 ± 0.045	-0.024 ± 0.016	-0.088 ± 0.035
MemRL [†]	$+0.058 \pm 0.030$	-0.104 ± 0.033	$+0.088 \pm 0.032$	-0.093 ± 0.050	$+0.077 \pm 0.046$	-0.048 ± 0.034	$+0.077 \pm 0.049$	-0.012 ± 0.041
Reflexion	-0.009 ± 0.072	-0.090 ± 0.064	-0.001 ± 0.095	-0.049 ± 0.036	$+0.066 \pm 0.026$	-0.032 ± 0.096	$+0.051 \pm 0.014$	-0.093 ± 0.019
ExpeL	$+0.039 \pm 0.005$	-0.140 ± 0.012	$+0.009 \pm 0.041$	$-0.015 \pm \mathbf{0.009}$	$+0.089 \pm 0.018$	$+0.017 \pm \mathbf{0.004}$	$+0.033 \pm 0.032$	-0.134 ± 0.025
Draft averaging	$+0.077 \pm 0.007$	-0.091 ± 0.012	$+0.087 \pm 0.003$	-0.100 ± 0.006	$+0.100 \pm 0.006$	-0.049 ± 0.007	$+0.107 \pm 0.004$	-0.020 ± 0.014
Counterfactual selection	$+0.078 \pm 0.039$	-0.096 ± 0.047	$+0.063 \pm 0.034$	-0.083 ± 0.016	$+0.118 \pm 0.005$	-0.064 ± 0.003	$+0.140 \pm \mathbf{0.014}$	$+0.008 \pm \mathbf{0.037}$
MemTrial (ours)	$+0.003 \pm 0.009$	$-0.041 \pm \mathbf{0.050}$	$+0.016 \pm 0.000$	$-0.017 \pm \mathbf{0.001}$	$+0.040 \pm 0.038$	-0.024 ± 0.029	-0.010 ± 0.047	-0.033 ± 0.015

the decision dates that follow 2024-12-02 under the same calendar rule (the first trading day of each month; a month is skipped when its 20-day holding window would overlap the previous one). This gives twelve dates from 2025-01-02 to 2026-02-02; March 2025, December 2025 and March 2026 are skipped by the overlap rule, and April 2026 has no complete holding window before the data end on 2026-04-16. Everything else is identical to the original months: the experience pool and the FinMem and MemRL providers (which receive no feedback after July 2019), the 19 arms per date, both agents, gpt-4.1-mini at temperature 0.7, three seeds, and the code that builds the inputs and runs the agents, which was reused unchanged. Before any new month was run, the inputs of all 59 original months were rebuilt with the same code and compared byte by byte with the frozen inputs, including the hash of every first LLM request; they were identical. The six Reflexion chains continue from their state after 2024-12-02. The evaluation reuses the code of the main results, run forward over all 71 dates without re-tuning, and reproduces the 2019–2024 results exactly.

The publisher’s price files have gaps in this period: 18 large equities (including AAPL, MSFT and NVDA) have no prices from 10 December 2024 to 24 January 2025 and after 19 February 2026, MATIC-USD ends on 24 March 2025 but remains in the agent’s market view until June 2025, and INDS lacks 24 October 2025. As for December 2024 in the original months, a month in which any execution held an asset without a price is dropped for that agent. This removes six of the twelve months (January, April, May, June and October 2025, and February 2026) and leaves 2025-02-03, 2025-07-01, 2025-08-01, 2025-09-02, 2025-11-03 and 2026-01-02 for both agents. The new runs cost USD 9.29 for 1,368 executions and USD 0.48 for the Reflexion chains.

Table D.24 reports the six new months and all eleven test months after the knowledge cutoff of gpt-4.1-mini (June 2024), averaged over the three investors. Over the eleven months, MemTrial is the best LLM-based method with both agents and stays close to $1/N$ (0.14 and 0.01 pp per month below it), while the other LLM-based methods trail $1/N$ by 0.36–1.57 pp; its values never passed the trust

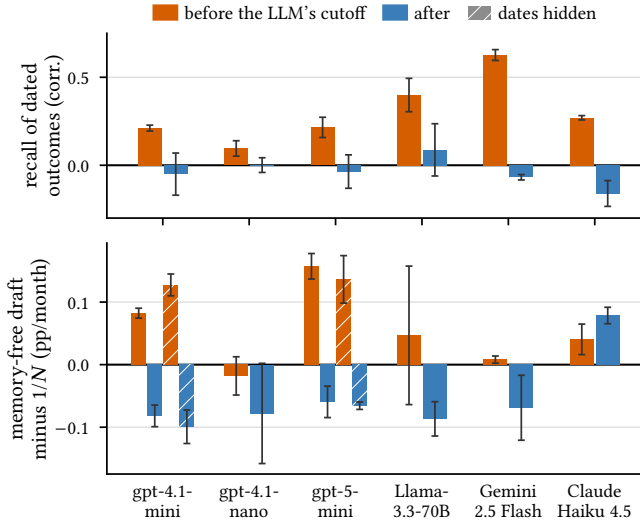


Figure D.6: ClassAlloc at each LLM’s knowledge cutoff. The LLMs recall dated outcomes from before their cutoffs, but hiding the dates does not remove the memory-free draft’s advantage before the cutoff.

test after the cutoff. In the six new months alone, the memory-free draft of PortBench-Full beat $1/N$ by 0.80 pp per month, and MemTrial, still anchored at $1/N$, gained only 0.06 pp; on PortBench-Raw, every LLM-based method is within 0.17 pp of $1/N$. With six months, no difference involving MemTrial is significant.

E Case Study: Two Decisions in the 2022 Bear Market

This appendix follows MemTrial through two monthly decisions on ClassAlloc (balanced investor), on 1 April and 1 September 2022. Both lie in the largest falling phase of the test period, in which the equal-weight equity index lost 21% between January and the end of September 2022 (Appendix D.6). The retrieved lessons depend only on the date, so all seeds see the same four lessons; the drafts, contributions and learned values differ by seed. In seed 1 the trust test passed on 1 April and narrowly failed on 1 September ($p = 0.058$); in seeds 0 and 2 it passed on neither date. The two dates thus show one decision that acted on the learned values and one that kept the anchored action. Table E.1a lists the eight designed drafts of seed 1 on both dates: the lessons each draft was given, its weights in % (Eq. equities, Bd. bonds, Cm. commodities, RE real estate) and its utility in % over the following 20 trading days, with the deployed draft in bold (on 1 September seed 1 deployed the anchored blend) and, in the last row, the realized return of each class over these 20 days. Table E.1b gives the contributions (Contr.) and learned values of the retrieved lessons in seed 1, in pp. Table E.1c gives the trust test and the action in each seed: the number n of forward scores, their mean $\bar{p}$, the p -value of the test, the action and its utility in %, with trusted decisions in bold; the column $1/N$ gives the utility of moving to $1/N$ from the same holdings, which differs slightly across seeds because of trading costs.

Retrieved lessons. Each lesson was written by the agent after one of its warm-up decisions, and its label gives the month of that decision. On 1 April 2022 the agent retrieved L18/04, L16/08, L17/05 and L15/12, and on 1 September L18/04, L16/08, L19/10 and L17/05.

L15/12 In December 2015, equities and real estate showed strong recent gains with moderate volatility, while commodities declined sharply. I overweighted equities and real estate, avoided commodities, and held cash. Despite this, bonds and real estate declined significantly, hurting returns. In similar scenarios, diversify more cautiously, temper bond exposure during rising volatility, and avoid complacency from recent positive trends.

L16/08 In August 2016, equities and bonds showed steady gains with low volatility, while commodities declined sharply and real estate dipped despite strong prior returns. I overweighted equities and real estate, underweighted commodities and cash. The portfolio outperformed due to equities and bonds. In similar conditions, maintain exposure to stable asset classes and reduce commodities when short-term momentum weakens.

L17/05 In May 2017, equities and real estate showed strong recent gains and moderate volatility, while commodities lagged with negative returns and high volatility. I overweighted equities and real estate, underweighted commodities, and outperformed the equal-weight portfolio. In similar conditions, favor assets with positive momentum and manageable volatility, reducing exposure to weak, volatile sectors like commodities.

L18/04 In April 2018, equities showed recent weakness but strong 6-month gains; bonds were stable with low volatility; commodities mildly positive; real estate mixed signals. I favored equities and bonds over cash, yielding a slightly better return than equal weights. In similar conditions, maintain diversified exposure but overweight assets with solid medium-term momentum and manageable volatility for balanced growth.

L19/10 In October 2019, real estate and equities showed strong recent returns with moderate volatility, while bonds declined. Allocating higher weights to real estate (35%) and equities (30%) capitalized on momentum, delivering a portfolio return above the equal-weight benchmark. In similar cases, emphasize assets with strong momentum and moderate risk, but maintain diversification to manage downside risks.

1 April 2022. Equities had fallen 5.4% over three months but risen 3.2% in the last month, and commodities had risen 21.6% over three months. Three of the lessons (L18/04, L16/08 and L17/05) describe months in which overweighting assets with positive momentum paid off; L15/12 describes one in which it did not and warns against complacency after recent gains. All eight drafts of seed 1, including the one without lessons, put 35% in equities ($1/N$: 20%) and 20–30% in real estate (Table E.1a), so this tilt comes from the LLM’s reading of the market rather than from the lessons. The lessons changed the drafts only slightly, and mostly for the worse: three of the four contributions are negative, and the drafts that used L17/05 earned 0.42 pp less on average than those that did not. The 11 forward scores of seed 1 so far had a mean of 0.24 and passed the trust test ($p = 0.045$), so MemTrial deployed the draft with the highest total learned value (L18/04 and L15/12). Equities then fell 8.1% and real estate 7.5%, and the decision earned -5.48% , against -3.83% for $1/N$. In seeds 0 and 2 the test did not pass ($p = 0.12$ and 0.16), and the anchored action, with a weight of 0.835 on $1/N$, earned -4.08% and -4.10% .

1 September 2022. Every risky class had lost money over the past one, three and six months, and all eight drafts of seed 1 were

Table D.24: PortBench after the knowledge cutoff (% per month). Over the eleven months after the cutoff, MemTrial is the best LLM-based method and stays close to 1/N; in the six new months, the memory-free draft beats 1/N on PortBench-Full.

Method	New months (2025–2026)		All months after the cutoff	
	Full	Raw	Full	Raw
<i>Rule-based portfolios</i>				
1/N	1.025 ± 0.000	1.025 ± 0.000	2.075 ± 0.000	2.075 ± 0.000
Minimum variance	0.029 ± 0.000	0.029 ± 0.000	0.078 ± 0.000	0.078 ± 0.000
<i>LLM agents without experience learning</i>				
Memory-free	<u>1.825 ± 0.228</u>	1.055 ± 0.103	1.290 ± 0.277	1.363 ± 0.053
Self-consistency	1.867 ± 0.000	1.058 ± 0.000	1.328 ± 0.000	1.367 ± 0.000
Similarity retrieval (top-2)	1.131 ± 0.351	1.059 ± 0.141	0.883 ± 0.299	1.265 ± 0.229
Similarity retrieval (top-4)	1.350 ± 0.353	1.002 ± 0.069	1.007 ± 0.283	1.211 ± 0.044
<i>Experience-learning agents</i>				
FinMem	1.071 ± 0.293	0.874 ± 0.203	0.691 ± 0.347	1.183 ± 0.104
MemRL	1.358 ± 0.676	1.191 ± 0.147	0.916 ± 0.343	1.365 ± 0.050
Reflexion	0.869 ± 0.092	<u>1.091 ± 0.066</u>	0.502 ± 0.185	1.404 ± 0.153
ExpeL [†]	1.197 ± 0.103	0.920 ± 0.155	1.033 ± 0.174	1.270 ± 0.152
<i>Credit-based and ensemble learners</i>				
Uplift credit [†]	1.034 ± 0.528	1.051 ± 0.042	0.598 ± 0.232	1.425 ± 0.237
Counterfactual selection	1.794 ± 0.367	1.024 ± 0.182	1.217 ± 0.242	1.542 ± 0.423
Draft averaging	1.437 ± 0.109	1.004 ± 0.033	1.039 ± 0.047	1.413 ± 0.115
Hedge	1.499 ± 0.130	1.029 ± 0.023	1.234 ± 0.090	1.711 ± 0.032
MemTrial (ours)	1.086 ± 0.014	1.027 ± 0.000	<u>1.939 ± 0.011</u>	<u>2.064 ± 0.002</u>
Improv. (% , p)	−20.1 ± 49.7 (0.61)	−13.8 ± 12.3 (0.64)	+87.8 ± 17.9 (0.12)	+47.0 ± 10.9 (0.24)
MemTrial − 1/N	+0.061 ± 0.014	+0.001 ± 0.000	−0.135 ± 0.011	−0.011 ± 0.002
Trust rate of MemTrial (%)	0.0	0.0	0.0	0.0

Table E.1: Case study on ClassAlloc (balanced investor). The contributions separate the lessons although every draft lost money; in seed 1, acting on the values cost 1.66 pp in April, and keeping the anchored action gave up 2.02 pp in September.

(a)													
1 April 2022							1 September 2022						
Lessons in the draft	Eq.	Bd.	Cm.	RE	Cash	Utility	Lessons in the draft	Eq.	Bd.	Cm.	RE	Cash	Utility
none	35	10	25	20	10	−4.63	none	15	35	15	15	20	−5.43
L18/04, L16/08	35	10	20	25	10	−5.14	L18/04, L16/08	15	30	10	15	30	−5.05
L18/04, L17/05	35	10	15	30	10	−5.67	L18/04, L19/10	10	30	10	10	40	−3.89
L16/08, L17/05	35	15	10	30	10	−6.02	L16/08, L19/10	15	25	10	15	35	−4.81
L18/04, L15/12	35	15	15	25	10	−5.48	L18/04, L17/05	15	35	10	15	25	−5.29
L16/08, L15/12	35	15	10	25	15	−5.58	L16/08, L17/05	15	35	10	15	25	−5.29
L17/05, L15/12	35	10	15	30	10	−5.67	L19/10, L17/05	15	15	20	15	35	−4.64
all four	35	10	20	25	10	−5.14	all four	15	25	10	15	35	−4.81
							anchored blend	19	21	19	19	21	−5.91
1/N	20	20	20	20	20	−3.83	1/N	20	20	20	20	20	−6.06
Class return (%)	−8.1	−4.8	2.2	−7.5	0.0		Class return (%)	−9.5	−4.3	−1.8	−12.3	0.1	

(b)						(c)						
1 Apr.			1 Sep.			Date	Seed	n	ρ	p	Action	Utility
Lesson	Contr.	Value	Contr.	Value								1/N
L15/12	−0.11	+0.004	−	−		1 Apr.	0	18	0.17	0.123	anchored, $q = 0.835$	−4.08
L16/08	−0.11	−0.015	−0.18	−0.016			1	11	0.24	0.045	draft L18/04, L15/12	−5.48
L17/05	−0.42	−0.002	−0.21	−0.002			2	15	0.13	0.163	anchored, $q = 0.835$	−4.10
L18/04	+0.12	+0.004	+0.28	+0.003		1 Sep.	0	22	0.07	0.286	anchored, $q = 0.885$	−5.84
L19/10	−	−	+0.73	+0.005			1	16	0.17	0.058	anchored, $q = 0.875$	−5.91
							2	18	0.03	0.421	anchored, $q = 0.875$	−5.79

defensive, with 10–15% in equities and 20–40% in cash. The drafts that used L19/10 held 35–40% in cash, against 20–30% for those that did not, and L19/10 had the largest contribution (+0.73 pp). In seed 1 the test narrowly failed ($p = 0.058$; 16 forward scores, mean 0.17), so MemTrial kept the anchored action, with a weight of 0.875 on 1/N, which earned −5.91%, against −6.06% for 1/N; the draft it would have deployed (L18/04 and L19/10: 40% cash, 30% bonds, 10%

equities) earned −3.89%. In seeds 0 and 2 ($p = 0.29$ and 0.42), the anchored action earned −5.84% and −5.79%.

What the case shows. First, contributions are measured on the same date, so the part of the outcome that all drafts share cancels: on both dates every draft had a utility between −6.1% and −3.8%, which outcome credit would charge to every lesson used, yet the contributions separate the lessons by tenths of a pp. Second, the

Algorithm F.1 MemTrial at decision date t

Input: state s_t ; retrieved experiences $R_t = (h_1, \dots, h_k)$; design $\mathcal{D}$ (8 subsets of R_t); reference x^{ref} ; contributions of past dates; forward scores; FTRL statistics
Act, before the outcome

- 1: **for** each $S \in \mathcal{D}$ **do**
- 2: draft $x^S \sim \pi(\cdot \mid s_t, S)$ ▷ 8 drafts in parallel
- 3: fit the content-aware and the per-experience value models on past dates ▷ Eqs. (5)–(7)
- 4: predict $\hat{v}_j$ for every $h_j \in R_t$ with both models; keep both predictions for scoring
- 5: use the model with the higher mean forward score;
 $S^* = \arg \max_{S \in \mathcal{D}} \sum_{j \in S} \hat{v}_j$ ▷ Eq. (10)
- 6: **if** the trust test passes ($n \geq 10$ and $p_t < \alpha$, Eq. (9)) and $\sum_{j \in S^*} \hat{v}_j > 0$ **then**
- 7: deploy $x_t = x^{S^*}$ ▷ trusted
- 8: **else**
- 9: deploy $x_t = q_t x^{\text{ref}} + (1 - q_t) \bar{x}_t$, with q_t from Eq. (11) ▷ anchored
Learn, once the outcome is realized
- 10: $U(S) = u_t(x^S)$ for $S \in \mathcal{D}$; contributions c_j (Eq. (3)); centred y_j (Eq. (4))
- 11: forward score ρ_t of each model (Eq. (8)); update the FTRL statistics of Eq. (11)

contributions are noisy and the learned values are small: L18/04 helped on both dates (+0.12 and +0.28 pp) and L17/05 hurt on both (−0.42 and −0.21 pp), but after shrinkage all values in seed 1 are below 0.02 pp in absolute value, so the deployed draft is decided by small differences. Third, the trust test decides whether MemTrial takes on the drafts’ market exposure: in April, when every draft overweighted equities before a fall, acting on the values cost 1.66 pp against 1/ N ; in September, when every draft was defensive, the test narrowly failed and the anchored action gave up the 2.02 pp by which the selected draft beat it. Over all 71 months, acting on the values did not pay on ClassAlloc: on the 93 decisions on which the test passed, the trusted draft earned 0.05 pp less (s.e. 0.07) than the anchored blend it replaced, and the variant whose test never passes is 0.016 pp per month better (Figure 4). Keeping part of the anchor even on trusted decisions is a natural extension (Appendix H).

F Algorithms

Algorithm F.1 gives one date of MemTrial (§3.6). Algorithms F.2 and F.3 expand its learning steps.

G LLM Prompts

This appendix gives the prompts used on the three real benchmarks, copied from the logged requests. Text in angle brackets is filled in at run time, and [...] marks lines omitted here. On each benchmark, all LLM-based methods use the same prompts and differ only in the memory block that follows the market context (Table G.1); the other LLMs of §4.4 receive the same prompts unchanged. The drafts of all 16 subsets of the four retrieved lessons are logged on every date and seed, so the methods in the second row of Table G.1 differ only in which draft they deploy or how they combine the drafts. On InvestorBench and ClassAlloc, a decision is one chat completion with a system and a user message, and an invalid reply is answered

Algorithm F.2 Value estimation (line 3 of Algorithm F.1)

Input: centred contributions $y_{s,j}$ of past dates, each with the experience it belongs to; content features z_h ; grid G of (σ_β^2, τ^2)

- 1: **for** each experience h measured on $n_h \geq 1$ past dates **do**
- 2: $m_h \leftarrow$ mean of its centred contributions
- 3: $\sigma^2 \leftarrow$ pooled within-experience variance of the centred contributions
- 4: per-experience model: $\sigma_\beta^2 = 0$, $\tau^2 = \arg \max_G$ of the marginal likelihood ▷ Eq. (6)
- 5: content-aware model: $(\sigma_\beta^2, \tau^2) = \arg \max_G$ of Eq. (6)
- 6: **for** each model **do**
- 7: $\hat{v} \leftarrow (\sigma_\beta^2 Z Z^\top + \tau^2 I) K^{-1} m$; $\hat{\beta} \leftarrow \sigma_\beta^2 Z^\top K^{-1} m$ ▷ Eq. (7)
- 8: prediction for a retrieved experience h : $\hat{v}_h$ if $n_h \geq 1$, else $z_h^\top \hat{\beta}$ (0 for the per-experience model)

Output: the predictions of both models

Algorithm F.3 Trust test and anchored action (Algorithm F.1)

Input: forward scores of both models; utilities of the blends of earlier dates

- 1: selected model $\leftarrow$ per-experience model until both models have ≥ 10 forward scores, then the model with the higher mean forward score
- 2: $n, \bar{p}, s_p \leftarrow$ number, mean and sd of the selected model’s forward scores
- 3: $p \leftarrow \Pr(T_{n-1} \geq \bar{p}/(s_p/\sqrt{n}))$ ▷ Eq. (9)
- 4: $S^* \leftarrow \arg \max_{S \in \mathcal{D}} \sum_{j \in S} \hat{v}_j$, ties towards fewer experiences ▷ Eq. (10)
- 5: trusted $\leftarrow (n \geq 10) \wedge (p < \alpha) \wedge (\sum_{j \in S^*} \hat{v}_j > 0)$
- 6: **if** not trusted **then**
- 7: $q_t \leftarrow \arg \max_{q \in \{0, 0.005, \dots, 1\}} \text{mean}_{s < t} u_s(q x^{\text{ref}} + (1 - q) \bar{x}_s) - \lambda_n \text{KL}(q \parallel \alpha_0)$ ▷ Eq. (11)
- 8: after the outcome of date t : for each model, $\rho_t \leftarrow \text{corr}(\hat{v}_{t,\cdot}, y_{t,\cdot})$, appended to its forward scores; store u_t of every blend on the grid ▷ Eq. (8)

once with a retry message; on PortBench, a draft is one run of the benchmark’s three-stage agent, without retries.

Table G.1: Memory block of each LLM-based method in the decision prompt on InvestorBench and ClassAlloc.

Method	Memory block
Memory-free, Self-consistency	none
MemTrial, Similarity retrieval, Counterfactual selection, Uplift credit, Draft averaging	experiences block with a subset of the four retrieved lessons
FinMem	experiences block with four lessons ranked by similarity, recency and importance
MemRL	experiences block with four of the ten most similar lessons, ranked by learned value (ϵ -greedy)
Reflexion	reflections block with its three most recent reflections
ExpeL	insights block with at most eight insights distilled from all warm-up lessons

G.1 InvestorBench

System message (all calls).

You are a portfolio manager. You allocate a long-only portfolio across four U.S. stocks (HON, JNJ, MSFT, UVV) and cash. Each trading day you set target weights that are held until the next trading day’s close.

Decision prompt (user message), here for 25 February 2021 and the draft with all four retrieved lessons.

Date: 2021-02-25

Market data (returns in %; volatility = standard deviation of daily returns over 20 days, in %):
HON: close 190.74; 1-day -2.53; 5-day +1.07; 20-day +3.17; volatility 1.48
JNJ: close 148.73; 1-day +0.10; 5-day -1.19; 20-day -2.45; volatility 1.14
MSFT: close 222.30; 1-day -2.37; 5-day -6.07; 20-day -1.45; volatility 1.53
UVV: close 41.36; 1-day -1.53; 5-day -1.16; 20-day +7.24; volatility 2.02

News today:
- JNJ: The global Cold Pain Therapy Market is projected to grow from USD 1.6 billion in 2020 to USD 2.0 billion by 2025, at a CAGR of 4.4%, driven by the increasing prevalence of arthritis and musculoskeletal disorders and growing awareness of col
- JNJ: Lucid has announced the integration of its virtual whiteboard, Lucidspark, with Microsoft Teams, enhancing real-time collaboration and innovation for Teams users. This integration allows seamless movement between Lucidchart and Lucidspark d
[... 7 more headlines: at most three per stock, each cut at 240 characters]

Experiences from your past decisions (use them only where relevant):
[1] (2020-09-14) In a mixed market with positive news for JNJ and MSFT but recent negative 5-day returns for MSFT, I overweighted JNJ and MSFT, underweighted UVV and HON, and held some cash. The portfolio outperformed slightly. In similar cases, balance recent momentum with positive news, maintain diversification, and keep modest cash to manage uncertainty.
[2] (2020-08-18) In a mixed market with modest gains and losses, overweighting MSFT and JNJ aligned with positive trends but UVV's rebound was underestimated. Diversifying more evenly and maintaining a slightly higher cash position could reduce downside risk. In similar conditions, balance growth potential with volatility by avoiding concentration in fewer stocks despite recent strong performance.
[3] (2020-07-17) In a mixed market with strong news for JNJ and MSFT but recent declines in MSFT and UVV, I overweighted HON and JNJ for stability and growth potential. Despite a dip in HON, MSFT's rebound boosted returns. In similar cases, balance stable leaders with growth stocks showing recovery signs, maintaining some cash to manage volatility.
[4] (2020-07-16) In a mixed market with strong recent gains in HON and JNJ but weakness in MSFT and UVV, overweighting HON and JNJ while underweighting MSFT and UVV led to outperformance. For similar conditions, favor stocks with positive momentum and stable news, maintain some cash for flexibility, and cautiously reduce exposure to recent laggards to improve returns.

Decide the target weights for the next trading day. Respond with a JSON object only, for example {"HON": 0.2, "JNJ": 0.2, "MSFT": 0.2, "UVV": 0.2, "CASH": 0.2, "reason": "one short sentence"}. Weights must be non-negative and sum to 1.

Reply of gpt-4.1-mini (seed 0), as parsed.

```
{"HON": 0.25, "JNJ": 0.35, "MSFT": 0.2, "UVV": 0.1, "CASH": 0.1, "reason": "Overweight JNJ for positive news and stability; moderate HON for steady gains; reduce MSFT and UVV due to recent declines and higher volatility; keep cash for risk management."}
```

Retry message, sent once after an invalid reply.

Your reply was not a valid JSON weight object. Reply again with the JSON object only, weights non-negative and summing to 1.

Lesson prompt. After each warm-up decision, the agent sees the outcome and writes a lesson. The prompt for 2 July 2020:

<market context of the warm-up day, in the format of the decision prompt>

Your decision: HON 15%, JNJ 30%, MSFT 40%, UVV 5%, CASH 10%
Next trading day returns (%): HON +1.52, JNJ +1.43, MSFT +2.15, UVV +1.45
Your portfolio return: +1.59%; equal-weight portfolio (four stocks and cash): +1.31%.

Write one reusable lesson for future allocation decisions in at most 60 words: describe the market situation, what you did, what happened, and what you would do in a similar situation. Plain text only.

The lesson written, stored with its date as experience E000:

(2020-07-02) In a mixed market with strong MSFT momentum and positive JNJ news, overweighting MSFT and JNJ while maintaining modest exposure to HON and UVV outperformed an equal-weight approach. Future allocations should emphasize stocks with positive catalysts and momentum, balancing risk with cash to capture upside while managing volatility.

Reflection prompt (Reflexion): the lesson prompt with its last paragraph replaced by

Reflect on this decision in at most 60 words: what worked, what did not, and what you will do differently next time. Plain text only.

Insight prompt (ExpeL), run once on all 63 warm-up lessons:

Below are lessons you wrote after past daily decisions, each with the outcome of that decision.

- [2020-07-02; portfolio +1.59% vs equal weight +1.31%] In a mixed market with strong MSFT momentum and positive JNJ news, overweighting MSFT and JNJ while maintaining modest exposure to HON and UVV outperformed an equal-weight approach. Future allocations should emphasize stocks with positive catalysts and momentum, balancing risk with cash to capture upside while managing volatility.
[... 62 more lessons, one per warm-up day]

Compare the successful and unsuccessful decisions and extract at most 8 general insights for future daily allocation across HON, JNJ, MSFT, UVV and cash. Return a numbered list; each insight at most 30 words.

Insights and reflections blocks, which take the place of the experiences block for ExpeL and Reflexion:

Insights distilled from your past decisions:

1. Balance growth momentum with recent price trends and volatility to avoid overexposure to short-term weakness or reversals in high-volatility stocks like MSFT and UVV.
 2. Emphasize stocks with strong fundamentals and positive catalysts (e.g., HON, JNJ, MSFT) while maintaining modest cash to manage downside risk and market uncertainty.
 3. Maintain diversification across all stocks to capture unexpected rebounds, especially in volatile or uncertain markets, avoiding concentrated bets in recent winners.
- [... at most 8 insights]

Your reflections on your most recent decisions (oldest first):

- <reflection on the third most recent decision>
- <reflection on the second most recent decision>
- <reflection on the most recent decision>

G.2 ClassAlloc

System message (all calls).

You are a portfolio manager. You allocate a long-only portfolio across four asset classes (equities, bonds, commodities, real estate) and cash. At the start of each month you set target weights that are held for the next 20 trading days.

Decision prompt, here for 1 April 2022 (Appendix E) and the draft with all four retrieved lessons.

Date: 2022-04-01

Market data (each asset class and cash is an equal-weighted index of exchange-traded assets; returns in %; volatility = standard deviation of daily returns over the last 20 trading days, in %):
equities: 1-month +3.20; 3-month -5.35; 6-month +5.93; volatility 1.62
bonds: 1-month -2.06; 3-month -5.59; 6-month -5.23; volatility 0.45
commodities: 1-month -1.35; 3-month +21.55; 6-month +22.05; volatility 2.03
real estate: 1-month +2.17; 3-month -8.63; 6-month +2.81; volatility 1.38
cash: 1-month -0.05; 3-month -0.12; 6-month -0.16; volatility 0.01

Experiences from your past decisions (use them only where relevant):

[1] (2018-04-02) In April 2018, equities showed recent weakness but strong 6-month gains; bonds were stable with low volatility; commodities mildly positive; real estate mixed signals. I favored equities and bonds over cash, yielding a slightly better return than equal weights. In similar conditions, maintain diversified exposure but overweight assets with solid medium-term momentum and manageable volatility for balanced growth.

[2]-[4] <the lessons of August 2016, May 2017 and December 2015; Appendix E>

Decide the target weights for the next 20 trading days. Respond with a JSON object only, for example {"equities": 0.2, "bonds": 0.2, "commodities": 0.2, "real_estate": 0.2, "cash": 0.2, "reason": "one short sentence"}. Weights must be non-negative and sum to 1.

The retry message, the reflection prompt and the memory blocks are those of InvestorBench. In the lesson and reflection prompts, the outcome lines become

Your decision: equities <x>%, bonds <x>%, commodities <x>%, real estate <x>%, cash <x>%
Returns over the next 20 trading days (%): equities <+x.xx>, bonds <+x.xx>, commodities <+x.xx>, real estate <+x.xx>, cash <+x.xx>
Your portfolio return: <+x.xx>%; equal-weight portfolio (the five asset classes): <+x.xx>%.

and the insight prompt (ExpeL) becomes

Below are lessons you wrote after past monthly decisions, each with the outcome of that decision.

- [date]; portfolio <+x.xx>% vs equal weight <+x.xx>%] <lesson>
[... one line per warm-up month, 58 in all]

Compare the successful and unsuccessful decisions and extract at most 8 general insights for future monthly allocation across equities, bonds, commodities, real estate and cash. Return a numbered list; each insight at most 30 words.

Memorization probe (Appendix D.14), here for 1 April 2022; the system message is “You answer questions about financial markets.”

Date: 2022-04-01

Consider five asset classes, each an equal-weighted index of exchange-traded assets: equities, bonds, commodities, real estate and cash. From what you know about the markets, give the return in % of each asset class over the 20 trading days starting on this date. If you are not sure, give your best estimate. Respond with a JSON object only, for example {"equities": 1.5, "bonds": -0.3, "commodities": 2.0, "real_estate": 0.4, "cash": 0.1, "reason": "one short sentence"}.

G.3 PortBench

A draft is one run of the benchmark’s native agent in three stages: market interpretation, signal generation and weight optimization. Each stage is one chat completion that returns a JSON object (JSON mode, at most 8,192 output tokens) and receives the output of the previous stage; the memory enters all three stages as the benchmark’s experience block, which holds the records selected by the method. The prompts below are those of PortBench-Raw for the balanced investor on 1 April 2022, for the draft with all four retrieved records; PortBench-Full extends the market-data table of stage 1 from eight assets to all eligible assets.

System message (all stages).

You are the investment agent in a simulated research benchmark. Follow the native stage instruction and return its JSON object; historical experiences are observations, not guarantees.

Stage 1: market interpretation.

You are a portfolio manager analyzing market conditions on 2022-04-01.

MARKET DATA (trailing 60 trading days):

QQQ: price=352.83, trailing_return=-5.72%, ann_vol=31.03%
IWM: price=197.08, trailing_return=-4.32%, ann_vol=26.82%
VTI: price=215.83, trailing_return=-3.35%, ann_vol=22.60%
IVV: price=429.40, trailing_return=-2.98%, ann_vol=21.48%
EEM: price=41.63, trailing_return=-4.83%, ann_vol=28.28%
EFA: price=65.73, trailing_return=-5.89%, ann_vol=22.76%
VEA: price=42.58, trailing_return=-5.04%, ann_vol=21.35%
FXI: price=30.07, trailing_return=-5.79%, ann_vol=58.06%
Macro: fed_funds_rate=0.330, cpi_yoy=0.000, unemployment=3.900, gdp_growth_qoq=0.000, t10y2y_spread=0.040

MACRO INDICATORS:

Fed Funds Rate (%): 0.33
Unemployment (%): 3.90
10Y-2Y spread (pp): 0.04
[... 12 more indicators]

[... pairwise, intra-class and inter-class return correlations]

Current market regime context: unknown

RECENT NEWS / FILINGS:

[INVESTOR PROFILE] This investor is BALANCED: allocate at most 65% to equities and cryptocurrency combined; allocate at least 20% to bonds and cash; the portfolio may accept up to a 20% maximum drawdown.

[Kaggle 2022-03-31] ["Apple now allows Netflix, Spotify and other 'reader' apps to link to their sites for payment It's part of the tech giant's settlement with the Japan Fair Trade Commission.", [... further headlines of the previous day]

TASK: Interpret the market data and provide structured asset views.

Return a sparse set of material asset views. Any visible asset omitted from "asset_views" is deterministically interpreted as neutral (0.0). Only use visible asset identifiers and do not invent assets. You may include an explicit neutral 0.0 view, but do not need to list every neutral asset.

For each reported asset, assign a sentiment score in [-1.0, +1.0]:

+1.0 = strongly bullish (expect strong outperformance)
0.0 = neutral
-1.0 = strongly bearish (expect significant underperformance)

Identify the overall market regime: one of "bull", "bear", "sideways", "high-volatility".

JSON SCHEMA:

```
{
  "asset_views": { "ASSET": <float in [-1, 1]> },
  "detected_regime": "<bull|bear|sideways|high-volatility>",
  "confidence": <float in [0, 1]>,
  "macro_summary": "<one sentence>"
}
```

OUTPUT FORMAT – STRICT:

1. Reply with EXACTLY ONE JSON object and nothing else.
2. The very first character of your reply MUST be `{` and the very last character MUST be `}`. No leading whitespace, no trailing newline text.
3. Do NOT wrap the object in markdown code fences (no ```json, no ```).
4. Do NOT add any explanation, preamble, or commentary before/after.
5. Use double quotes for all keys and string values; numbers must be raw
JSON numbers (no quotes, no `%`, no commas).
6. Every key listed in the schema below MUST appear exactly once.

FROZEN HISTORICAL EXPERIENCE (observations, not proof of causal advantage):

```
{"text": "[{"id": "\ec8700d3d7c4...", "source_ids": [...], "text": "\<the record below, JSON-encoded>\"}, <3 more records>]", "unique_source_ids": [...]}
```

One record of the experience block, decoded. Each record describes an earlier decision of the benchmark’s agent (2015–2019): its allocation by asset class, a readout of its execution (net return J , fee, turnover, concentration HHI, largest weight and fraction of

assets held), public market features at the time, and the realized 20-day net return.

```
{
  "actual_target_weights_by_asset_class": {"bonds": 0.111, "cash": 0.011,
  "commodities": 0.1184, "cryptocurrency": 0.0074, "equities": 0.6856, "
  real_estate": 0.0666},
  "benchmark": "PortBench-Dynamic-99asset-native-shared-memory-raw-price-
  adaptation",
  "decision_date": "2016-05-02",
  "execution_readout_J_fee_turnover_HHI_maxweight_nonzero_fraction":
  [0.010039484803091359, 0.0010229302, 0.682, 0.029007079999999998, 0.0735,
  0.9883720930232558],
  "experience_id": "portbench/ec8700d3d7c4...:portbench:2016-05-02:r0",
  "feedback_scope": "actual20sessiontradeoutcome;
  unexecutedalternativesunknown; notcausalsuperiority",
  "public_state_features": [0.0, 0.86, 0.00153055, 0.01327625, 0.61627907,
  0.19767442],
  "realized_net_interval_return": 0.010039484803091359,
  "settled_date": "2016-05-31"
}
```

Stage 2: signal generation.

You are a portfolio manager on 2022-04-01.

Stage 1 market interpretation produced these asset views (scale: -1= bearish, +1=bullish):

```
QQQ: view=-0.400
IWM: view=+0.000
VTI: view=+0.000
[... the other assets]
```

Detected market regime: high-volatility

Macro summary: Market shows elevated volatility with flat CPI, low Fed funds rate, narrow yield curve, and moderate unemployment, indicating uncertain economic conditions and cautious sentiment.

Recent news / filings:

[investor profile and news, as in stage 1]

TASK: Convert each asset view into an actionable trading signal.

Rules:

- view > 0.15: consider "buy"
- view < -0.15: consider "sell"
- otherwise: consider "hold"

Use your judgement to refine signals based on regime and macro context. Signal strength should reflect conviction (0.0 = low, 1.0 = high).

Return only non-hold signals. Omitted visible assets are interpreted as hold with strength 0.5.

The signals and strengths objects must contain exactly the same selected assets.

JSON SCHEMA:

```
{
  "signals": {"SELECTED_ASSET": "<buy|sell>", ...},
  "strengths": {"SELECTED_ASSET": "<float in [0, 1]>", ...},
  "reasoning": "<one sentence>"
}
```

OUTPUT FORMAT - STRICT:

[the six rules of stage 1]

[experience block, as in stage 1]

Stage 3: weight optimization.

You are a portfolio manager on 2022-04-01.

Stage 2 signals:

```
QQQ: signal=sell, strength=0.60
IWM: signal=hold, strength=0.50
VTI: signal=hold, strength=0.50
[... the other assets]
```

Current portfolio weights: QQQ=0.0103, IWM=0.0103, VTI=0.0103, [...] (equal weights)

Portfolio NAV: \$1,000,000

Market regime: unknown

[return correlations, as in stage 1]

TASK: Select a sparse portfolio allocation based on the signals above.

Constraints:

- allocation_scores must be finite, non-negative decimal numbers, with at least one strictly positive score
- Do not normalize allocation_scores; the deterministic environment normalizes their relative magnitudes into target weights
- Omitted visible assets receive a target weight of 0.0; a listed 0.0 explicitly excludes that asset
- Prefer a concise allocation, while retaining any material existing holdings
- "sell" signals should receive reduced weight (ideally 0.0)
- "buy" signals should receive increased weight
- Minimize unnecessary turnover from current weights

JSON SCHEMA:

```
{
  "allocation_scores": {"SELECTED_ASSET": "<non-negative decimal>", ...},
  "expected_return": "<annualized decimal, e.g. 0.08>",
  "expected_vol": "<annualized decimal, e.g. 0.12>",
  "sharpe_estimate": "<decimal>"
}
```

OUTPUT FORMAT - STRICT:

[the six rules of stage 1]

[experience block, as in stage 1]

Replies of gpt-4.1-mini in this draft (seed 1). The environment turns the stage-3 scores into target weights, which are then projected to the investor's mandate (Appendix D.3). Here, stage 3 gives the same score to the 93 assets without a sell signal and none to the four with one.

Stage 1: {"asset_views": {"FXI": -0.7, "QQQ": -0.4, "EFA": -0.5, "EEM": -0.4}, "detected_regime": "high-volatility", "confidence": 0.8, "macro_summary": "Market shows elevated volatility with flat CPI, low Fed funds rate, narrow yield curve, and moderate unemployment, indicating uncertain economic conditions and cautious sentiment."}

Stage 2: {"signals": {"QQQ": "sell", "EEM": "sell", "EFA": "sell", "FXI": "sell"}, "strengths": {"QQQ": 0.6, "EEM": 0.6, "EFA": 0.7, "FXI": 0.9}, "reasoning": "In a high-volatility regime with cautious sentiment and bearish views on QQQ, EEM, EFA, and FXI, moderate to high conviction sell signals are warranted to reduce risk exposure in these assets."}

Stage 3: {"allocation_scores": {"IWM": 0.5, "VTI": 0.5, "IVV": 0.5, "VEA": 0.5, ... (89 more assets)}, "expected_return": 0.07, "expected_vol": 0.14, "sharpe_estimate": 0.5}

H Limitations

- *Gains from memory on real markets remain to be shown.* On PortBench and InvestorBench, MemTrial's advantage over the experience-learning agents comes from abstaining: its values passed the trust test on fewer than 1% of the test decisions, and with the same anchor the baselines do as well (Appendix D.5). On ClassAlloc they passed on 15%, and the trusted drafts did not beat the anchored blend they replaced. This shows that MemTrial avoids the losses of acting on untested memory, not that memory improves real decisions; showing the latter may need more decisions, more informative experiences or tasks in which experiences matter more. When its values are trusted, MemTrial deploys the selected draft in full and so takes on its market exposure (Appendix E); keeping part of the anchor even then is a natural extension.

- *Pretraining exposure (look-ahead bias).* The knowledge cutoff of gpt-4.1-mini (June 2024) is later than the whole InvestorBench test period and than 15 of the 20 PortBench test months, so the LLM may have seen these market outcomes during pretraining; the same holds for 50 of the 71 ClassAlloc test months. On ClassAlloc, the memory-free LLM beats $1/N$ by 0.08 pp per month before the cutoff and trails it by 0.08 pp after it (Appendix D.11), although 21 months are too few for this difference to be significant. Every LLM recalls dated outcomes from before its cutoff, and hiding the dates on ClassAlloc did not remove this pattern (Appendix D.14), but the LLM may still recognize a period from the returns it is shown; we did not hide the dates on PortBench and InvestorBench. This affects all LLM-based methods alike and leaves the paired comparisons fair, and PlantedMem, whose drafts do not depend on the date, is free of it; it may still change how much any LLM agent gains from memory.
- *Error rate of the trust test.* The trust test is applied on every date to dependent scores of a selected learner (Appendix C.3), so its false-pass rate has no formal bound; we measure it instead (2–3% of decisions without signal). A sequential test with time-uniform error control on block or permutation scores would bound it.
- *Cost.* MemTrial needs eight drafts per decision instead of one. The drafts run in parallel, so latency does not grow, but the LLM cost does; adaptive designs that spend more drafts only on dates where contributions are uncertain could reduce it.
- *Benchmark protocols.* PortBench scores each monthly decision as a separate episode that starts from the equal-weight book, so its costs are charged on the distance from that book rather than on actual turnover, and its experience pool is fixed before the first decision date, so learning happens in the credit rather than in the content of memory. Our InvestorBench task is a multi-asset adaptation of the benchmark’s single-stock task, with four stocks and cash. ClassAlloc’s class indices are equal-weighted and rebalanced daily, and its cryptocurrency class is left out. PlantedMem is semi-synthetic: its experiences act on the drafts through a simple tilt model (Appendix D.3), which is cleaner than the way real lessons act on an LLM, and each period’s outcome is known before the next decision.