UNIVERSE-AS-CODE: EXECUTABLE ASSET SELECTION FOR PORTFOLIO REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Portfolio reinforcement learning depends on both the assets available to a policy and the reward used to train it. We propose UCAPS (Universe-as-Code Agent Portfolio System), which learns executable upstream design artefacts for these coupled decisions. Universe-as-Code (UaC) generates multiple asset-selection programs, validates them in a sandbox, and combines their outputs through deterministic per-ticker voting. Dual-Artefact Co-Evolution (DACE) evaluates universe–reward pairs through downstream policy training and uses successful outputs to update separate Selector and Coder adapters. Its Composite Robust Score (CoRS) aggregates multi-seed feedback on historical performance, drawdown, temporal dispersion, and program complexity. On a 30-ticker multi-asset pool, UCAPS achieves 44.1% mean Test cumulative return and 1.31 Sharpe across four Selector backbones and 80 policy runs, net of transaction costs. The mean return exceeds the strongest reported same-pool large language model (LLM) agent adaptation by 10.5 percentage points. Price-matched classical baselines supply a common data reference, and eight matched selection controls isolate the selection rule. Under a fixed reference reward, the Qwen2.5-7B and Llama-3.1-8B Selectors exceed the strongest alternative by 20.6 and 11.1 percentage points. These results establish executable universe selection as an effective upstream optimization target for portfolio reinforcement learning.

1 INTRODUCTION

Portfolio construction allocates capital across financial assets over time (Markowitz, 1952; Liu et al., 2020). Before a reinforcement-learning (RL) policy learns allocation weights, its designer chooses an investable universe and a reward function. The universe determines which opportunities the policy can access; the reward determines which portfolio behaviors training favors. Both choices depend on market characteristics, and their interaction shapes the downstream learning problem.

Large language models (LLMs) can automate parts of this design process. Reward-code generators produce executable objectives and refine them using simulation feedback (Ma et al., 2024; Xie et al., 2024; Anonymous, 2025). Financial agents generate factors, trading decisions, or allocations (Tang et al., 2025; Li et al., 2025b; Zhang et al., 2024; Yu et al., 2024; Xiao et al., 2024). These approaches motivate a further upstream target: learning the executable mapping from a candidate pool's market statistics to the subset on which a portfolio policy will train.

Two issues determine how to learn that mapping. First, useful selection logic must combine cross-asset signals with constraints on the selected subset and remain executable when the input statistics change. Second, the utility of a reward depends on the universe on which its policy trains. Feedback should therefore evaluate complete universe–reward pairs and improve the generators responsible for both artefacts. Updating these generators also makes successful designs available to later iterations through their parameters.

We introduce **UCAPS**, organized around three mechanisms (Figure 1; Table 1 situates them among existing systems). **Universe-as-Code (UaC)** represents selection through generated programs, sandbox validation, and deterministic aggregation of their proposed subsets. **Dual-Artefact Co-Evolution (DACE)** trains portfolio policies for universe–reward pairs and converts successful outputs into separate supervised updates of the Selector and Coder adapters. **Composite Robust Score (CoRS)** ranks pairs using multi-seed feedback combining historical Sharpe, drawdown, chronological

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Representative comparison of classical, reward-code, and LLM-augmented trading systems. "Universe as code" denotes executable subset-selection logic; "generator update" denotes supervised updating of the LLM code generator from downstream evaluation; and robust pair ranking denotes universe–reward evaluation beyond a single fitness metric. The table characterizes the mechanisms used in this paper rather than reproducing published performance numbers.

Method	Universe as code	Generator update	Robust pair ranking	Reward as code	Universe form
Mean–Variance (Markowitz, 1952)	×	×	×	×	Static
FinRL (Liu et al., 2020)	×	×	×	×	Static
Eureka (Ma et al., 2024)	×	×	×	✓	Static
Multi-agent reward evolution (Anonymous, 2025)	×	×	×	✓	Static
Text2Reward (Xie et al., 2024)	×	×	×	✓	Static
Language-to-Rewards (Yu et al., 2023a)	×	×	×	✓	Static
Auto MC-Reward (Li et al., 2024)	×	×	×	✓	Static
AlphaAgent (Tang et al., 2025)	×	×	×	×	Static
RD-Agent (Li et al., 2025b)	×	×	×	×	Static
FinAgent (Zhang et al., 2024)	×	×	×	×	Static
FinCon (Yu et al., 2024)	×	×	×	×	Static
FinMem (Yu et al., 2023b)	×	×	×	×	Static
TradingAgents (Xiao et al., 2024)	×	×	×	×	Static
DeepFund (Li et al., 2025a)	×	×	×	×	Static
UCAPS (ours)	✓	✓	✓	✓	Code

dispersion, and code length. The final Selector generates a deployment universe at the calibration boundary, and fresh policies train with the retained reward for that universe.

Experiments distinguish system performance from selection-mechanism evidence. On the shared 30-ticker pool, UCAPS reaches 44.1% mean Test cumulative return and 1.31 Sharpe over four Selector backbones, with a strongest configuration of 48.8%. The mean return is 10.5 percentage points (pp) above the strongest same-pool financial-agent adaptation. Nine classical and rule-based baselines additionally share the main price field. Eight matched selection controls hold the pool, statistics, selection budget, reward, and proximal policy optimization (PPO) trainer fixed; they isolate the contribution of the selection rule. Component ablations examine both adapter updates and CoRS, while cross-pool and re-selection studies characterize how the learned rule depends on its deployment context.

2 Related Work

Portfolio construction and selection. Mean–variance allocation, covariance shrinkage, Bayesian estimation, and factor models establish statistical approaches to portfolio design (Markowitz, 1952; Ledoit & Wolf, 2004; Jorion, 1986; Fama & French, 1993). Hierarchical risk allocation and learned subset portfolios address diversification and asset selection (López de Prado, 2016; Lee, 2020). Deep hedging and FinRL learn sequential decisions for specified instruments (Buehler et al., 2019; Liu et al., 2020). UCAPS targets the selection program itself and evaluates its induced universe jointly with an executable training reward.

Reward and program generation. Eureka, Text2Reward, Language-to-Rewards, Auto MC-Reward, and a multi-agent reward-evolution workflow generate or refine reward code through environment feedback (Ma et al., 2024; Xie et al., 2024; Yu et al., 2023a; Li et al., 2024; Anonymous, 2025). More broadly, FunSearch optimizes programs using evaluators; Voyager accumulates executable skills; Code as Policies and ProgPrompt generate control or planning programs; and Program of Thoughts delegates computation to code (Romera-Paredes et al., 2024; Wang et al., 2024; Liang et al., 2023; Singh et al., 2023; Chen et al., 2023). UCAPS combines executable universe selection with pair-conditioned portfolio evaluation and separate parameter updates of two artefact generators.

LLM financial agents. AlphaAgent searches for alpha-factor code, and R&D-Agent-Quant jointly optimizes factors and predictive models (Tang et al., 2025; Li et al., 2025b). FinAgent, FinCon, and FinMem use tools, communication, or memory to support trading decisions (Zhang et al., 2024; Yu et al., 2024; 2023b). TradingAgents and DeepFund organize investment decisions through specialized roles (Xiao et al., 2024; Li et al., 2025a); MetaGPT and AutoGen provide general multi-agent

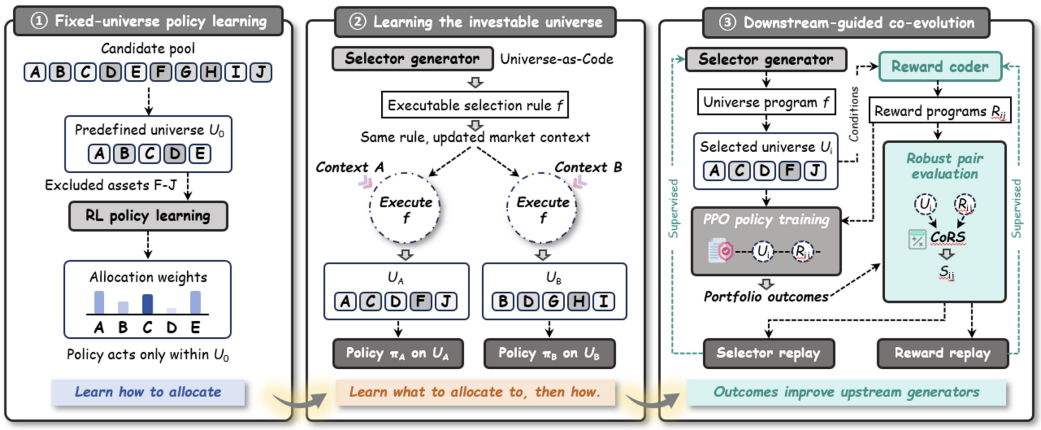

Figure 1: From fixed-universe policy learning to UCAPS. **(1) Fixed-universe policy learning:** a predefined universe limits the assets over which the RL policy learns to allocate. **(2) Learning the investable universe:** Universe-as-Code generates executable selection programs whose validated outputs are combined by deterministic per-ticker voting; the schematic rule f summarizes this aggregated procedure. Re-executing the learned rule under a different compatible market context can produce a different universe, with a separately trained policy for that universe. **(3) Downstream-guided co-evolution:** the selected universe conditions reward generation, PPO trains policies for universe–reward pairs, and CoRS ranks the pairs using historical portfolio outcomes. Eligible outputs populate separate replays for supervised Selector and Coder updates. In the reported deployment protocol, the final voted universe is selected at the calibration boundary and stays fixed throughout Test.

coordination mechanisms (Hong et al., 2024; Wu et al., 2024). Our optimization target is the executable selection/reward pair that configures downstream RL before deployment.

3 Method

3.1 Portfolio environment and information flow

Let $\mathcal{P}$ be a candidate pool and $\mathcal{U}$ an ordered tuple of K distinct tickers from $\mathcal{P}$, where K is the selection budget. Each selected ticker receives at most a per-asset cap u; the values of $|\mathcal{P}|$, K, and u are given in Section 4.1. At the close of day t, a policy observes portfolio and market features s_t, including previous target weights $\mathbf{w}_{t-1}$, and emits $a_t \in [-1, 1]^K$. Shifting, normalizing, and redistributing clipped weight produces a long-only allocation $\mathbf{w}_t$, with entries $w_{t,k}$, satisfying

$$0 \le w_{t,k} \le u, \qquad \mathbf{1}^\top \mathbf{w}_t \le 1. \tag{1}$$

The residual is implicit cash with zero modeled return. The transform targets exposure $\min\{1, Ku\}$ before downstream risk controls; Appendix C specifies the state and execution details. The next close-to-close return $\mathbf{r}_{t+1}$ and a cost on changes in target weights determine portfolio value V_t:

$$V_{t+1} = V_t\big(1 + \mathbf{w}_t^\top \mathbf{r}_{t+1}\big) - c\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1 V_t. \tag{2}$$

Here c is the transaction-cost rate (Section 4.1). All performance metrics use net return $q_{t+1} = (V_{t+1} - V_t)/V_t$. Generated reward code $R(s_t, a_t, s_{t+1})$ shapes training on this transition. The reference reward $R_0(s_t, a_t, s_{t+1}) = \log(1 + q_{t+1})$, the net log return, is held fixed in every comparison that isolates the selection procedure. For a fixed pair and seed s, proximal policy optimization (PPO) produces seed-dependent parameters

$$\theta^{(s)}(\mathcal{U}, R) = \mathrm{PPO}_s(\mathcal{U}, R; \mathrm{Train}), \tag{3}$$

under the fixed training schedule of Appendix A. A historical calibration window supplies search feedback and checkpoint selection. Test evaluates the finalized configurations.

3.2 UNIVERSE-AS-CODE

At calibration boundary t_{end}, the Selector receives seven statistics for each ticker a:

$$\boldsymbol{\sigma} : \mathcal{P} \to \mathbb{R}^7, \qquad \boldsymbol{\sigma}(a) = \big(\mu_L, \sigma_L, D_L, m_{60}, \rho_{\text{SPY},L}, \text{SR}_{\text{train}}, \text{SR}_L\big)(a). \tag{4}$$

Here μ_L and σ_L are annualized mean return and volatility; D_L is maximum-drawdown magnitude; m_{60} is 60-session momentum; and $\rho_{\text{SPY},L}$ is correlation with SPY. SR_{train} and SR_L are annualized Sharpe ratios over Train and over the trailing window of length L, respectively. The Selector prompt contains these records, pool and class identifiers, K, and the program interface.

At iteration i, the Selector generator G_ψ^{Sel}, an LLM with adapter parameters ψ, samples a batch of executable programs. Each program accepts $(\boldsymbol{\sigma}, \mathcal{P}, K)$ and returns a proposed subset. The sandbox permits `numpy` and `math`, disables file-system access, imposes a timeout, and checks that the returned subset contains exactly K unique pool members. Let $\mathcal{A}_i$ index valid programs and B_{im} their selections. The effective Selector Φ is

$$\begin{aligned} v_i(a) &= \sum_{m \in \mathcal{A}_i} \mathbf{1}\{a \in B_{im}\}, \\ \mathcal{U}_i &= \begin{cases} \text{OrderedTopK}_{a \in \mathcal{P}}\ v_i(a), & \mathcal{A}_i \neq \varnothing, \\ \phi_{\text{rule}}(\boldsymbol{\sigma}, \mathcal{P}, K), & \mathcal{A}_i = \varnothing. \end{cases} \end{aligned} \tag{5}$$

A fixed configuration-specific order resolves vote ties deterministically. The rule fallback ϕ_{rule} ranks by Sharpe with a per-class quota (Appendix A). Invalid programs supply neither votes nor replay code. Selection takes the K highest vote counts, and the resulting universe can differ from every individual program's subset. The generated programs apply a drawdown eligibility filter before scoring (Appendix D shows one program); Section 4.4 reports the effect of removing it.

The executable artefact consists of validated programs and their aggregation rule. It can be re-executed on compatible updated statistics without another LLM call. The deployment protocol executes a final ensemble once at the calibration boundary, fixes the ordered deployment universe $\mathcal{U}^{\text{dep}}$, and trains policies for it. The re-selection diagnostic trains a new policy whenever the universe is refreshed.

3.3 DUAL-ARTEFACT CO-EVOLUTION

DACE runs T iterations with five prompted roles sharing an LLM backbone. The Selector generates selection programs; the Strategist proposes a reward objective; the Architect specifies its components; the Coder implements them; and the Critic pre-screens candidates. Separate low-rank adapters update the Selector and Coder. For each $\mathcal{U}_i$, Critic-filtered outputs and mutations form N_R evaluated reward candidates R_{ij}. Each pair trains with a set of screening seeds $\mathcal{S}_{\text{scr}}$, yielding

$$J_{ij} = \text{median}_{s \in \mathcal{S}_{\text{scr}}}\, \text{CoRS}_s(\mathcal{U}_i, R_{ij}), \qquad j_i^* = \arg\max_j J_{ij}, \qquad R_i^* = R_{ij_i^*}. \tag{6}$$

CoRS is defined below. Within an iteration, rewards share a universe and evaluation windows; across iterations, the score compares complete observed pairs and maintains a search incumbent and a mutation pool.

Replay and ensemble-level credit. Write $h_{ij} = \text{median}_s\, \text{SR}_{\text{cal}}^{(s)}(\mathcal{U}_i, R_{ij})$, the median calibration Sharpe of a screened pair. Coder admissions require h_{ij} to exceed an iteration-dependent threshold and satisfy a per-iteration name-prefix cap (Appendix A). Its accumulated prompt–reward replay retains a bounded number of examples, ordered by recorded calibration Sharpe. Selector admission additionally requires the iteration winner's calibration Sharpe to reach that of the preceding highest-CoRS incumbent, within numerical tolerance. Eligible Selector supervision consists of the retained valid program texts and their generation contexts. The winning pair evaluates the voted universe, so its feedback is shared at the ensemble level; each replay target remains the actual generated program. Algorithm 1 gives the replay bookkeeping.

For role $\ell \in \{\text{Sel}, \text{Coder}\}$, admitted prompt–code pairs form $\mathcal{Q}_\ell$. Eligible updates with non-empty replay minimize token-level supervised fine-tuning (SFT) loss:

$$\mathcal{L}_\ell(\psi) = -\frac{1}{|\mathcal{Q}_\ell|} \sum_{(x,y) \in \mathcal{Q}_\ell} \sum_{\kappa=1}^{|y|} \log G_\psi^\ell(y_\kappa \mid x, y_{<\kappa}). \tag{7}$$

An adapter is carried forward when its update condition is unmet.

Search and deployment. Three objects are distinct: the search incumbent $(\mathcal{U}^{\text{search}}, R^*)$, the retained reward R^*, and the deployment universe $\mathcal{U}^{\text{dep}}$. The search incumbent $(\mathcal{U}^{\text{search}}, R^*)$ maximizes Eq. (6) over evaluated pairs. A final Selector sampling pass independently determines $\mathcal{U}^{\text{dep}}$. Fresh PPO policies train on $(\mathcal{U}^{\text{dep}}, R^*)$, and each run's checkpoint is selected by calibration Sharpe. These policies supply the Test results. This distinction keeps the historical winning pair, retained reward, and deployed universe explicit throughout Algorithm 1.

3.4 Composite Robust Score

For the policy trained with seed s, CoRS combines Train and calibration Sharpe with drawdown, temporal dispersion, and code complexity:

$$\begin{aligned}\text{CoRS}_s(\mathcal{U}, R) = \lambda_1 \, \text{SR}^{(s)}_{\text{train}}(\mathcal{U}, R) + \lambda_2 \, \text{SR}^{(s)}_{\text{cal}}(\mathcal{U}, R) \\ - \lambda_3 \, D^{(s)}(\mathcal{U}, R) - \lambda_4 \, \sigma^{(s)}_{\text{folds}}(\mathcal{U}, R) - \lambda_5 \, \text{LOC}(R).\end{aligned} \tag{8}$$

Here $D^{(s)} \geq 0$ is the calibration-window rollout's maximum-drawdown magnitude in fractional units. The dispersion term is the standard deviation of Sharpe over chronological calibration folds (Appendix A), and LOC, the number of lines of code, counts non-empty reward-source lines after whitespace normalization. The coefficients $\lambda_1, \ldots, \lambda_5$ are fixed across configurations (Section 4.1). Equation (6) aggregates the single-seed scores by the median; replay admission uses the separately defined calibration statistic h_{ij}. The resulting feedback couples historical portfolio performance to updates of both code generators.

4 Experiments

4.1 Setup

Data and protocol. The fixed candidate pool contains 30 tickers: 11 equities, 18 exchange-traded funds or trusts, and one BTC/USD series, grouped into six economic exposures. UCAPS uses archived Financial Modeling Prep (FMP) `close` prices. Train spans 2020-01-01–2024-08-31; calibration spans 2024-09-01–2024-12-31; and Test spans 2025-01-02–2026-03-31 (311 common price bars, 310 returns). Available histories are aligned on common trading dates. Selector features, DACE feedback, retained rewards, and policy checkpoints use historical data through the calibration boundary. The main Test universe is fixed. Appendix A provides the ticker list, pre-Test public documentation, and information-flow details.

Metrics and replication. Cumulative return (CR) is the primary metric; annualized Sharpe supplies the risk-adjusted view. Both use Eq. (2), zero risk-free return, and a 252-session annualization convention. Signed MaxDD (maximum drawdown) reports the negative drawdown percentage. The pooled result combines 20 PPO runs for each of four Selector backbones under the same co-evolved reward: 80 runs in total. Means and standard deviations summarize these run outcomes; Appendix A gives the complete table and uncertainty estimates.

Comparisons and implementation. Table 2 contains 27 baselines. Nine classical/rule-based strategies, including simple-moving-average (SMA), moving-average-convergence/divergence (MACD), and relative-strength-index (RSI) rules, share the 30-ticker pool and `close` prices. Same-pool LLM-agent adaptations and RL baselines retain their specified decision rules; market-reference portfolios appear separately. Appendix E records price conventions, rebalancing schedules, and preprocessing. Eight matched selection controls additionally fix the statistics, $K = 5$, the reference reward R_0, PPO training, and evaluation protocol. The main Selector is Qwen2.5-7B-Instruct with rank-16 LoRA; the alternative backbones use the sampling recipe in Appendix A. Main deployment PPO training totals 30,000 steps per run.

Configuration constants. The main configuration uses $|\mathcal{P}| = 30$, $K = 5$, $u = 0.30$, $c = 3$ bps (basis points), a nine-month Selector window L ending at the calibration boundary, $T = 5$ DACE iterations with $N_R = 12$ evaluated rewards and 3 screening runs per iteration, and CoRS coefficients $(\lambda_1, \ldots, \lambda_5) = (0.5, 0.5, 0.5, 0.5, 0.02)$ over four calendar-month calibration folds.

Table 2: Test performance (2025-01–2026-03), net of a 3 bps turnover cost. UCAPS pools 80 runs; multi-run baselines report means. Classical/rule-based rows share the 30-ticker pool and `close` data. Agent adaptations use the same pool with the method-specific price and decision conventions in Appendix E. ‡: dividend-adjusted pricing.

Category	Method	Test CR (%)	Sharpe
Classical (30 assets)	Min-Variance	9.5	1.30
	Mean/vol weighting	14.8	0.55
	Inverse volatility	18.6	0.99
	Equal-Weight	17.0	0.70
Rule-based (30 assets)	Cross-sectional momentum (top decile)	21.4	0.56
	MACD timing, equal-weight	23.4	1.06
	SMA(200) trend, equal-weight	19.1	0.83
	RSI mean-reversion (bottom decile)	−14.6	−0.09
	RSI filter, equal-weight	16.4	0.68
Market references	Equal-weight sectors	14.4	0.72
	SPY buy-and-hold	11.2	0.56
	60/40 SPY/TLT	6.3	0.46
RL	Full-pool PPO (30 assets)	17.1	0.70
	Random universe (50 draws)	16.0	0.66
	FinRL-PPO (Liu et al., 2020)	5.5	0.32
	FinRL-A2C	−7.0	−0.17
	FinRL-TD3	−12.9	−0.45
	FinRL-SAC	−14.9	−0.64
	FinRL-DDPG	−16.9	−0.66
LLM-based	TradingAgents‡ (Xiao et al., 2024)	33.6	0.89
	FinAgent (Zhang et al., 2024)	32.0	0.89
	FinCon (Yu et al., 2024)	22.9	0.85
	FinMem (Yu et al., 2023b)	13.7	0.46
	DeepFund (Li et al., 2025a)	17.0	0.58
	Eureka (Ma et al., 2024)	17.5	0.71
	Multi-agent reward evolution (Anonymous, 2025)	17.5	0.71
	Zero-shot LLM advisor	7.3	0.34
Ours	UCAPS (four Selector backbones)	**44.1 ± 5.1**	**1.31 ± 0.10**

Replay thresholds and capacities, the sandbox, tie-break, fallback, PPO, and adapter settings, and the recorded update timing are in Appendix A.

4.2 System performance and cross-backbone results

UCAPS achieves $44.1 \pm 5.1\%$ Test CR and 1.31 ± 0.10 Sharpe (Table 2). TradingAgents, the highest-return same-pool agent adaptation, reaches $33.6 \pm 10.5\%$ CR and 0.89 Sharpe. The difference in mean CR is 10.5 pp. Among the nine price-matched classical/rule-based strategies, MACD reaches the highest CR, 23.4%, giving UCAPS a 20.7 pp advantage; Equal-Weight reaches 17.0%. Figure 2 shows their Test trajectories.

The four Selector configurations attain mean returns from 40.8% to 48.8% under the shared reward (Table 3). Qwen2.5-7B achieves $48.8 \pm 5.1\%$, with a 95% seed-mean interval of approximately ± 2.4 pp. Llama-3.1-8B gives the highest mean Sharpe, 1.39. The pooled-mean backbone-block bootstrap interval is $[41.5, 47.4]\%$. Separate runs pairing each backbone's universe with its backbone-specific co-evolved reward reach 45.5% for Mistral-7B-v0.3 and 41.8% for Llama-3.1-8B (Appendix B).

4.3 Matched selection-mechanism comparison

Table 4 replaces the Selector while fixing the input statistics, candidate pool, $K = 5$, reward, and downstream PPO protocol. Fixed rankers obtain 13.3–18.4% CR. The symbolic-regression scorer is the strongest alternative at $26.6 \pm 3.4\%$, while the ridge, boosted-tree, and multilayer-perceptron (MLP) scorers range from -17.3% to 1.4%. The Qwen2.5-7B Selector reaches $47.2 \pm 4.3\%$ and

Table 3: Selector-backbone configurations under the shared co-evolved reward and PPO stack; 20 runs per row. The pooled row combines all 80 runs. MaxDD is the mean signed maximum drawdown.

Backbone	Test CR (%)	Sharpe	MaxDD (%)
Qwen2.5-7B	48.8 ± 5.1	1.30 ± 0.10	-23.5
Phi-3-medium-14B	40.8 ± 4.6	1.26 ± 0.09	-21.7
Mistral-7B-v0.3	43.6 ± 4.2	1.30 ± 0.09	-22.2
Llama-3.1-8B	43.2 ± 3.2	1.39 ± 0.08	-20.4
Aggregate	44.1 ± 5.1	1.31 ± 0.10	-22.0

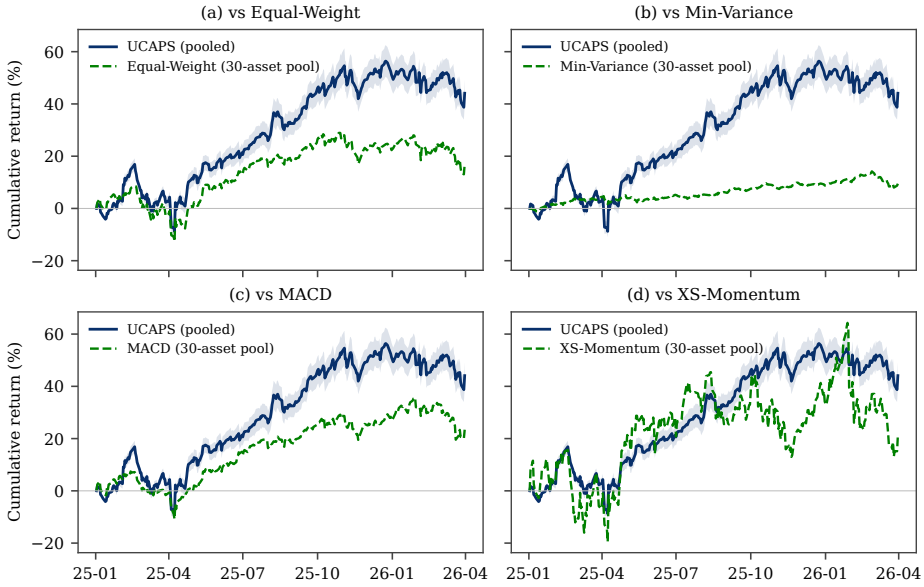

Figure 2: Test cumulative returns on the same 30-ticker pool and `close` prices. UCAPS uses the pooled mean equity path with a one-standard-deviation band. Baseline decision schedules are monthly; UCAPS acts daily. Each panel compares the same UCAPS series with one baseline.

the Llama-3.1-8B Selector $37.7 \pm 3.9\%$, improving on the symbolic scorer by 20.6 and 11.1 pp, respectively. Thus informed selection retains its advantage with the reward held fixed. Min-MaxDD ranking trades lower return for Sharpe 1.34; its drawdown and all training definitions appear in Appendix B.

4.4 Component ablations and sensitivity

Learning both artefacts. The main configuration (Qwen2.5-7B with the co-evolved reward) exceeds Full-pool PPO by 31.7 pp (Table 5). Uniform random $K = 5$ universes average 16.0%, close to Full-pool PPO. Starting from the Selector without DACE updates (25.9%), the two recorded Selector updates yield 39.4% and 44.1% under R_0. Removing the Selector update reduces the main configuration's CR by 18.6 pp; removing the Coder update reduces it by 6.9 pp. Replacing CoRS with calibration-Sharpe argmax ranking reduces CR by 5.4 pp. These interventions change the search trajectories; fixed-universe reward substitutions are reported separately in Appendix B.

Composition and selection budget. Removing the programs' drawdown filter lowers CR to 6.2% and deepens MaxDD to -33.7%. A 15-ticker subpool reaches 19.2%; a three-month statistics window reaches 28.2%. Direct LLM voting over ticker lists reaches $37.5 \pm 1.6\%$. Budget changes yield 34.8% at $K = 3$, 35.1% at $K = 8$, and 30.1% at $K = 10$. At $K = 3$, the cap sets all asset weights to 0.30, leaving 10% cash and making the resulting allocation policy-independent. The corresponding cells are included in Table 5; Appendix B gives protocol details and additional diagnostics.

Table 4: Matched selection comparison: identical seven statistics, 30-ticker pool, $K = 5$, reference reward R_0, and PPO/evaluation protocol; 20 runs per row. Selector rows here use R_0, whereas Table 3 uses the co-evolved reward.

Selection mechanism	Test CR (%)	Sharpe
Rule fallback ϕ_{rule}	18.4 ± 2.0	0.73 ± 0.07
Min-MaxDD ranking	18.0 ± 1.6	1.34 ± 0.09
Momentum ranking	15.5 ± 3.9	0.48 ± 0.06
Top-Sharpe ranking	13.3 ± 2.9	0.45 ± 0.05
Symbolic-regression scorer	26.6 ± 3.4	0.79 ± 0.07
Ridge scorer	1.4 ± 3.7	0.26 ± 0.06
Boosted-tree scorer	-9.7 ± 2.8	0.06 ± 0.05
MLP scorer	-17.3 ± 1.7	-0.05 ± 0.04
UaC: Qwen2.5-7B	$\mathbf{47.2 \pm 4.3}$	1.27
UaC: Llama-3.1-8B	37.7 ± 3.9	1.19

Table 5: Ablations and deployment sensitivity on the main Test period. The main configuration uses 20 PPO runs; most ablation and cadence cells use 3. The random-universe control aggregates 50 draws. Composition, direct-output, budget, and cadence rows use the reference reward R_0. The iteration-2 CR of 44.1% is a 3-run cell under R_0.

Configuration	Test CR (%)	Sharpe
System and DACE controls		
UCAPS main configuration (Qwen2.5-7B)	48.8 ± 5.1	1.30 ± 0.10
Full-pool PPO (30 assets)	17.1 ± 1.7	0.70 ± 0.06
Random universe (50 draws)	16.0 ± 15.0	0.66 ± 0.46
Selector after iteration 1; R_0	39.4 ± 3.8	1.04 ± 0.08
Selector after iteration 2; R_0	44.1 ± 2.6	1.05 ± 0.04
Without DACE updates	25.9 ± 3.0	0.78 ± 0.06
Without Selector updates	30.2 ± 5.4	0.84 ± 0.10
Without Coder updates	41.9 ± 0.6	1.22 ± 0.04
Without CoRS	43.4 ± 1.0	1.25 ± 0.03
Composition and selection budget		
Single random universe	-0.9 ± 3.6	0.20 ± 0.07
Without drawdown filter	6.2 ± 3.6	0.32 ± 0.07
Pool-15	19.2 ± 3.1	0.60 ± 0.06
Three-month lookback	28.2 ± 3.1	0.87 ± 0.06
Direct LLM universe-list vote	37.5 ± 1.6	0.89 ± 0.04
$K = 3$	34.8 ± 0.0	0.96 ± 0.00
$K = 8$	35.1 ± 2.7	1.21 ± 0.06
$K = 10$	30.1 ± 2.4	1.14 ± 0.06
Re-selection cadence		
Deployment protocol (no re-selection)	47.2 ± 3.4	1.28 ± 0.07
Annual	52.5 ± 3.7	1.28 ± 0.05
Semiannual	33.3 ± 0.4	0.79 ± 0.07
Quarterly	39.9 ± 2.6	1.17 ± 0.06
Monthly	25.2 ± 0.4	0.76 ± 0.07

Re-execution and deployment context. Table 5 reports the cadence sweep under R_0: the deployment protocol reaches 47.2%, while annual, semiannual, quarterly, and monthly re-selection reach 52.5%, 33.3%, 39.9%, and 25.2%. A historical replay over 2023–2024 orders the intervals differently. A separate frozen-program, equal-weight cross-pool diagnostic (Table 6, panel a) gives a 21.8 pp top-five selection gain on the equity pool and -2.6 pp on the ETF pool. A trailing-volatility overlay provides a lower-risk execution profile: 30.4% CR, 1.33 Sharpe, and -11.5% MaxDD (Appendix B).

Robustness summary. The frozen-program transfer study isolates the selection rule from PPO and shows a positive top-five gap on the stock-bearing pool and the original heterogeneous pool, while the homogeneous ETF pool is more budget-sensitive. On the main Test path, the Selector under

Table 6: Deployment-context diagnostics using existing evaluation records. Panel (a) applies one frozen Selector program to equal-weight buy-and-hold pools and reports the top-five CR gap relative to the full pool. Panel (b) slices the Selector (R_0) and Full-pool PPO paths at the same Test closes.

(a) Cross-pool ranking transfer

Pool	Full CR (%)	Top-5 gap (pp)
US equities	6.3%	+21.8 pp
Cross-asset ETFs	25.1%	−2.6 pp
Original pool	15.4%	+11.4 pp

(b) Temporal Test slices

Period	Selector	Full-pool	SPY
2025-01–09	45.6	24.3	14.0
2025-10–26-03	0.65	−6.85	−2.7

R_0 retains a positive return in the 2025-10–2026-03 stress slice while Full-pool PPO and SPY are negative. Appendix D provides the complete budget sweep, correlations, and interval definitions.

Attribution across the two generated artefacts. The controlled results distinguish learning the investable universe from reward refinement. Random $K = 5$ selection remains near Full-pool PPO (16.0% versus 17.1%), while the matched learned Selectors reach 37.7–47.2% under the reference reward R_0 and the eight alternative selection mechanisms remain at or below 26.6%. Direct ticker-list voting reaches 37.5%. Once the Qwen2.5-7B universe is fixed, three handwritten rewards span 47.0–48.5% CR and the co-evolved reward reaches 48.8% (Appendix B). Thus the largest observed lift is associated with learned universe selection; reward-side co-evolution adds a smaller final fixed-universe increment while still shaping search, as freezing the Coder update and replacing CoRS reduce the main configuration's CR to 41.9% and 43.4%, respectively (Table 5).

Search and deployment profile. With the Selector and PPO trainer fixed, six Coder backbones run four five-iteration DACE chains each; their mean best search Sharpe spans only 2.412–2.454 under common re-scoring (Appendix B). Deployment then removes the LLM from the decision path: the retained Selector is executable Python and allocation uses the trained PPO policy. A no-lookahead 20-day volatility overlay provides a lower-risk operating point, reducing the Qwen2.5-7B (R_0) MaxDD from -23.8% to -11.5% while yielding 30.4% CR and 1.33 Sharpe.

Deployment composition. Across the four pooled Selector backbones, every deployment universe contains AAPL, GLD, and PLTR; the remaining two positions vary among NVDA, SPY, QQQ, XLU, and XLY (Appendix A). This common core is paired with backbone-specific broad-market or sector exposure rather than a single fixed universe. Together with the matched selection controls and cross-pool diagnostic, the observed compositions show that the learned executable rule changes which assets enter the downstream policy while preserving the same portfolio-learning stack.

5 Conclusion

UCAPS learns executable universe-selection and reward artefacts through pair-conditioned portfolio evaluation. UaC combines validated selection programs through deterministic voting; DACE updates their Selector and Coder generators; and CoRS aggregates multi-seed historical performance and robustness terms. On the 30-ticker pool, UCAPS achieves 44.1% mean Test CR and 1.31 Sharpe, with a strongest configuration of 48.8%. The mean CR exceeds the strongest same-pool agent adaptation by 10.5 pp. Price-matched baselines, matched selection controls, and component ablations support the contribution of executable universe selection and coupled generator updates. Deployment executes the retained programs and trained policy without further LLM calls.

Reproducibility Statement

Appendix A specifies the pools, data fields, historical splits, generation settings, training schedules, and run-level outcomes. Appendix C gives the complete search and deployment procedure; Appendix E specifies the baseline inputs and execution conventions. Appendix B reports the matched controls, reward substitutions, backbone configurations, cadence protocols, and sensitivity series; Appendix D reports the cross-pool and temporal analyses; Appendix G reports the reproduction cost of the deployed configuration.

REFERENCES

Anonymous. Forging better rewards: A multi-agent LLM framework for automated reward evolution. OpenReview, 2025. URL https://openreview.net/forum?id=Z6GStCfccl. ICLR 2026 submission, withdrawn.

Bitwise. Bitwise crypto equity indexes: December 2024 rebalance results, 2024. URL https://bitwiseinvestments.com/indexes/rebalance-results/bitwise-crypto-equity-indexes/december-2024. Crypto Innovators 30 Index weights published December 30, 2024.

Hans Buehler, Lukas Gonon, Josef Teichmann, and Ben Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019. doi: 10.1080/14697688.2019.1571683.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=YfZ4ZPt8zd.

CoinMarketCap. Historical snapshot: 29 december 2024, 2024. URL https://coinmarketcap.com/historical/20241229/.

Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56, 1993. doi: 10.1016/0304-405X(93)90023-5.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations (ICLR) 2024*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Philippe Jorion. Bayes-Stein estimation for portfolio analysis. *Journal of Financial and Quantitative Analysis*, 21(3):279–292, 1986. doi: 10.2307/2331042.

J.P. Morgan. 2024 global ETF handbook. Technical report, Global Equity Derivatives Strategy, J.P. Morgan, 2024. URL https://d1e00ek4ebabms.cloudfront.net/production/uploaded-files/JPM%202024%20Global%20ETF%20Handbook_PRINT-917c16b1-af18-4578-b747-f9e738601818.pdf. 20th annual edition; definitions p. 5, listings pp. 27–28, 36–40, 46, and 51.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004. doi: 10.1016/S0047-259X(03)00096-4.

Sang Il Lee. Deeply equal-weighted subset portfolios, 2020. URL https://arxiv.org/abs/2006.14402.

Changlun Li, Yao Shi, Yuyu Luo, and Nan Tang. Will LLMs be professional at fund investment? DeepFund: A live arena perspective. *arXiv preprint arXiv:2503.18313*, 2025a.

Hao Li, Xue Yang, Zhaokai Wang, Xizhou Zhu, Jie Zhou, Yu Qiao, Xiaogang Wang, Hongsheng Li, Lewei Lu, and Jifeng Dai. Auto MC-Reward: Automated dense reward design with large language models for minecraft. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yuante Li, Xu Yang, Xiao Yang, Minrui Xu, Xisen Wang, Weiqing Liu, and Jiang Bian. R&D-Agent-Quant: A multi-agent framework for data-centric factors and model joint optimization. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025b.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA) 2023*, pp. 9493–9500, 2023. doi: 10.1109/ICRA48891.2023.10160591.

Listed Funds Trust. Roundhill magnificent seven ETF: Schedule of investments, september 30, 2024, 2024. URL https://www.sec.gov/Archives/edgar/data/1683471/000114554924075939/mags2.htm. Form N-PORT exhibit, filed November 27, 2024.

Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bo Xiao, and Christina Dan Wang. FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. In *Deep Reinforcement Learning Workshop, Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Marcos López de Prado. Building diversified portfolios that outperform out-of-sample. *The Journal of Portfolio Management*, 42(4):59–69, 2016. doi: 10.3905/jpm.2016.42.4.059.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=IEduRUO55F.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. doi: 10.2307/2975974.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: Generating situated robot task plans using large language models. In *IEEE International Conference on Robotics and Automation (ICRA) 2023*, 2023. doi: 10.1109/ICRA48891.2023.10161317.

S&P Dow Jones Indices. 2024 press release archive, 2024. URL https://press.spglobal.com/index.php?s=2429&year=2024&l=100. Index membership announcements.

Ziyi Tang, Zechuan Chen, Jiarui Yang, Jiayao Mai, Yongsen Zheng, Keze Wang, Jinrui Chen, and Liang Lin. AlphaAgent: LLM-driven alpha mining with regularized exploration to counteract alpha decay. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '25)*, pp. 2813–2822. ACM, 2025. doi: 10.1145/3711896.3736838.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *First Conference on Language Modeling (COLM)*, 2024.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. TradingAgents: Multi-agents LLM financial trading framework. *arXiv preprint arXiv:2412.20138*, 2024.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2Reward: Reward shaping with language models for reinforcement learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024. URL https://openreview.net/forum?id=tUM39YTRxH.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning (CoRL)*, 2023a.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W. Suchow, and Khaldoun Khashanah. FinMem: A performance-enhanced LLM trading agent with layered memory and character design. *arXiv preprint arXiv:2311.13743*, 2023b.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan W. Suchow, Zhenyu Cui, Rong Liu, Zhaozhuo Xu, Denghui Zhang, Koduvayur Subbalakshmi, Guojun Xiong, Yueru He, Jimin Huang, Dong Li, and Qianqian Xie. FinCon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f7ae4fe91d96f50abc2211f09b6a7e49-Abstract-Conference.html.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. 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 (KDD '24)*, pp. 4314–4325. ACM, 2024. doi: 10.1145/3637528.3671801.

AI USE STATEMENT

Generative AI assisted with manuscript editing, cross-version reconciliation, notation and algorithm clarification, reference checks, and typesetting. Numerical checks during revision recomputed summaries from the supplied experimental records; no new experimental outcomes were generated. LLM-generated selection and reward programs are research components described in Section 3. The authors are responsible for the scientific content and the submission.

A Reproducibility Details

A.1 Data, candidate pool, and information flow

Data field and alignment. The main UCAPS environment uses the FMP `close` field in the archived offline snapshot. Dividend-adjusted `adjClose` is a separate price convention used in the baseline robustness comparison of Appendix E. Available constituent histories are aligned on the intersection of their trading dates, without imputation or forward filling in the UCAPS environment. Train bounds are 2020-01-01–2024-08-31; later-listed instruments contribute their available history within those bounds. For the Test pool, the shared calendar is the New York Stock Exchange (NYSE) calendar: 311 price bars and 310 returns over 2025-01-02–2026-03-31. BTC/USD has 454 bars during that span; its 143 weekend and holiday bars are excluded from the common calendar, and the intervening price move enters the next common close-to-close return. A date missing for a universe member is removed for that universe; all main Test series cover the 311 dates. Classical-estimator preprocessing is specified separately in Appendix E.

Fixed candidate pool. The pool contains the 30 tickers in Table 7. Six economic groups are used in selection: individual equities (NVDA, PLTR, SMCI, GOOGL, META, TSLA, AAPL, AMZN, MSFT); sector funds (XLK, XLU, XLP, XLV, XLE, XLF, XLY, XLI); broad-market funds (SPY, QQQ, DIA, IWM, EFA, EEM); precious metals (GLD, SLV); Treasuries (TLT, IEF); and crypto-linked instruments (MSTR, BTCUSD, COIN). MSTR and COIN are equities, giving 11 equities, 18 exchange-traded funds/trusts, and one crypto price series in total. Table 7 documents constituents using public records whose observation dates precede Test. All main UCAPS configurations and pool-matched controls receive this same ticker list.

Table 7: The fixed 30-ticker pool and its pre-Test public documentation. Group counts sum to 30. Documentation groups identify sources; the six selection classes are defined in the text.

Documentation group	Tickers	Pre-Test reference
Large technology (7)	AAPL, AMZN, GOOGL, META, MSFT, NVDA, TSLA	MAGS holdings dated 2024-09-30, filed 2024-11-27; GOOGL is Alphabet Class A (Listed Funds Trust, 2024).
S&P 500 entrants (2)	SMCI, PLTR	2024 additions; the pool uses the two leading ordinary additions by 2024 median daily close-times-volume (S&P Dow Jones Indices, 2024).
Crypto equities (2)	MSTR, COIN	The two largest Bitwise Crypto Innovators 30 weights on 2024-12-30 (9.58%, 9.44%) (Bitwise, 2024).
Sector funds (8)	XLE, XLF, XLI, XLK, XLP, XLU, XLV, XLY	H-rated members of the Select Sector SPDR family in the 2024 ETF handbook (J.P. Morgan, 2024).
Precious metals (2)	GLD, SLV	Passive, unleveraged, H-rated precious-metal instruments (J.P. Morgan, 2024).
Treasuries (2)	IEF, TLT	Passive, unleveraged, H-rated Treasury instruments (J.P. Morgan, 2024).
Broad/region funds (6)	SPY, QQQ, DIA, IWM, EFA, EEM	Passive, unleveraged, H-rated broad-market funds; highest six-month cash ADV among funds with the same constituents (J.P. Morgan, 2024).
Cryptoasset (1)	BTCUSD	Largest cryptoasset by capitalization in the 2024-12-29 weekly snapshot; FMP's USD price series (CoinMarketCap, 2024).

H denotes three-month average daily options notional above USD 100 million. Handbook observations are as of end-May 2024; cash average daily volume (ADV) uses six months.

Feature and evaluation timing. The main Selector feature table is computed at 2024-12-31. Its trailing statistics use 2024-04-01–2024-12-31, while $\mathrm{SR}_{\mathrm{train}}$ uses Train. DACE trains screening policies on Train under the deployment schedule and scores them on the fixed Train/calibration windows. Calibration supplies reward retention, replay admission, and checkpoint selection. Final universe generation uses that historical feature table; the deployed universe and reward remain fixed during the main Test evaluation. Matched learned scorers use forward 63-session targets whose windows finish before Test. The earlier-period cadence study is a retrospective replay; the separate

Test cadence study re-executes the frozen rules and retrains policies by segment. Neither diagnostic changes the deployment protocol.

Metric conventions. For a sequence of N net daily simple returns, $\mathrm{CR} = 100(\prod_{t=1}^{N}(1+q_t)-1)$ and $\mathrm{SR} = \sqrt{252}\,\bar{q}/s_q$, with sample standard deviation s_q and zero risk-free return. Portfolio drawdown magnitude is $D = \max_t(1 - V_t/\max_{\tau\le t} V_\tau)$; tables report signed $\mathrm{MaxDD} = -100D$. Selector return means and volatilities use the same 252-session annualization convention. Sixty-session momentum is the endpoint price ratio minus one, and SPY correlation uses aligned trailing returns. CoRS receives drawdown magnitudes in fractional units; the frozen-program diagnostic in Appendix D explicitly converts them to percentage units.

A.2 Training, generation, and replication

PPO and adapters. PPO uses learning rate 3×10^{-4}, batch size 256, and 1024 environment steps per epoch. Main deployment training has three stages of 10,000 steps and chooses the checkpoint by calibration Sharpe. The main LoRA recipe has rank 16, scaling $\alpha = 32$, SFT learning rate 3×10^{-5}, and QLoRA 4-bit NF4 quantization. Generation uses temperature 0.6 and top-$p = 0.95$. The alternative Selector recipe uses a 120-program Selector replay, rank 32, and three SFT epochs. The four main backbone releases precede the Test window: Qwen2.5-7B (2024-09-19), Phi-3-medium-14B (2024-05-21), Mistral-7B-v0.3 (2024-05-22), and Llama-3.1-8B (2024-07-23).

Table 8: Selector sampling settings and resulting deployment universes. Samples are generated once per final Selector evaluation; only valid outputs vote. Universes are displayed alphabetically, while each policy keeps its fixed coordinate order.

Backbone	Rank	Temp.	Samples	Validated	Deployment universe
Qwen2.5-7B	16	0.6	10	7	AAPL, GLD, NVDA, PLTR, SPY
Phi-3-medium-14B	32	0.8	30	10	AAPL, GLD, PLTR, SPY, XLY
Mistral-7B-v0.3	32	0.8	30	30	AAPL, GLD, PLTR, QQQ, SPY
Llama-3.1-8B	32	0.8	30	20	AAPL, GLD, PLTR, XLU, XLY

Voting and retention. Vote ties in Eq. (5) are ranked by $\mathrm{SR}_{\mathrm{train}} + \mathrm{SR}_L$ and then by ticker symbol in all four configurations; the resulting order fixes the returned tuple. The sandbox permits `numpy` and `math`, disables file-system access, and imposes a 5-second timeout. Programs failing parsing, execution, timeout, or output checks are discarded. The rule fallback is invoked when the valid ensemble is empty. For the main pool and evaluated budgets, $\sum_c \min\{|\mathcal{P}_c|, \lceil K/4 \rceil\} \ge K$, where $\mathcal{P}_c$ is a selection class, so the fallback quota admits a size-K result. Selector replay contains retained valid prompt–program pairs associated with a successful ensemble, and each program inherits the ensemble's downstream admission decision.

DACE schedule. Search runs for five iterations with 12 evaluated rewards and 3 screening PPO runs per iteration. The Critic keeps the top 40% of its candidate inputs; mutations supply the remaining evaluated candidates. The mutation fraction increases from 0.40 to 0.75 at iteration five. CoRS ranks observed pairs for the incumbent and mutation pool. Replay uses a separate quality gate: calibration Sharpe exceeds $\beta_i = 0.60 + 0.01i$, followed by a per-iteration name-prefix cap of two admitted entries per prefix. Accumulated Coder examples are truncated to $M = 60$ by recorded calibration Sharpe, and the Coder adapter is updated after every iteration that admits a new example. Selector admission compares the qualifying winner with the calibration Sharpe of the highest-CoRS incumbent from preceding iterations. The recorded Selector updates occur in iterations one and two. An ineligible update leaves the corresponding adapter unchanged.

CoRS diagnostics. The Train and calibration windows remain fixed. Four calendar-month folds cover September–December 2024, with no shuffling. Fold-failure screening precedes ranking. The five coefficients in Eq. (8) are $(\lambda_1, \ldots, \lambda_5) = (0.5, 0.5, 0.5, 0.5, 0.02)$ and are shared across configurations. In an instrumented candidate set, varying each coefficient across five settings preserves the retained candidate in 22 of 25 cases.

Hardware and software. Experiments use single NVIDIA RTX 3090, RTX 4090, or A6000 devices (24–48 GB). The software stack is Ubuntu 22.04, CUDA 12.4, `torch==2.6.0+cu124`, `vllm==0.8.5`, `transformers==4.46.3`, `peft==0.13.0`, `trl==0.11.4`, and `bitsandbytes`. Cross-Coder chains additionally use `sentencepiece`, `tiktoken`, `einops`, and `protobuf`; SFT runs with `HF_HUB_OFFLINE=1`. Seed labels identify runs within a configuration.

Replication units. The four deployment configurations, Full-pool PPO, and the matched selection controls use 20 PPO runs per cell. Three-run component, composition, budget, and Test-cadence cells use 3; Direct LLM Universe Selection reports the first three of 20 runs in seed order. The random-universe control aggregates 50 draws. Abbreviated sensitivity sweeps use 10 runs. Other repeated configurations use 10 unless their captions specify otherwise. Each Coder diagnostic has four search chains, each using 3-run median re-scoring. The four deployment universes are fixed within their 20-run cells (Table 9); the pooled minimum Test CR is 34.3%. The main configuration's seed-mean 95% interval has half-width approximately 2.4 pp, and the reported backbone-block bootstrap interval for the pooled mean is $[41.5, 47.4]\%$.

A.3 Run-level values

The anonymous implementation reference is `https://anonymous.4open.science/r/ucaps-anon-iclr2027-4D62`.

Table 9: Run-level Test CR (percent) and Sharpe for the 80 pooled runs. The bottom row gives each cell's mean CR, sample standard deviation, and mean Sharpe.

	Qwen2.5-7B			Phi-3-medium-14B			Mistral-7B-v0.3			Llama-3.1-8B	
Run	CR	SR	Run	CR	SR	Run	CR	SR	Run	CR	SR
1	44.3	1.24	1	37.1	1.23	1	38.5	1.24	1	38.7	1.39
2	50.2	1.33	2	34.3	1.17	2	37.4	1.26	2	40.1	1.45
3	55.6	1.45	3	36.1	1.18	3	40.9	1.22	3	41.5	1.31
4	51.6	1.36	4	45.3	1.25	4	47.3	1.28	4	46.5	1.41
5	48.9	1.34	5	44.4	1.36	5	50.6	1.43	5	50.3	1.42
6	44.2	1.28	6	42.0	1.35	6	39.5	1.25	6	41.7	1.36
7	53.7	1.26	7	37.7	1.22	7	49.6	1.47	7	43.4	1.40
8	44.3	1.16	8	40.1	1.23	8	46.9	1.36	8	42.5	1.30
9	44.6	1.25	9	37.4	1.17	9	39.6	1.20	9	50.4	1.51
10	51.2	1.41	10	49.0	1.47	10	47.6	1.45	10	42.6	1.54
11	40.1	1.12	11	38.2	1.17	11	42.2	1.24	11	44.7	1.41
12	52.3	1.50	12	40.1	1.30	12	45.6	1.38	12	42.4	1.47
13	49.5	1.32	13	41.1	1.30	13	45.7	1.33	13	46.0	1.50
14	52.8	1.23	14	44.5	1.36	14	47.2	1.32	14	40.3	1.32
15	49.0	1.26	15	37.3	1.19	15	42.7	1.32	15	44.9	1.33
16	58.0	1.44	16	39.2	1.23	16	38.6	1.17	16	40.4	1.34
17	39.6	1.16	17	51.5	1.40	17	48.9	1.38	17	42.2	1.33
18	45.1	1.20	18	45.4	1.26	18	41.8	1.25	18	42.7	1.29
19	46.0	1.26	19	36.0	1.13	19	39.6	1.21	19	39.9	1.27
20	54.4	1.39	20	40.0	1.27	20	41.1	1.32	20	43.5	1.45
Mean	48.8 ± 5.1	1.30	Mean	40.8 ± 4.6	1.26	Mean	43.6 ± 4.2	1.30	Mean	43.2 ± 3.2	1.39

B Extended Results and Sensitivity

B.1 Matched controls, reward substitutions, and backbone configurations

Matched Selector training. The eight matched selection controls in Table 4 use the same seven-statistic table, 30-ticker pool, $K = 5$, reference reward R_0, and PPO trainer. Fixed rankers use historical statistics at the calibration date. Learned scorers fit rolling monthly cross-sections, with month-end features and forward 63-session Sharpe targets; all feature and target windows finish before Test. Symbolic regression runs twelve restarts and selects the program with the highest mean cross-sectional rank information coefficient on a 2023-10–2024-09 selection window. The ridge,

boosted-tree, and MLP scorers produce crypto-heavy selections with MaxDD between -37.3% and -45.2%. Fixed-ranker MaxDD ranges from -8.5% for Min-MaxDD to -31.9% for Top-Sharpe. The Qwen2.5-7B Selector under R_0 has MaxDD -23.8%; the Llama-3.1-8B Selector has -22.1%. Each matched cell uses 20 policy runs.

Fixed-universe reward substitutions. Table 10 evaluates three handwritten rewards on the selected Qwen2.5-7B universe. Their CR range is 1.5 pp; the co-evolved reward has the highest mean, 48.8%. These substitutions keep the deployment universe fixed. The Coder-update ablation in Table 5 instead changes reward generation, pair screening, and the feedback available during DACE.

Table 10: Reward substitutions on the fixed Qwen2.5-7B deployment universe under the main PPO protocol; 20 runs per cell. Program identifiers name the corresponding reward implementations.

Reward	Test CR (%)
`risk_adjusted_momentum`	47.4 ± 4.5
`simple_return`	48.5 ± 3.9
`leverage_aware`	47.0 ± 3.5
Co-evolved reward	48.8 ± 5.1

Reward-code diagnostics. The full loop's selected reward contains three branches and a momentum–concentration interaction. Without DACE updates, the Coder's chosen program optimizes drawdown and volatility without a return term, and its four candidates contain two distinct structures. Replacing CoRS with calibration-Sharpe ranking yields seven coefficient variants of one structure in the final pool. A separate 5,000-step, 20-run-per-cell diagnostic places ten generated and handwritten reward variants within a 0.9 pp CR band around 46%.

Own-reward deployment. Table 11 pairs each backbone's universe with the reward selected by its own DACE chains and uses the main deployment PPO protocol. Relative to the shared-reward configuration means, the CR changes are 0.0, $+1.9$, and -1.4 pp for Qwen2.5-7B, Mistral-7B-v0.3, and Llama-3.1-8B. The Qwen2.5-7B row is the main configuration; its minimum run CR is 39.6%.

Table 11: Backbone-specific reward configurations R_b^*; 20 PPO runs per cell. Each reward is retained by its historical search evaluation.

Backbone	Test CR (%)	Sharpe
Qwen2.5-7B	48.8 ± 5.1	1.30
Mistral-7B-v0.3	45.5 ± 3.3	1.34
Llama-3.1-8B	41.8 ± 3.3	1.45

Coder-backbone search. The Coder study fixes the Qwen2.5-7B Selector and PPO trainer and runs four five-iteration DACE chains for each of six backbones. At each iteration, the top three rewards are rescored on a common evaluation market, using the median of three PPO runs. Table 12 reports the best reward-search Sharpe in each chain. Means lie between 2.412 and 2.454 on the search-diagnostic scale, which differs from the Test Sharpe scale of Table 3.

B.2 Composition, budget, and re-selection

Configuration definitions. Composition and budget cells (Table 13) use R_0. The single-random-universe cell is one specified universe across runs; the random-universe control in Table 5 aggregates 50 draws. The drawdown-filter ablation removes the programs' eligibility filter. Pool-15 uses the recorded 15-ticker composition-ablation pool; the lookback variant uses the preceding three months. Direct LLM Universe Selection prompts base Qwen2.5-7B-Instruct for ticker lists, aggregates them by voting, and then trains PPO with R_0.

Exposure and drawdown. At $K = 3$, $Ku = 0.9$, so the cap-and-redistribute transform sets the three weights to 0.30 and leaves 0.10 in implicit cash. The allocation is consequently constant across

Table 12: Coder-backbone search diagnostic: four independent chains per backbone, five iterations per chain, and 3-run median re-scoring of each iteration's top three rewards. Columns 1–4 contain chain-level best search Sharpe.

Coder backbone	1	2	3	4	Mean ± std
Qwen2.5-Coder-7B	2.333	2.427	2.472	2.472	2.426 ± 0.066
Llama-3.1-8B	2.415	2.457	2.472	2.472	2.454 ± 0.027
Yi-1.5-9B	2.414	2.393	2.427	2.415	2.412 ± 0.014
Mistral-7B-v0.3	2.422	2.411	2.422	2.419	2.419 ± 0.005
DeepSeek-Coder-7B	2.419	2.402	2.419	2.419	2.415 ± 0.008
Qwen3-8B	2.472	2.390	2.472	2.472	2.452 ± 0.041

Table 13: Additional ablations on the main Test period; 3 PPO runs per cell. Composition, direct-output, and budget rows use R_0. The $K = 3$ row invests 90% in assets and 10% in cash under the common 0.30 cap.

Axis	Configuration	Test CR (%)	Sharpe
Composition	Single random universe	-0.9 ± 3.6	0.20 ± 0.07
	Without MDD filter	6.2 ± 3.6	0.32 ± 0.07
	Pool-15	19.2 ± 3.1	0.60 ± 0.06
	Three-month lookback	28.2 ± 3.1	0.87 ± 0.06
Direct output	LLM universe-list vote	37.5 ± 1.6	0.89 ± 0.04
Budget	$K = 3$	34.8 ± 0.0	0.96 ± 0.00
	$K = 8$	35.1 ± 2.7	1.21 ± 0.06
	$K = 10$	30.1 ± 2.4	1.14 ± 0.06

policies and its cross-seed standard deviation rounds to zero. The drawdown-filter removal has mean signed MaxDD -33.7%; the single-random-universe cell has -40.3%.

Cadence protocol. Each schedule in Table 14 covers the complete Test window. It re-executes the same seven frozen selection programs and vote at its specified boundaries, with per-segment PPO retraining and R_0. Realized observations enter the statistics available at subsequent boundaries. The deployment protocol makes no re-selection; annual re-selection makes one switch within this Test window. Universe changes unwind dropped positions through the standard rebalance; the one-off target-weight turnover is at most 2 and its cost at most $2c$.

Table 14: Re-selection cadence on Test: 3 PPO runs per setting, reference reward R_0, same selection programs and voting rule. The deployment-protocol row is this study's 3-run reference; Table 4 reports the 20-run estimate under R_0.

Cadence	Test CR (%)	Sharpe
Deployment protocol (no re-selection)	47.2 ± 3.4	1.28 ± 0.07
Annual	52.5 ± 3.7	1.28 ± 0.05
Semiannual	33.3 ± 0.4	0.79 ± 0.07
Quarterly	39.9 ± 2.6	1.17 ± 0.06
Monthly	25.2 ± 0.4	0.76 ± 0.07

Historical cadence replay. Table 15 replays the same programs, aggregation, reward, and per-segment retraining over 2023-01–2024-12, with 20 runs and 501 trading days. Monthly re-selection has the highest return in that history; annual has the highest return on Test.

B.3 Risk controls and abbreviated sensitivity

The Qwen2.5-7B universe with R_0 has reported Sortino 1.82 and Sharpe 1.27. A trailing 20-day realized-volatility overlay scales exposure toward a 15% annualized target using information

Table 15: Retrospective cadence replay over 2023–2024; 20 runs. The last column reproduces the separate Test-cadence means from Table 14.

Cadence	Historical segments	2023–2024 CR (%)	Sharpe	Test CR (%)
Deployment protocol	1	23.8 ± 2.5	1.10	47.2
Annual	2	69.1 ± 5.5	1.48	52.5
Semiannual	4	86.0 ± 6.3	1.51	33.3
Quarterly	8	110.5 ± 9.5	1.77	39.9
Monthly	24	124.4 ± 9.9	1.91	25.2

available at each decision. This execution profile yields 30.4% CR, 1.33 Sharpe, and -11.5% MaxDD, compared with -23.8% MaxDD without that overlay. Released weight remains as cash.

The sensitivity series in Table 16 uses 10 runs per cell. Its default training length is 20,000 PPO steps, compared with 30,000 for main deployment. Each axis is a separate series with its own reference cell.

Table 16: Abbreviated sensitivity series for the Qwen2.5-7B universe and the reference reward R_0; 10 runs per cell. Cap-row Sharpe values are shown in parentheses.

Axis	Setting	Test CR (%)
Transaction cost	0 bps	41.9 ± 3.3
	3 bps	43.8 ± 3.4
	10 bps	43.5 ± 2.7
Per-asset cap	0.30 (Sharpe 1.37)	40.6 ± 3.0
	0.50 (Sharpe 1.40)	40.1 ± 3.0
	0.70 (Sharpe 1.38)	39.6 ± 2.9
	0.99 (Sharpe 1.38)	39.9 ± 3.0
PPO steps	5,000	42.3 ± 2.4
	20,000	43.8 ± 6.0

C Algorithm, State, and Notation

Table 17 summarizes the notation used in the method and in Algorithm 1.

State and action coordinates. Each policy uses one ordered universe throughout training and deployment. Its state is a fixed-order vector containing sixteen portfolio/market scalars: current portfolio value and running peak; current and running-maximum drawdown; trailing 20-day mean and volatility of realized portfolio returns; net and gross exposure; previous-step turnover; long, short, and total position value; five-day return momentum; the equal-weight benchmark return; the last step return; and episode progress. It also contains the previous K target weights, current K prices, previous K raw actions, and the last ten portfolio and benchmark returns.

Cap-and-redistribute transform. Raw actions are shifted by $(a_t + 1)/2$ and normalized to a tentative long-only vector $\widetilde{\mathbf{w}}_t$. The transform targets total exposure $g_t = \min\{\mathbf{1}^\top \widetilde{\mathbf{w}}_t, Ku\}$, clips weights at u, and redistributes clipped mass in proportion to remaining capacity below the cap. The executed vector satisfies $0 \le w_{t,k} \le u$ and $\mathbf{1}^\top \mathbf{w}_t \le 1$, up to floating-point precision. The residual earns zero modeled cash return. Downstream risk controls may further reduce exposure. With $K = 3$ and $u = 0.30$, the allocation is $(0.30, 0.30, 0.30)$ before any additional exposure reduction.

Transition and costs. Target weights chosen at close t are held for the following close-to-close return. Transaction cost is $c\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_1$ as a fraction of current portfolio value, using consecutive target vectors. A generated reward sees the same transition as the environment. This target-weight convention defines the reported net CR and Sharpe. A universe change uses the union of old and new ticker coordinates to compute its rebalance cost; non-negative weights with total exposure at most one imply turnover at most two.

Algorithm 1 DACE search and fixed-universe deployment

Require: Pool $\mathcal{P}$, budget K, boundary t_{end}, iterations T, candidates per iteration N_R, screening seeds $\mathcal{S}_{\mathrm{scr}}$, Coder replay capacity M (values in Section 4.1 and Appendix A)
Ensure: Deployment universe $\mathcal{U}^{\mathrm{dep}}$, retained reward R^*, independent deployment policies

$\boldsymbol{\sigma} \leftarrow \mathrm{ComputeStats}(\mathcal{P}, t_{\mathrm{end}})$ using Eq. (4)
Initialize Selector/Coder adapters, empty replays and mutation pool
Initialize search incumbent $I \leftarrow \varnothing$, incumbent score $s^* \leftarrow -\infty$
for $i = 1, \ldots, T$ **do**
    Sample F_i from the Selector; validate samples; retain valid outputs
    $\mathcal{U}_i \leftarrow \Phi(F_i, \boldsymbol{\sigma}, \mathcal{P}, K)$ by Eq. (5), with the configuration's deterministic tie order
    Strategist/Architect specify rewards conditioned on $\mathcal{U}_i$
    Coder generates programs; Critic keeps top 40% of its inputs
    Complete $\{R_{ij}\}_{j=1}^{N_R}$ with score-pool mutations
    for each candidate R_{ij} **do**
        for $s \in \mathcal{S}_{\mathrm{scr}}$ **do**
            Train $\theta_{ij}^{(s)} \leftarrow \mathrm{PPO}_s(\mathcal{U}_i, R_{ij}; \mathrm{Train}, \mathcal{H})$
            Obtain historical Sharpe, drawdown, monthly dispersion, and CoRS_s
        end for
        $J_{ij} \leftarrow \mathrm{median}_s\, \mathrm{CoRS}_s(\mathcal{U}_i, R_{ij})$
        $h_{ij} \leftarrow \mathrm{median}_s\, \mathrm{SR}_{\mathrm{cal}}^{(s)}(\mathcal{U}_i, R_{ij})$
    end for
    $j_i^* \leftarrow \arg\max_j J_{ij}$; $R_i^* \leftarrow R_{ij_i^*}$
    Set Selector eligibility by quality and the *preceding* incumbent's calibration Sharpe
    Update score-ranked mutation pool with evaluated pairs
    Admit Coder examples with $h_{ij} > \beta_i$ (β_i in Appendix A), subject to the prefix cap
    Keep top M accumulated Coder examples by recorded calibration Sharpe
    Update the Coder by SFT if this iteration admitted a new example; otherwise carry the adapter forward
    if Selector eligibility holds and retained valid programs are available **then**
        Add retained prompt–program examples from F_i to Selector replay
        Update Selector by SFT using Eq. (7)
    else
        Carry the Selector adapter forward
    end if
    if $J_{ij_i^*} > s^*$ **then**
        $I \leftarrow (\mathcal{U}_i, R_i^*, h_{ij_i^*})$; $s^* \leftarrow J_{ij_i^*}$
    end if
end for
Set $(\mathcal{U}^{\mathrm{search}}, R^*)$ from the retained incumbent I
Sample the final Selector ensemble; obtain $\mathcal{U}^{\mathrm{dep}}$ by Eq. (5)
for each independent deployment seed **do**
    Train fresh PPO on $(\mathcal{U}^{\mathrm{dep}}, R^*)$ using Train
    Select that run's checkpoint by calibration Sharpe
    Evaluate the fixed universe, retained reward, and selected policy on Test
end for
return $\mathcal{U}^{\mathrm{dep}}$, R^*, and the trained policies with their Test outcomes

Incumbent and replay ordering. Before the first incumbent exists, a valid best pair passing the calibration-quality gate establishes the initial reference. Later Selector admissions compare against the incumbent retained *before* the current iteration. Its calibration statistic is distinct from its CoRS ranking score. Coder quality gates and replay truncation use calibration Sharpe, while candidate/mutation ranking uses CoRS. These separate quantities preserve the retention protocol across universe changes.

Table 17: Notation shared by the method and Algorithm 1.

Symbol	Meaning
$\mathcal{P}$,$\mathcal{U}, K$	Candidate pool, ordered selected universe, and selection budget
$u, \mathbf{w}_t, c$	Per-asset cap, executed target weights, and cost per unit target turnover
$\boldsymbol{\sigma}, L$	Seven-statistic per-ticker record and trailing lookback window
$F_i, \mathcal{A}_i, B_{im}$	Sampled programs, valid-sample indices, and individual selections
$v_i, \Phi, \phi_{\text{rule}}$	Per-ticker votes, ensemble Selector with deterministic tie order, and rule fallback
ψ, G^{ℓ}_{ψ}	Adapter parameters and the LLM generator of role ℓ
$T, N_R, M, \mathcal{S}_{\text{scr}}$	Iterations, evaluated rewards per iteration, Coder replay capacity, and screening seeds
$\beta_i, \lambda_1, \ldots, \lambda_5$	Replay admission threshold (Appendix A) and CoRS coefficients (Section 4.1)
$R_{ij}, \theta^{(s)}_{ij}$	Candidate reward and policy trained under $(\mathcal{U}_i, R_{ij})$ with seed s
J_{ij}, h_{ij}	Median CoRS and median calibration Sharpe of a screened pair
$\mathcal{Q}_{\text{Sel}}, \mathcal{Q}_{\text{Coder}}$	Separate prompt–program supervised replays
$\mathcal{U}^{\text{search}}, R^*, \mathcal{U}^{\text{dep}}$	Search-incumbent universe, retained reward, and final deployment universe

D Cross-Pool and Temporal Analyses

D.1 Frozen-program cross-pool diagnostic

The study applies one validated program from the Qwen2.5-7B Selector (iteration 2, sample 0) to three pools. Its unit of evaluation is the generated scoring rule with equal-weight buy-and-hold allocation. The additional equity pool contains NVDA, MSFT, AAPL, GOOGL, META, AMZN, TSLA, PLTR, MSTR, and COIN. The ETF pool contains the 18 funds/trusts in the main pool: eight sector funds, six broad/region funds, GLD, SLV, IEF, and TLT. Train and nine-month statistics are computed at 2024-12-31 for each pool.

For price series P_t, the program uses $D_L = \max_t(1 - P_t/\max_{\tau \le t} P_\tau)$ and its percentage-unit representation $d_L = 100 D_L$. Tickers are eligible when $d_L \le 25$. Eligible tickers are ranked by the recorded executable score

$$\gamma(a) = 0.5\,\mathrm{SR}_L(a) + 0.5\,\mathrm{SR}_{\text{train}}(a)\left(1 + \frac{d_L(a)}{15}\right). \tag{9}$$

The eligibility gate controls maximum drawdown; within it, Eq. (9) multiplies the Train Sharpe by $1 + d_L/15$, favouring long-run performers that have recently drawn down. Selected top-K and full-pool portfolios are equally weighted at entry and held over Test, without PPO or a per-asset cap. Thus the full-pool 15.4% CR in Table 18 belongs to buy-and-hold, while the 17.0% Equal-Weight cell in Table 2 follows its target-weight execution schedule.

Table 18: Frozen-program cross-pool diagnostic. Entries under K are CR percentage-point gaps between top-K and full-pool equal-weight buy-and-hold. Spearman ρ_S compares frozen scores with Test returns over the n_ρ eligible tickers.

	Full-pool	Top-K minus full pool (pp)				
Pool	CR (%)	$K=3$	$K=5$	$K=8$	$K=10$	ρ (n_ρ)
US equities (10)	6.3	43.9	21.8	n/a	n/a	0.94 (6)
Cross-asset ETFs (18)	25.1	9.2	−2.6	10.9	6.9	0.09 (18)
Original pool (30)	15.4	7.6	11.4	12.0	5.9	−0.01 (25)

Six equity tickers pass the gate, so budgets above six are inapplicable. All comparisons in this table are deterministic.

The equity pool has $\rho_S = 0.94$ over its six eligible tickers and positive selection gains at both feasible budgets. The top-K selection beats the full pool for every feasible K on the equity and original pools and for three of four budgets on the ETF pool.

D.2 TEMPORAL RETURN PROFILES

Broad sub-periods. Table 19 slices the same three full-window paths from the Qwen2.5-7B (R_0) and Full-pool PPO configurations at the indicated closes. The first interval contains 185 returns and the second 124. In the latter interval, the mean difference between the learned-policy columns is 7.50 pp.

Table 19: CR (percent) over two specified close-to-close intervals. Learned-policy cells use 3 runs under R_0; SPY is buy-and-hold between the same closes.

Close boundaries	Selector (R_0)	Full-pool PPO	SPY
2025-01-02–2025-09-30	45.6 ± 4.2	24.3 ± 0.9	14.0
2025-10-01–2026-03-31	0.65 ± 0.32	-6.85 ± 1.11	-2.7

Illustrative aligned paths. Tables 20 and 21 partition individual trajectories into five non-overlapping 60-return-day intervals and the remaining ten days. UCAPS uses the Qwen2.5-7B co-evolved-reward trajectory nearest its cell's mean full-Test CR; Full-pool PPO uses the same representative-run rule. TradingAgents uses one run of its adaptation, and MACD and Equal-Weight are deterministic. Repeated-run performance is reported in Table 2. TradingAgents uses dividend-adjusted prices, while the other displayed paths use `close`; all include transaction costs. Each policy continues its original decision schedule across the display boundaries.

Interval CR is the endpoint net asset value (NAV) ratio minus one, and interval Sharpe is recomputed from its daily returns. The full-span Sharpe uses all 310 returns. UCAPS leads CR and Sharpe in three of the five 60-day intervals.

Table 20: Illustrative sub-period CR (percent), net of costs. Column paths, price conventions, and representative-run selection are specified above.

Close boundaries	UCAPS	TradingAgents	Full-pool PPO	MACD	Equal-Weight
2025-01-02–2025-04-01	2.34	−8.78	−1.69	−0.82	−1.75
2025-04-01–2025-06-27	18.38	25.41	14.79	15.25	15.82
2025-06-27–2025-09-23	21.43	10.47	8.74	8.68	8.96
2025-09-23–2025-12-17	4.39	−2.06	−1.59	1.13	−2.08
2025-12-17–2026-03-17	0.74	3.81	2.03	1.77	1.42
2026-03-17–2026-03-31	−3.78	−2.68	−4.95	−3.49	−5.00
Full Test span	48.86	25.06	17.13	23.40	16.97

Table 21: Annualized Sharpe for the same paths and intervals as Table 20. The full-span row is computed from all 310 returns.

Close boundaries	UCAPS	TradingAgents	Full-pool PPO	MACD	Equal-Weight
2025-01-02–2025-04-01	0.46	−0.77	−0.22	−0.10	−0.22
2025-04-01–2025-06-27	2.03	2.51	2.04	2.70	2.10
2025-06-27–2025-09-23	5.64	2.03	3.24	3.15	3.26
2025-09-23–2025-12-17	1.00	−0.15	−0.34	0.39	−0.45
2025-12-17–2026-03-17	0.25	1.05	0.56	0.55	0.42
2026-03-17–2026-03-31	−2.69	−3.37	−3.94	−2.59	−3.89
Full Test span	1.34	0.74	0.72	1.06	0.70

Rolling-window diagnostic. A complementary series takes thirteen 60-session windows at 20-session strides from the frozen Test trajectories. Adjacent windows share 40 returns. The Qwen2.5-7B universe with R_0 leads Full-pool PPO in 12 of the 13 windows, with a mean gap of 5.4 pp. Table 22 reports five non-overlapping intervals from the same recorded series.

Table 22: Five non-overlapping 60-return-day intervals from the same Test path. The first four aggregate the full evaluation set; the final interval uses the reported 3-run subset. Displayed means and gaps are separately rounded.

Close boundaries	Selector (R_0) CR (%)	Full-pool PPO CR (%)	Gap (pp)
2025-01-02–2025-04-01	0.9	−1.9	2.7
2025-04-01–2025-06-27	19.8	15.6	4.2
2025-06-27–2025-09-23	21.0	9.0	12.0
2025-09-23–2025-12-17	3.7	−1.9	5.6
2025-12-17–2026-03-17	2.3	1.8	0.4

D.3 Deployment interpretation

Every pooled deployment universe contains AAPL, GLD, and PLTR. The remaining positions vary among NVDA, SPY, QQQ, XLU, and XLY. The resulting configurations combine equity exposure with a precious-metal holding and, in several cases, broad-market or sector exposure. Matched controls connect the performance differences to how subsets are selected under a fixed downstream stack. Cross-pool and cadence results further show that composition, budget, and temporal context determine how those selection rules translate into portfolio outcomes.

E Baseline Configurations and Price Conventions

E.1 Classical and rule-based controls

All nine classical/rule-based baselines in Table 2 use the same 30-ticker pool and archived FMP `close` field as UCAPS. Test prices align to the common 311 dates (310 returns). A 3 bps cost applies to changes in target weights, including initial entry. Decision dates are the first Test close of each month; the selected target vector is retained between decisions, and daily returns and costs use consecutive target vectors. Each method retains its specified fully invested long-only allocation rule. The UCAPS cap and policy risk controls apply to the UCAPS policy.

Classical allocations. Equal-Weight assigns $1/30$ to every asset. Min-Variance estimates covariance from Train, applies shrinkage $0.7\Sigma + 0.3\,\mathrm{diag}(\Sigma)$, and uses a long-only active-set solution. Mean/vol weighting assigns normalized positive $\max(\widehat{\mu}_i/\widehat{\sigma}_i, 0)$ scores. Inverse volatility uses normalized inverse-volatility weights. These Train-estimated target vectors remain fixed during Test. Pre-listing Train returns unavailable to the classical estimators are set to zero; all 30 Test series are complete.

Rule-based allocations. SMA(200), MACD, and the RSI filter equally weight qualifying names and use full-pool equal weight when no name qualifies. MACD computes 12/26-period exponential moving averages on a trailing 61-close window. The RSI filter accepts $30 < \mathrm{RSI} < 70$. Cross-sectional momentum ranks trailing 252-session returns and holds the top decile, while RSI mean-reversion holds the bottom RSI decile. Both decile portfolios contain three equally weighted assets; both RSI rules use the preceding 15 price differences. Signals use information through the decision close and apply to subsequent close-to-close returns.

Market references. SPY buy-and-hold, the 60/40 SPY/TLT portfolio, and equal-weight sector ETFs retain their defining instrument sets and allocation conventions. Their separate table category distinguishes these contextual portfolios from the 30-ticker mechanism comparisons.

E.2 RL and LLM-agent implementations

RL controls. Full-pool PPO uses the reference reward R_0 and the UCAPS training stack on all 30 tickers. The random-universe control replaces the learned rule with uniform $K = 5$ draws and aggregates 50 universes. The additional FinRL suite (Liu et al., 2020) uses PPO, advantage actor–critic (A2C), twin-delayed deep deterministic policy gradient (TD3), soft actor–critic (SAC),

and deep deterministic policy gradient (DDPG) through Stable-Baselines3, with checkpoint selection on the historical calibration window.

Reward-code methods. The Eureka adaptation (Ma et al., 2024) generates reward code for full-pool PPO. The multi-agent reward-evolution adaptation (Anonymous, 2025) combines that loop with a meta-judge prefilter.

Financial-agent adaptations. The five allocator adaptations use GPT-4o-mini on the 30-ticker pool at temperature 0.7, with three generation-seed runs. DeepFund (Li et al., 2025a) uses an analyst-to-portfolio-manager allocation followed by buy-and-hold. FinAgent (Zhang et al., 2024) receives computed momentum, volatility, and RSI indicators in its prompt. TradingAgents (Xiao et al., 2024) uses bull/bear analysts and a portfolio-manager debate. FinMem (Yu et al., 2023b) maintains prior decisions and realized P&L in memory, and FinCon (Yu et al., 2024) routes a manager proposal through a risk-control agent. A zero-shot GPT-4o-mini advisor is included separately. Allocators rebalance monthly except DeepFund, which allocates once; UCAPS acts daily.

Shared conventions and price scope. Comparisons share the Test dates, a 3 bps unit-turnover cost, and daily-return Sharpe annualized by $\sqrt{252}$. Candidate and checkpoint choices use the historical calibration window. The nine classical/rule-based rows and the matched selection controls share UCAPS's `close` prices. Financial-agent rows retain their archived input conventions; the TradingAgents paths use dividend-adjusted prices, as specified in the aligned-path analysis. The 10.5 pp difference compares UCAPS and TradingAgents mean CR on the same candidate pool. Table 23 holds each classical/rule-based strategy fixed and varies its price field explicitly.

Table 23: Price-field comparison for all nine classical/rule-based strategies on the 30-ticker pool. Main `close` values reproduce Table 2; the alternative uses FMP `adjClose`. Rules, dates, and transaction-cost convention are unchanged.

	`close` (main)		Dividend-adjusted	
Method	CR (%)	Sharpe	CR (%)	Sharpe
Equal-Weight	17.0	0.70	18.6	0.75
Min-Variance	9.5	1.30	13.5	1.84
Mean/vol weighting	14.8	0.55	16.1	0.60
Inverse volatility	18.6	0.99	20.9	1.09
SMA(200) trend, equal-weight	19.1	0.83	20.2	0.90
MACD timing, equal-weight	23.4	1.06	27.2	1.22
RSI filter, equal-weight	16.4	0.68	17.2	0.70
Cross-sectional momentum (top decile)	21.4	0.56	21.4	0.56
RSI mean-reversion (bottom decile)	−14.6	−0.09	−15.0	−0.10

With dividend-adjusted prices, MACD attains the highest CR among these nine controls (27.2%), while Min-Variance attains Sharpe 1.84 and CR 13.5%. These results describe price-field sensitivity of the comparison strategies under their existing allocation rules.

F Applicability and Broader Impact

UCAPS applies to candidate pools with consistent historical features, executable subset-selection interfaces, and a downstream simulator matching the intended allocation constraints. The evaluated setting is a fixed pool of U.S.-listed multi-asset instruments and BTC/USD, with daily long-only allocations and a proportional target-turnover cost. Pool composition, historical signal quality, and the interaction between selected assets and rewards determine its effectiveness. The cross-pool diagnostic identifies variation in score usefulness and portfolio gains across budgets; the cadence study identifies regime-dependent trade-offs between holding a universe fixed and refreshing it with retraining.

Generated programs inherit assumptions from their prompts and pretrained models. Sandbox validation enforces the supported program interface, while financial deployment additionally requires appropriate data governance, operational isolation, and monitoring of concentration and trading costs.

The modeled cost convention suits the reported backtests; deploying at scale requires execution controls suited to liquidity and market impact. Potential risks include biased financial priors, correlated allocations across users, and treating backtest outputs as personal investment advice. The intended use is research on upstream portfolio-learning design, with human oversight for financial decisions.

G Computational Cost

Table 24 reports the workflow for reproducing the main configuration (Qwen2.5-7B) from its base weights and retained training artefacts: one adapter update, one final Selector sampling pass, then fresh PPO training and evaluation per run. Data preparation is CPU work; accelerator measurements report peak device memory. The fixed overhead is amortized across runs. This workflow starts from the retained replay/reward artefacts produced by DACE; full five-iteration search is a separate workload.

Table 24: Deployed-configuration reproduction on one 24 GB RTX 3090. Stage times are the median of 3 measurement runs. Totals sum the displayed stage times.

Stage	Wall-clock	Peak memory
Once per reproduced configuration		
FMP cache and statistics preparation	3 min	0.2 GB (host)
Rank-16 adapter update, one pass	4.5 min	23.0 GB (GPU)
Selector sampling, ten samples	8 s	14 GB (GPU)
Per PPO run		
PPO training, three stages of 10,000 steps	145 s	0.35 GB (GPU)
Test rollout, 310 returns	2.5 s	0.35 GB (GPU)
One-run workflow	$\sim$10 min	23 GB peak GPU
20-run workflow	$\sim$57 min	23 GB peak GPU

Using the displayed timings, fixed preparation is 458 seconds and each run adds 147.5 seconds. One run therefore takes about 10 minutes, and 20 runs about 57 minutes. The 23 GB peak occurs during the adapter update; PPO and evaluation use less than 1 GB of device memory. These measurements refer to the main adapter recipe (rank 16); the alternative Selector configurations use the rank-32 recipe in Table 8.

DACE evaluates 12 reward candidates under 3 screening runs in each of five DACE iterations, corresponding to 180 screening runs, together with program generation, validation, and eligible SFT updates. The retained universe/reward configuration is then evaluated by independent deployment policies. At deployment, execution of the saved selection programs and trained policy requires no LLM calls. Re-selection reuses the executable artefact and introduces a new downstream training stage for the refreshed universe.