DIHAS: Estimating Conditional Action Contributions for Reinforcement Learning in Trade Execution

Anonymous authors
Paper under double-blind review

Abstract

Executing a fixed parent order requires balancing immediate trading costs against exposure to future prices. Because final cost also depends on subsequent decisions and market innovations, it provides limited evidence about the contribution of one current action. We introduce Data-Induced High-Dimensional Action-Conditional Ambiguity Suppression (DIHAS), an action-conditional evidence layer that compares a candidate option with frozen-reference continuation under shared market innovations. A local financial surrogate and bounded residual estimate the resulting paired saving; a paired inventory control variate improves conditional-mean estimation, and ambiguity aggregation evaluates actions under alternative conditional price means. This evidence supplies bounded action biases to DDQN, PPO, and GRPO, while a held-out calibration diagnostic audits overstatement. Full-model training uses a verified terminal paired-saving reward, and final evaluation uses transaction cost. On held-out test dates within each year of the fixed 52-stock Shenzhen replay panel (2019–2021), all nine DIHAS configurations achieve lower three-year mean test-set transaction cost than locally retrained DQN and PPO baselines and TWAP. Diagnostics on Dev, an internal training diagnostic split, show lower mean trace variance with PICV and lower mean pinball loss with the residual across three algorithms. These retraining comparisons use each variant's own probes; ambiguity and policy-cost effects remain mixed.

1 Introduction

Executing a fixed parent order means deciding how quickly to sell, how much inventory to carry, and when to pay for immediate liquidity before a deadline. The same price signal can favor waiting with little inventory but favor an urgent fill when inventory is large and time is short. Execution models describe the interaction of timing, cost, and inventory risk (Almgren & Chriss, 2001; Cont & Kukanov, 2017). Yet final transaction cost also reflects later policy decisions, price innovations, and realized fills. It judges the complete policy, but does not by itself identify the contribution of one current option.

Adjacent work answers different questions. Execution RL improves policies and market representations (Pan et al., 2022; Niu et al., 2024; Zhang et al., 2023); counterfactual credit assignment uses delayed outcomes to assess actions (Mesnard et al., 2021; Meulemans et al., 2023); robust RL addresses uncertain values or models (Kumar et al., 2020; Yoo & Woo, 2025). For a fixed parent order in finite historical replay, our question is more specific: what final-cost change follows from replacing one legal option while holding the subsequent decision rule fixed? Our method defines *paired saving* by starting two branches at the same state, executing the reference option or one candidate option, and then following the same frozen reference policy. Both branches share the parent order, exogenous innovations, and matching rules, while their inventory paths and fills may differ. This is a coupled simulated contribution under a specified replay contract, not an unconditional causal effect in a live market. Figure 1 shows the two feedback routes.

That comparison exposes three concrete estimation problems. A local financial surrogate encodes schedule, quote, and inventory-time exposure, and a bounded residual corrects replay behavior that this approximation misses. Shared innovations remove path mismatch, but a changed inventory path

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.

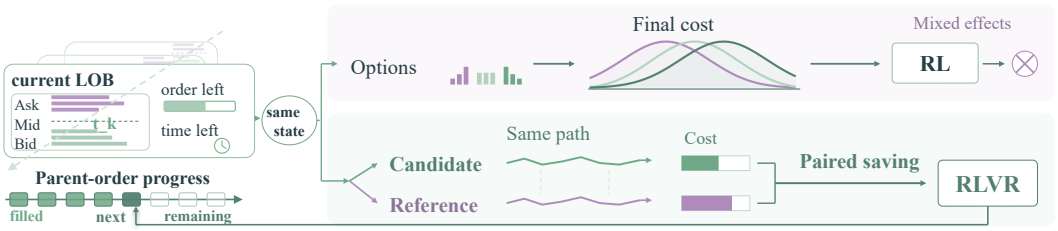


Figure 1: Parent-order execution and paired evidence. The upper route sends the terminal cost of a trajectory to RL; the lower route compares a one-option candidate (green) with frozen-reference continuation (purple) under a shared path. Book and trajectory drawings are schematic.

still loads differently on price shocks; the paired inventory control variate (PICV) targets the finite-replay variance of the conditional-mean saving. Finally, if candidate and reference carry different inventory-time exposure, a plausible shift in the conditional price mean can reverse their ranking. Our method evaluates prespecified mean scenarios to inspect this sensitivity; its fixed AA50 radius is a stress scenario rule, not a general robustness guarantee. These estimates provide bounded action biases to DDQN, PPO, and GRPO, while verified terminal reward trains the complete policy and raw transaction cost evaluates it.

To address the difficulty of estimating a current execution action's contribution from mixed trajectory outcomes, we propose an action-conditional evidence method with three contributions: (1) a paired-saving target defined by one-option replacement, frozen continuation, and shared innovations under a fixed replay contract; (2) a financial estimator and bounded feedback interface that address approximation error, finite-replay variance, and conditional-mean sensitivity; and (3) an empirical assessment separating complete-system test-set transaction costs from component diagnostics and failure cases. On held-out dates within each of 2019–2021, the complete systems improve on the available local execution references. The evidence-layer contribution requires a matched interface and reward; estimator diagnostics and mean-scenario cases test narrower claims and do not establish uniform policy gains.

2 Related work

Financial execution models explain how trading schedules interact with price risk and the mechanisms governing fills. Fixed-quantity models connect trading schedules to market impact and price risk (Almgren & Chriss, 2001; Donnelly, 2022); recent actor–critic analysis studies policy learning and error within a continuous-time execution model (Wang et al., 2026). Order-placement models make order flow, queues, and fees relevant to the market-versus-limit choice (Cont & Kukanov, 2017). Empirical and strategic models further characterize trading patience, liquidity supply to large sellers, and information in exchange messages (Chiu et al., 2017; Capponi et al., 2025; Cartea et al., 2025). These mechanisms motivate our inventory-and-schedule surrogate, while paired replay supplies the comparison between alternative continuations of the same parent order.

Execution RL learns adaptive policies, while reliability-oriented methods address the limitations of their training data and value estimates. RL adapts timing, quantity, and limit prices (Akbarzadeh et al., 2018; Moallemi & Wang, 2022; Pan et al., 2022). Multi-order, hierarchical, joint, preference-based, and dynamic-action methods expand execution choices (Fang et al., 2023; Niu et al., 2024; Li et al., 2024; Xu et al., 2025; Cheridito & Weiss, 2026; Wu & Li, 2026). CASH and CATE encode finite histories (Zhang et al., 2023); conservative, implicit, ensemble, and model-risk-sensitive RL address limited or shifted data (Kumar et al., 2020; Kostrikov et al., 2022; Ghasemipour et al., 2022; Yoo & Woo, 2025). Our evidence interface complements policy learning by specifying a local comparison under fixed continuation and matched innovations.

Counterfactual credit-assignment methods investigate how an action contributes to outcomes observed later in a trajectory. Future-conditioned policy-gradient baselines separate action influence from external luck (Mesnard et al., 2021); counterfactual contribution analysis asks whether a later rewarding event would still occur after another action (Meulemans et al., 2023). Occupancy-based shaping also connects delayed rewards to earlier decisions (Venugopal et al., 2026). These are substantive credit-assignment methods. Our distinction is the comparison target and the execution con-

straints: one legal option replaces the current reference action, the continuation policy is frozen, the innovations are coupled, and the outcome is final parent-order cost after changed fills and inventory exposure. Financial structure, PICV, and mean scenarios estimate and interrogate this specified paired-saving target rather than claiming a general solution to credit assignment.

3 PRELIMINARIES

Historical order-book feeds do not identify brokers' client parent orders. We therefore fix a proxy sell order before execution: its quantity is $Q_{i,d} = \rho V_{i,d^-}^{t_{\text{cut}}}$, using volume observed by a prespecified cutoff on the preceding available trading day. All policies face the same parent order, exogenous market path, minute-grid matcher, terminal completion, and accounting rules. Sell-side transaction cost is $\text{TC} = 10^4(1 - P^{\text{exec}}/P^{\text{TWAP}})$ bp, where P^{exec} is the volume-weighted execution price and P^{TWAP} is the execution-window mean midprice. Lower TC is better; the benchmark rule is fixed beforehand, but its realized value is unavailable to the policy during execution. Matching allows depth-limited partial fills and capped passive fills, with terminal completion of remaining inventory.

At each minute the policy sees causal order-book history, remaining inventory, time to deadline, and past fills. The progress deviation $\ell_t = I_t/Q_{i,d} - \tau_t$ is positive when execution lags a uniform schedule. A legal option adjusts a state-dependent neutral quote and quantity for one to three minutes. The 12 options cover neutral, patient liquidity, patient size-up, catch-up urgency, and impact control; quotes and quantities still respond to the observed state. Each option remaps its command every minute, uses the neutral command when a shifted index is invalid, and advances by its duration or the remaining horizon, whichever is shorter.

Let $\pi_{\text{ref}}^{(r)}$ be a frozen reference policy. At state s, replace its current action by legal option o once, then return to the same reference. For a conditional price mean μ and shared exogenous innovations ω, define $Y^{(r)}(s, o, \mu, \omega) = \text{TC}(\pi_{\text{ref}}^{(r)}; \mu, \omega) - \text{TC}(o \to \pi_{\text{ref}}^{(r)}; \mu, \omega)$. The branches share the parent order, initial state, innovations, and execution rules; positive paired saving means that the replacement lowers cost. In particular, the reference action has zero saving. We first average path innovations at fixed mean, $M^{(r)}(s, o, \mu) = \mathbb{E}_\omega Y^{(r)}(s, o, \mu, \omega)$. On the prespecified interval $\mathcal{U}_\phi(s) = [\widehat{\mu}(s) - \phi(s), \widehat{\mu}(s) + \phi(s)]$, pessimistic, center, and optimistic evidence use the minimum of $M^{(r)}$, its center value, and its maximum, respectively. Thus path averaging precedes ambiguity aggregation; the interval adapts Sun et al. (2023) to paired saving. The ambiguity attitude is set by the evaluation protocol.

Full-policy training uses a separate cost difference, $\Delta\text{TC} = \text{TC}(\pi_{\text{ref}}^{(r)}) - \text{TC}(\pi_{\text{candidate}})$, comparing complete policies. The terminal reward is $R_{\text{train}}^{(r)} = \tanh(\Delta\text{TC}/5\,\text{bp})$; all preterminal rewards are zero. Final evaluation uses untransformed TC.

4 DIHAS METHODOLOGY

4.1 EVIDENCE INTERFACE AND SCOPE

To compare candidates against a common reference, each refresh freezes the policy and replays all legal candidates under shared innovations. The estimator, PICV, and calibration statistics are updated, then the evidence is frozen for RL rollouts. The financial estimator and RL policy model have separate parameters and optimizers; RL losses do not backpropagate through the evidence model. Figure 2 maps these computations onto the execution loop.

4.2 STRUCTURED ESTIMATION OF CONDITIONAL ACTION CONTRIBUTIONS

4.2.1 ZERO-SUM SCHEDULE ADJUSTMENTS AND LOCAL FINANCIAL SURROGATE

Following the fixed-quantity liquidation setting of Almgren & Chriss (2001), let $z \in \mathbb{R}^{H_s}$ adjust the normalized schedule from the current time to the deadline. The constraint $\mathbf{1}^\top z = 0$ redistributes planned execution without changing total quantity. DIHAS uses the local cost $C_\theta(s, z) = g_\theta(s)^\top z + \frac{1}{2} z^\top H_\theta(s) z$, where $H_\theta(s) = D_\theta(s) + U_\theta(s) U_\theta(s)^\top$. Its positive diagonal term controls local curvature; the low-rank term couples liquidity, inventory, and deadline pressure across time.

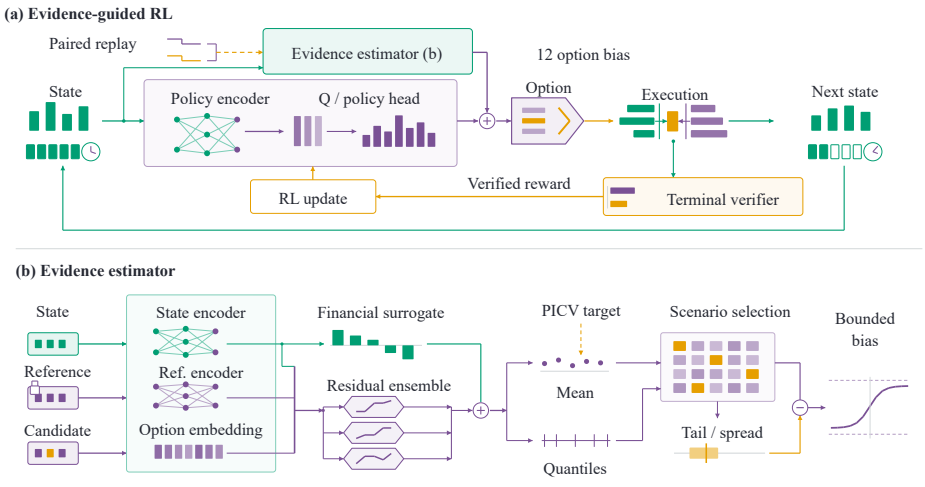


Figure 2: DIHAS computation and execution loop. **(a)** Paired replay supplies an option bias to the policy; execution yields the next state and verified reward for RL updates. **(b)** The financial surrogate and residual ensemble combine with PICV, scenario selection, and tail/spread penalties to produce bounded evidence feedback.

The surrogate saving of candidate o is $A_\theta(s,o) = C_\theta(s, z_{\text{ref}}) - C_\theta(s, z_o) + \alpha_\theta(s)\delta^p - \frac{1}{2}\lambda_\theta(s)(\delta^p)^2$. Here z_{ref} is a learned state-conditioned template, not the frozen policy's realized fills. The remaining terms represent price improvement and quote-displacement risk. Inventory timing also changes drift exposure, represented by $f_{\text{fin}}(s,o,\delta\mu) = A_\theta(s,o) + B_\theta(s,o)\delta\mu$. In this expression, $B_\theta(s,o)$ is the candidate–reference difference in inventory-time exposure, and $\delta\mu$ is drift displacement from the state-conditioned center. This surrogate gives an economic direction for option ranking; it does not estimate the true market-cost Hessian. The replay does not model queue position or endogenous liquidity recovery.

4.2.2 Financial surrogate and bounded residual

To model bounded corrections, ensemble member e uses normalized mean displacement $u = (\mu - \widehat{\mu}(s))/s_{\text{fit}}$. The paired conditional mean is estimated by $\widehat{M}_e^{(r)}(s,o,u) = f_{\text{fin}}(s,o,us_{\text{fit}}) + d_\theta(s,o)\tanh\big(\rho_{\psi_e}(s,o,u,d_r)\big)$, where $d_\theta(s,o) > 0$ bounds the residual magnitude and d_r describes the frozen reference policy. Piecewise-linear interpolation connects the finite mean nodes. Auxiliary quantile and positive-saving heads are fitted to the original paired outcomes, alongside the conditional-mean head.

4.3 Residual uncertainty control and PICV

Mean saving alone does not characterize uncertainty. DIHAS retains $E^{(r)}(s,o) = (\widehat{\Delta}, \widehat{q}_{0.1}, \widehat{p}^{+}, D_{\text{ens}}, \text{LCB}, p_{\text{router}})$ at selected mean-scenario nodes: mean saving, lower quantile, positive-saving probability, ensemble disagreement, calibrated conservative evidence, and option-family support. The raw evidence score is $S_{\text{raw}}(s,o) = \widehat{\Delta} - \lambda_q(\widehat{\Delta} - \widehat{q}_{0.1})_+ - \lambda_e D_{\text{ens}}(s,o)$. Separate calibration dates within the training split estimate an option-level overestimation radius $r_{\text{cal}}(o)$, giving $\text{LCB}(s,o) = S_{\text{raw}}(s,o) - r_{\text{cal}}(o)$. The LCB and router are audit diagnostics; neither masks actions nor enters the reported bias, which uses S_{raw} without subtracting r_{cal}.

Even with shared innovations, the paired conditional mean can retain high Monte Carlo variance because the candidate changes future inventory exposure to price innovations. Before innovation j, let $\Delta I_j(s,o)$ be the normalized remaining-inventory difference between candidate and reference, measurable with respect to $\mathcal{F}_{j-1}$. Define the Paired Inventory Control Variate $h(s,o,\omega) = s_{\text{fit}}\sum_{j=0}^{H-1}\Delta I_j(s,o)\xi_j$, with $\mathbb{E}[\xi_j \mid \mathcal{F}_{j-1}] = 0$. Then $\mathbb{E}[h \mid s,o,\mu,r] = 0$, and the controlled target for option family $g(o)$ is $\overline{Y}_{\text{cv}}(s,o,\mu) = L^{-1}\sum_{\ell=1}^{L}[Y_\ell^{(r)}(s,o,\mu) - \beta_{g(o)}h_\ell(s,o)]$. This target has the same conditional expectation as the original paired saving but can have lower finite-replay Monte Carlo variance. Coefficients are estimated by option family on fit dates and audited on disjoint

guard dates within the training split. They are enabled only if a trading-day block bootstrap satisfies $\mathrm{UCB}_{0.95}(\widehat{\rho}_M) < 1$; otherwise all coefficients revert to zero. PICV changes only the mean-head target. The quantile and positive-saving heads retain the original Y, preserving its tail and directional semantics.

4.4 Bounded evidence feedback for RL algorithms

To guide learning through a common interface, frozen evidence becomes the bounded action bias

$$b_E(s, o, z) = \mathrm{clip}\left[B \tanh\left(\frac{\gamma_E S_{\text{raw}}(s, o)}{\tau_E + D_{\text{ens}}(s, o)}\right) + \eta_E \log p_{\text{router}}(g(o) \mid s, z), -B, B\right].$$

Ensemble disagreement attenuates raw evidence. Reported runs set $\eta_E = 0$. The reference option and actions blocked by budget or cooldown receive zero bias.

For DDQN, $o_t^* = \arg\max_o [Q_\psi(s_t, o) + b_E(s_t, o, z)]$ selects the online next action; b_E also affects behavior, while the target network evaluates the selected action with its unmodified Q-value. PPO and GRPO use $\pi_{\psi,E}(o \mid s, z) \propto \exp(\ell_\psi(s, o) + b_E(s, o, z))$. Rollouts store behavior probabilities under frozen evidence bias, which is reconstructed in policy ratios. In full-model training, all three algorithms consume $R_{\text{train}}^{(r)}$, and GRPO forms within-group advantages from it.

5 Experiments

5.1 Experiment setup

Dataset. We evaluate fixed 30-minute sell orders on 52 Shenzhen stocks with local trade and order feeds in 2019–2021. Within each year, eligible stock-days are screened and split chronologically into training, validation, and final-evaluation test sets; models are trained separately. Fit, Dev (internal diagnostic), PICV guard, and calibration dates within training are disjoint; Dev uses neither validation nor test data. The stock universe is fixed across years; each eligible window requires a valid five-level book during lookback and execution.

Metrics and reproducibility. The primary outcome is sell-side TC in bp, weighted by evaluation-window repeat counts within each year and averaged equally across years; lower is better. These weights restore episode multiplicity, not order notional. Learned-policy results summarize five fixed seeds by mean and sample SD, which is not a trading-date confidence interval. DIHAS and local DQN/PPO use 120K-step checkpoints. The adapted OPD teacher and student each use 120K steps. All systems share the parent orders, minute matcher, terminal completion, and cost accounting. DIHAS adds counterfactual replay and estimator updates, so primitive-step budgets do not imply equal total compute.

5.2 Experiment results

The experiments assess execution cost, component diagnostics, scenario-sensitive choices, and the boundaries of their policy-level benefits:

RQ1. Does the full system lower test-set TC across years and matched orders?

RQ2. Do PICV and the residual improve their intended estimation targets?

RQ3. Can conditional-mean scenarios change an option choice at a fixed checkpoint?

RQ4. Where do local evidence gains translate into lower transaction cost, and where do they fail?

Main Experiment (RQ1)

To answer RQ1, we compare nine DIHAS configurations with local DQN/PPO implementations following Zhang et al. (2023), an adapted OPD student (Fang et al., 2021), TWAP, and exploratory momentum. Local DQN/PPO use 99 primitive actions; DIHAS uses 12 options with historical-prefix, CASH, or CATE inputs. OPD uses five quantities and a future-observing teacher during training; only its causal student is evaluated. TWAP executes uniformly, while momentum adjusts quantity to recent price direction and is not an exact published replication. The base DIHAS configurations for

the first three 2020 seeds use 120K milestones from runs planned for 480K, which changes DDQN's exploration schedule. A three-seed 2020 evidence-off archive preserves the 12 options and paired reward but mixes planned horizons. Its off-minus-Full differences are -0.268, -0.185, and $+0.160$ bp for DDQN, PPO, and GRPO: mixed outcomes that do not replace a fully matched panel.

We first assess complete-system performance under the common replay rules. Table 1 summarizes DIHAS and local execution systems by test-set TC, the primary outcome; annual weighted costs are averaged within each seed before the five-seed summary. The signed Gap $=$ $\text{TC}_{\text{test}} - \text{TC}_{\text{train}}$ describes the split difference.

Table 1: Fixed-parent-order execution. Parentheses show learned-policy seed SD; TWAP and Momentum are deterministic. Bold and underline mark the lowest and second-lowest cost means. Gap is signed: test-set TC − training-set TC; a negative value means lower test-set TC. OPD uses an adapted two-stage budget.

Algorithm	**Training (bp) ↓**	**Validation (bp) ↓**	**Testing (bp) ↓**	**Gap (bp)**
		Annual average (2019–2021)		
TWAP	8.4426	8.1310	7.6245	−0.8181
Momentum	10.1195	9.6352	9.4156	−0.7039
DQN baseline [Zhang et al., 2023]	2.2231 (1.4589)	2.6454 (1.4253)	2.3895 (1.4617)	0.1663
DIHAS + DDQN	−2.6442 (0.2132)	−2.2484 (0.2579)	−2.1506 (0.1376)	0.4935
DIHAS + DDQN + CASH	−2.5262 (0.1528)	−2.1271 (0.1734)	−2.0477 (0.2338)	0.4785
DIHAS + DDQN + CATE	−2.6521 (0.1690)	−2.1251 (0.2029)	−2.0907 (0.0893)	0.5613
PPO baseline [Zhang et al., 2023]	4.0950 (1.5089)	4.6665 (1.4126)	4.3819 (1.1116)	0.2868
OPD baseline [Fang et al., 2021]	9.5583 (0.5441)	9.2817 (0.6180)	9.2598 (0.7393)	−0.2985
DIHAS + PPO	−2.6427 (0.0739)	−2.2934 (0.1600)	−2.2466 (0.1710)	0.3961
DIHAS + PPO + CASH	−2.6152 (0.1092)	−2.3484 (0.2564)	−2.1612 (0.1431)	0.4541
DIHAS + PPO + CATE	−2.4940 (0.1862)	−1.9843 (0.1930)	−1.9033 (0.1455)	0.5908
DIHAS + GRPO	**−2.9458 (0.1541)**	**−2.6254 (0.1456)**	**−2.4914 (0.0832)**	0.4544
DIHAS + GRPO + CASH	−2.8538 (0.2274)	−2.5533 (0.2008)	−2.4891 (0.0970)	0.3648
DIHAS + GRPO + CATE	−2.8909 (0.1895)	−2.5503 (0.3167)	−2.4414 (0.2390)	0.4495

All nine DIHAS configurations have negative three-year mean test-set TC, from -2.4914 to -1.9033 bp; every external reference has a positive mean. Neither CASH nor CATE improves its base algorithm on the three-year mean. The 0.0023 bp difference between base GRPO and GRPO+CASH is small relative to their seed SDs and does not establish a ranking. Figure 3 resolves the DQN/PPO contrasts by year and seed. All 90 seed-level differences favor DIHAS, but repeated dates and seed IDs make them correlated; no local GRPO reference was trained with the external baseline interface.

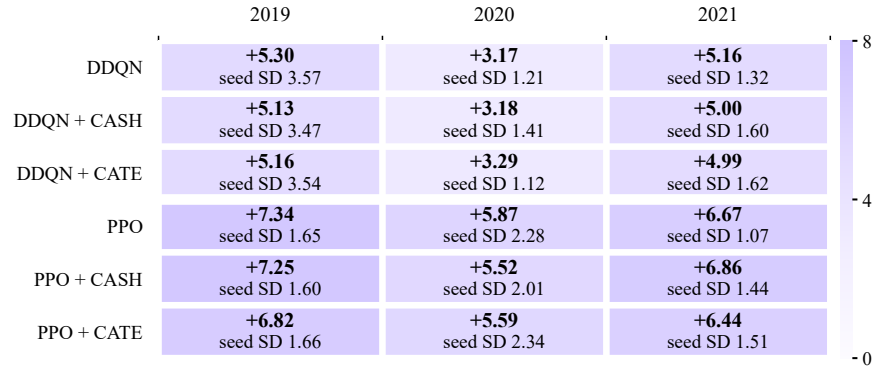


Figure 3: Local DQN/PPO baseline minus DIHAS test-set TC, paired by seed ID. Each cell gives the five-seed mean (large text) and sample seed SD (small text), in bp; color encodes the mean. Positive values favor DIHAS.

An equal-year aggregate can conceal period-specific failures. Figure 4 checks the annual direction. All nine DIHAS configurations have negative five-seed mean cost in each observed year, while the local DQN/PPO baselines, OPD adaptation, and deterministic rules have positive annual means. This is within-panel directional coverage, not evidence of transfer to an unseen market or a new regime.

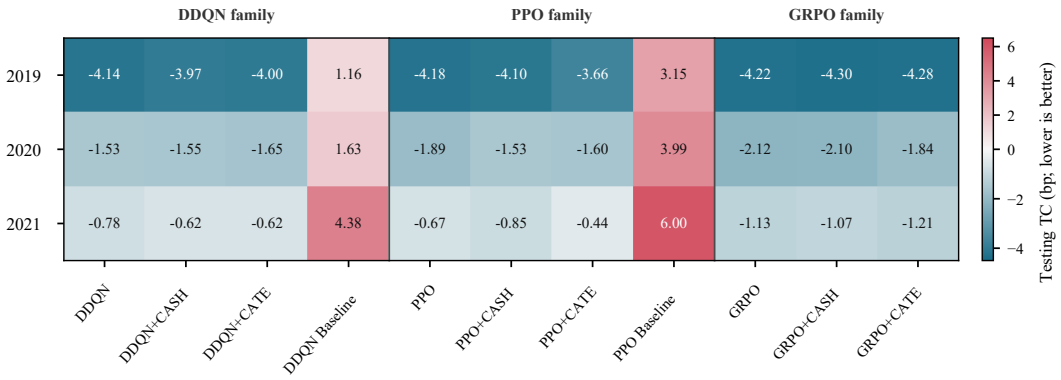


Figure 4: Annual weighted test-set TC of DIHAS variants and local DQN/PPO policies. Cells show five-seed means in bp; lower values are better. Columns are grouped by policy algorithm.

Annual rankings vary: GRPO+CATE beats base GRPO in 2021 by only 0.0748 bp, less than their seed SDs. Across 30 base DDQN/PPO year–algorithm–seed comparisons, the count-weighted episode share with lower DIHAS TC is 59.3%–88.4%. A post hoc paired bootstrap resamples five consecutive test dates in circular blocks, retaining all stocks, windows, and repeat counts within each date. After averaging the five fixed seeds, 4,000 resamples give equal-year baseline-minus-DIHAS differences of 4.540 bp (95% interval 4.328–4.740) for DDQN and 6.628 bp (6.215–7.012) for PPO. All six annual intervals are positive. These marginal intervals quantify date variation conditional on the trained seeds; they neither include retraining uncertainty nor isolate evidence feedback.

Component Analysis (RQ2)

Lower policy cost alone does not show why the estimator needs PICV or a residual. We therefore test their intended targets separately: finite-replay variance for PICV and prediction error beyond the financial surrogate for the residual. Diagnostics use Dev dates within the training split. Each removal variant is independently retrained for 120K steps with the same parent orders, 12-option interface, budget, optimizer, seed identifiers, and protocol for each algorithm, then evaluated on its own Dev probe. Figure 5 reports these diagnostics. Residual action regret is the replay-draw saving gap between the best legal option and the option selected by the estimated mean; the zero-saving reference remains available. Retraining also changes each variant's states and reference decisions, so these differences describe the removal training procedure rather than an isolated estimator effect.

Algorithm	Variant	Trace var. (bp²) ↓
DDQN	**DIHAS**	**20.37±5.73**
	w/o PICV	29.48±4.98
PPO	**DIHAS**	**17.49±6.04**
	w/o PICV	18.98±3.90
GRPO	**DIHAS**	**13.23±1.78**
	w/o PICV	17.60±3.37

Algorithm	Variant	Pinball (bp) ↓	Regret (bp) ↓
DDQN	**DIHAS**	**0.5534±0.0312**	**1.6654±0.0573**
	w/o residual	0.6251±0.0456	1.7044±0.0866
PPO	**DIHAS**	**0.4889±0.0340**	**1.5959±0.0576**
	w/o residual	0.5258±0.0320	1.6037±0.1063
GRPO	**DIHAS**	**0.4794±0.0191**	**1.5005±0.0778**
	w/o residual	0.5189±0.0120	1.5872±0.0331

Figure 5: Estimator diagnostics after independent retraining. Left: paired conditional-mean trace variance with and without PICV. Right: quantile pinball loss and action regret with and without the bounded residual. Entries are equal-year, five-seed mean ± sample SD on Dev; bold marks the lower mean.

Removing PICV increases paired conditional-mean trace variance in all three algorithms. Relative to w/o PICV, Full lowers the equal-year Dev variance by 30.9%, 7.9%, and 24.8% for DDQN, PPO, and GRPO. Removing the residual increases quantile pinball loss numerically by 0.0717, 0.0369, and 0.0394 bp, respectively; its PPO action-regret difference is only 0.0078 bp. These mean differences are consistent with the intended estimation roles; the small PPO regret change also shows that better prediction need not materially improve option selection. The financial surrogate has no separately reported held-out saving-prediction comparison; its available strategy-cost removal contrast appears in RQ4.

Beyond average diagnostics, we test variance control across individual runs using the PICV guard. On training dates disjoint from coefficient fitting, all 45 year–algorithm–seed variance-ratio upper

bounds are below one (Figure 6a). Panel b identifies 2020 PPO as closest to failure: its largest bound, 0.932, leaves a margin of 0.068. Every stored run therefore passes the guard on these dates; action-ranking recovery and final test-set TC are separate outcomes.

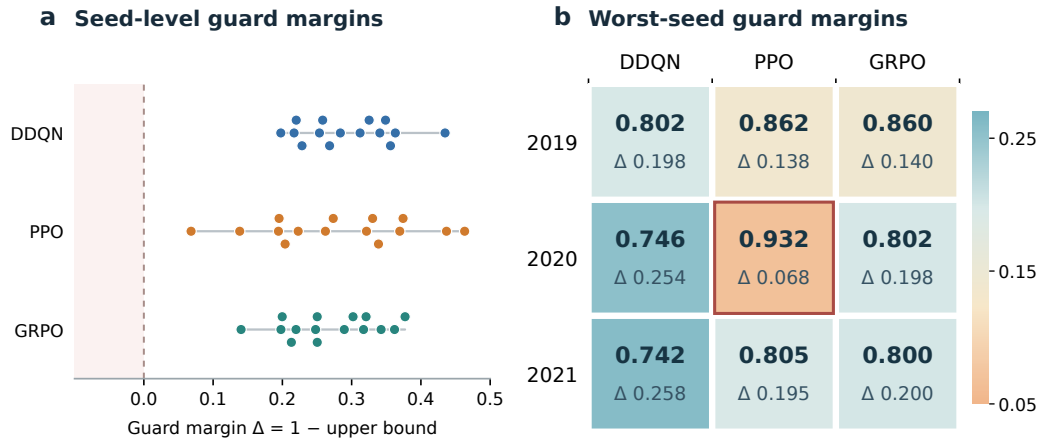


Figure 6: PICV guard margins $\Delta = 1 - U$; U is the seed-level 95% trading-day block-bootstrap upper bound on the conditional-mean variance ratio. **(a)** All 45 records, 15 per algorithm; vertical offsets separate points, gray lines show observed ranges, and shading marks failure ($\Delta \leq 0$). **(b)** Largest bound and its margin among five seeds per cell; color encodes the margin. The outlined 2020 PPO cell is closest to failure.

Conditional-Mean Analysis (RQ3)

RQ3 isolates scenario-dependent choices at identical Full checkpoints and Dev states. We compare AA50 (radius $0.5s_{\text{fit}}$) with center-only CENTER0. Worst-case saving is the selected option's minimum conditional-mean saving across nodes; ambiguity regret compares it with the best legal option at each node. Across 45 seed records, AA50 improves mean worst-case saving in all nine year–algorithm cells and mean regret in eight, except 2020 DDQN. Several trading-day block-bootstrap 95% intervals cross zero, so these means do not imply uniform seed-level gains.

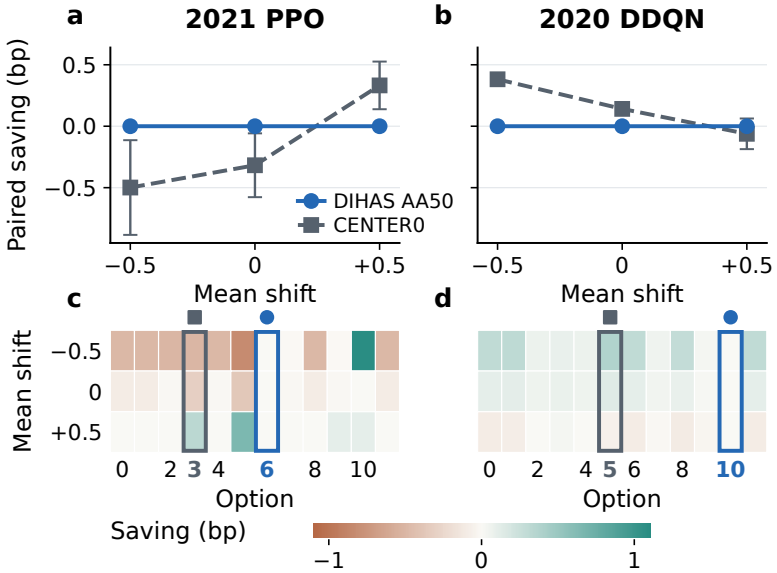


Figure 7: Post hoc Dev cases. **a,b:** Selected-option saving (mean ± replay SE, 16 draws/node). **c,d:** All 12 options on a shared mean-saving scale. Circles/blue outlines mark AA50 reference options 6 and 10; squares/gray outlines mark CENTER0 options 3 and 5. Mean shifts use fitted-scale units.

Two post hoc cases illustrate these choice changes. In Figure 7, AA50 retains the zero-saving reference; CENTER0 selects options whose saving changes sign across scenarios. The cases have median absolute regret changes within favorable 2021 PPO and adverse 2020 DDQN strata, restricted to switched choices with crossing endpoint savings. RQ4 examines their opposite regret outcomes against the full 12-option landscape.

Empirical Boundaries (RQ4)

RQ4 examines where local gains fail under historical-market replay. In Figure 7, AA50 raises worst-node saving to zero in both cases, from -0.499 bp in 2021 PPO and -0.062 bp in 2020 DDQN. Yet local regret falls by 0.161 bp in the former and rises by 0.154 bp in the latter. Panel d explains the opportunity cost: CENTER0's option 5 has the highest replay mean saving at the negative and central nodes. Retaining the reference forgoes these gains. Worst-node protection thus need not lower regret; these selected cases illustrate the trade-off, not its prevalence.

We next examine this trade-off after independent retraining. In Figure 8, Full improves mean worst-case saving over w/o ambiguity in all three algorithms. Regret decreases by 0.0405 bp for DDQN and only 0.0042 bp for GRPO, but increases by 0.0278 bp for PPO. The same distinction therefore appears in the retrained comparisons.

Algorithm	Variant	Worst saving (bp) ↑	Regret (bp) ↓
DDQN	**DIHAS**	**-0.125±0.108**	**0.8752±0.0476**
	w/o ambiguity	-0.535±0.060	0.9158±0.0469
PPO	**DIHAS**	**-0.343±0.193**	0.8272±0.0430
	w/o ambiguity	-0.383±0.082	**0.7994±0.0483**
GRPO	**DIHAS**	**-0.126±0.043**	**0.8011±0.0294**
	w/o ambiguity	-0.177±0.050	0.8053±0.0193

Figure 8: Full versus w/o ambiguity after independent retraining. Entries are Dev five-seed mean $\pm$ sample SD; higher worst-scenario saving and lower ambiguity regret are better.

At the policy level, Full lowers Dev TC in 5/9, 7/9, 5/9, and 7/9 year–algorithm cells against removal of PICV, residual, financial surrogate, and ambiguity aggregation, respectively. These exploratory counts lack date-paired uncertainty. Retraining changes checkpoints, reference decisions, and visited states, preventing single-component attribution. RQ4 therefore finds lower replay cost in some removal comparisons, without uniform policy gains.

Only radii zero and $0.5s_{\text{fit}}$ are compared; the radius–response relation and market-state-specific failure conditions remain unresolved.

5.3 Discussion

Interpreting these boundaries requires separating policy adaptation from market response. DIHAS remaps quotes and quantities each minute using causal book history, inventory, remaining time, and past fills. These audits do not stratify effectiveness by spread, depth, volatility, or urgency. Aligned evidence-on/off and exposure-stratified radius tests remain needed for attribution. The replay keeps market paths exogenous and omits queue position and endogenous liquidity recovery; live validation requires event-level fills and endogenous market response (Pace et al., 2024).

6 Conclusion

This paper presented DIHAS for estimating the contribution of one execution option under frozen-reference continuation. We defined paired saving with shared innovations, combined a financial surrogate and bounded residual with PICV and mean-scenario aggregation, and integrated this evidence into DDQN, PPO, and GRPO. Experiments on the 2019–2021 Shenzhen replay panel show lower equal-year mean test-set TC across all nine configurations than the local references, while component audits and failure cases reveal that local gains do not uniformly improve policy cost. Future work will isolate evidence-layer effects, test market-state and radius sensitivity, and evaluate event-level fills and endogenous market response.

AI Use Statement

Generative AI tools were used solely to assist with language editing, clarity, and formatting during manuscript preparation. They were not used to conduct experiments, derive scientific results, or make research decisions. The authors reviewed all AI-assisted content and remain fully responsible for the scientific content, analysis, and conclusions.

Reproducibility Statement

The paper describes the model architecture, learning objectives, historical replay data, baselines, evaluation metrics, and experimental settings. The supplementary sections document data selection, information access, training budgets, hyperparameters, aggregation rules, and additional diagnostic results. They also give conditional derivations for the paired estimator and inventory control variate, making their assumptions explicit. The accompanying source package includes the manuscript, figures, and selected numerical summaries; it does not include raw market feeds or all training checkpoints. Reproducing the full experiments requires those inputs as well as the recorded execution protocol.

Ethics Statement

This study evaluates simulated execution on historical market feeds and involves no interventions with human participants. The parent orders are constructed proxies rather than identified client instructions. Access to and redistribution of the underlying feeds depend on the data providers' terms; this paper does not claim that all raw data are publicly available. Replay assumptions may understate execution costs or market impact, and strategies can inherit biases from the selected stocks and dates. Any live deployment would require independent validation of execution risk, market impact, and operational controls, together with appropriate human oversight. The authors remain responsible for research integrity and compliance with the ICLR Code of Ethics.

References

Nima Akbarzadeh, Cem Tekin, and Mihaela van der Schaar. Online learning in limit order book trade execution. *IEEE Transactions on Signal Processing*, 66(17):4626–4641, 2018.

Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3: 5–40, 2001.

Agostino Capponi, Albert J. Menkveld, and Hongzhong Zhang. Large orders in small markets: execution with endogenous liquidity supply. *Review of Finance*, 29(1):201–239, 2025.

Álvaro Cartea, Samuel N Cohen, Robert Graumans, Saad Labyad, Leandro Sánchez-Betancourt, and Leon van Veldhuijzen. Statistical predictions of trading strategies in electronic markets. *Journal of Financial Econometrics*, 23(2):nbae025, 2025.

Patrick Cheridito and Moritz Weiss. Reinforcement learning for trade execution with market and limit orders. *Quantitative Finance*, pp. 1–21, 2026.

Junmao Chiu, Huimin Chung, and George HK Wang. Order aggressiveness, trading patience, and trader types in a limit order market. *Journal of Futures Markets*, 37(11):1094–1123, 2017.

Rama Cont and Arseniy Kukanov. Optimal order placement in limit order markets. *Quantitative Finance*, 17(1):21–39, 2017.

Ryan Donnelly. Optimal execution: A review. *Applied Mathematical Finance*, 29(3):181–212, 2022.

Yuchen Fang, Kan Ren, Weiqing Liu, Dong Zhou, Weinan Zhang, Jiang Bian, Yong Yu, and Tie-Yan Liu. Universal trading for order execution with oracle policy distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 107–115, 2021.

Yuchen Fang, Zhenggang Tang, Kan Ren, Weiqing Liu, Li Zhao, Jiang Bian, Dongsheng Li, Weinan Zhang, Yong Yu, and Tie-Yan Liu. Learning multi-agent intention-aware communication for optimal multi-order execution in finance. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4003–4012, 2023.

Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems*, 35:18267–18281, 2022.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Siyuan Li, Hui Niu, Jiani Lu, and Peng Liu. Joint learning of volume scheduling and order placement policies for optimal order execution. *Mathematics*, 12(21):3440, 2024.

Thomas Mesnard, Theophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Thomas S Stepleton, Nicolas Heess, Arthur Guez, et al. Counterfactual credit assignment in model-free reinforcement learning. In *International Conference on Machine Learning*, pp. 7654–7664. PMLR, 2021.

Alexander Meulemans, Simon Schug, Seijin Kobayashi, Gregory Wayne, et al. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. *Advances in Neural Information Processing Systems*, 36:68685–68735, 2023.

Ciamac C Moallemi and Muye Wang. A reinforcement learning approach to optimal execution. *Quantitative Finance*, 22(6):1051–1069, 2022.

Hui Niu, Siyuan Li, and Jian Li. Macmic: Executing iceberg orders via hierarchical reinforcement learning. In *IJCAI*, pp. 6008–6016, 2024.

Alizée Pace, Hugo Yèche, David Ha, Gunnar Ratsch, and Guy Tennenholtz. Delphic offline reinforcement learning under nonidentifiable hidden confounding. In *International Conference on Learning Representations*, volume 2024, pp. 2768–2796, 2024.

Feiyang Pan, Tongzhe Zhang, Ling Luo, Jia He, and Shuoling Liu. Learn continuously, act discretely: Hybrid action-space reinforcement learning for optimal execution. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 3912–3918, 2022. doi: 10.24963/ijcai.2022/543. URL https://www.ijcai.org/proceedings/2022/543.

Yuzhe Sun, Yanjie Wang, Shunming Zhang, and Helen Huang. The impact of ambiguity-loving attitude on market participation and asset pricing. *Economic Modelling*, 128:106527, 2023.

Aravind Venugopal, Jiayu Chen, Xudong Wu, Chongyi Zheng, Benjamin Eysenbach, and Jeff Schneider. Occupancy reward shaping: Improving credit assignment for offline goal-conditioned reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5683583b88da51c79d2eb263f295286e-Abstract-Conference.html.

Boyu Wang, Xuefeng Gao, and Lingfei Li. Reinforcement learning for continuous-time optimal execution: actor–critic algorithm and error analysis. *Finance and Stochastics*, 30(2):597–655, 2026.

Pangjing Wu and Xiaodong Li. Dynamic action space reinforcement learning for optimal trading execution. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1883–1891, 2026.

Haohan Xu, Jason Bohne, Pawel Polak, David Byrd, David Rosenberg, and Gary Kazantsev. Learning to trade with preferences: Interpretable execution via mixture-of-experts. In *Proceedings of the 6th ACM International Conference on AI in Finance*, pp. 762–770, 2025.

Gwangpyo Yoo and Honguk Woo. Model risk-sensitive offline reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 82343–82362, 2025.

Chuheng Zhang, Yitong Duan, Xiaoyu Chen, Jianyu Chen, Jian Li, and Li Zhao. Towards generalizable reinforcement learning for trade execution. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4975–4983, 2023. doi: 10.24963/ijcai.2023/553. URL https://www.ijcai.org/proceedings/2023/553.

A Annual Testing TC and seed dispersion (RQ1)

To assess whether the aggregate RQ1 result masks annual variation, we report each year's cost and seed dispersion. Table A2 expands Figure 4 with sample SD across the five fixed seeds and adds the adapted OPD student and deterministic rules. Each learned policy's annual TC first averages the eligible parent orders in that year using recorded repeat weights; the displayed mean and SD then aggregate five seed values. These within-year SDs differ from Table 1's SD after averaging all three years within each seed. The common parent-order and matcher protocol applies to every row, while the local DQN/PPO baselines retain their 99-action interface and adapted OPD uses five quantity actions.

A.1 Aggregation and uncertainty

A source record can represent multiple occurrences of the same execution window. Let $c_{y,k,i}$ be its transaction cost for year y, seed k, and unique window i, with recorded repeat count $w_{y,i}$. The annual statistic is

$$T_{y,k} = \frac{\sum_i w_{y,i} c_{y,k,i}}{\sum_i w_{y,i}}, \qquad \overline{T}_k = \frac{1}{3} \sum_{y=2019}^{2021} T_{y,k}.$$

The main table reports $\overline{T} = 5^{-1} \sum_k \overline{T}_k$ and sample SD $s_T = [\sum_k (\overline{T}_k - \overline{T})^2/4]^{1/2}$. Thus years receive equal weight after the recorded within-year aggregation. The weights restore episode multiplicity; they do not create independent trading dates. Deterministic rules have no training-seed SD. Gap is the signed Testing-minus-Training difference, so a negative value indicates lower cost on Testing and is not an absolute generalization-error measure.

To distinguish training variation from variation across market dates, we keep their uncertainty summaries separate. The seed-paired plot forms the local-baseline-minus-DIHAS difference within each year and seed before averaging five differences. Its SD describes seed variation, whereas the matched-order coverage plot reports $\sum_i w_i \mathbb{1}\{c_i^{\text{DIHAS}} < c_i^{\text{base}}\} / \sum_i w_i$. Neither statistic alone supplies a trading-date confidence interval. Our additional analysis first averages paired cost differences over the five fixed seeds, then resamples ordered Testing dates in circular moving blocks of five days within each year. Each resampled date retains every stock, window, and repeat count; candidate and baseline use the same resampling indices. Independent year resamples are combined with equal year weights. We use 4,000 replicates and marginal percentile 95% intervals, conditional on these trained seeds, with no multiplicity adjustment. This is a post hoc analysis with the block length fixed before examining its intervals.

Table A1: Paired Testing cost differences (local baseline minus base DIHAS, bp). Intervals resample five-day blocks after averaging the five fixed seeds. Positive values favor DIHAS; intervals are marginal and conditional on the trained policies.

Year	Algorithm	Mean difference	95% date-block interval
2019	DDQN	5.298	[5.034, 5.546]
2019	PPO	7.338	[6.753, 7.894]
2020	DDQN	3.166	[2.719, 3.575]
2020	PPO	5.873	[5.471, 6.243]
2021	DDQN	5.157	[4.740, 5.511]
2021	PPO	6.674	[5.653, 7.564]
Equal-year	DDQN	4.540	[4.328, 4.740]
Equal-year	PPO	6.628	[6.215, 7.012]

To determine how broadly the mean cost advantage extends across evaluated orders, we next examine matched-order coverage. Figure 9 expands the base DDQN/PPO comparison to matched Testing orders. Each seed uses the identical ordered evaluation manifest for DIHAS and its local baseline; order counts reproduce the recorded weighted Testing mean difference. All 30 year–algorithm–seed contrasts have positive mean savings, and the weighted fraction of episodes with lower DIHAS cost ranges from 59.3% to 88.4%. The same orders recur across seeds, so the points do not provide an independent-order confidence interval.

Table A2: Per-year Testing TC in bp: five-seed mean (sample SD) for learned policies; deterministic rules have no seed SD. Lower is better.

Method	2019	2020	2021
DIHAS + DDQN	−4.140 (0.068)	−1.532 (0.137)	−0.780 (0.327)
DIHAS + DDQN + CASH	−3.972 (0.174)	−1.550 (0.399)	−0.621 (0.303)
DIHAS + DDQN + CATE	−4.005 (0.458)	−1.652 (0.315)	−0.616 (0.240)
DQN baseline	1.158 (3.563)	1.633 (1.294)	4.377 (1.525)
DIHAS + PPO	−4.184 (0.172)	−1.886 (0.264)	−0.670 (0.201)
DIHAS + PPO + CASH	−4.096 (0.292)	−1.533 (0.238)	−0.853 (0.522)
DIHAS + PPO + CATE	−3.665 (0.462)	−1.604 (0.334)	−0.441 (0.563)
PPO baseline	3.154 (1.532)	3.987 (2.244)	6.004 (1.126)
DIHAS + GRPO	−4.222 (0.191)	−2.121 (0.315)	−1.131 (0.281)
DIHAS + GRPO + CASH	−4.297 (0.149)	−2.096 (0.260)	−1.074 (0.195)
DIHAS + GRPO + CATE	−4.279 (0.178)	−1.839 (0.329)	−1.206 (0.526)
OPD baseline (Fang et al., 2021)	8.836 (0.359)	10.582 (0.829)	8.362 (1.627)
TWAP (Zhang et al., 2023)	8.221	8.407	6.246
Momentum	9.262	10.860	8.125

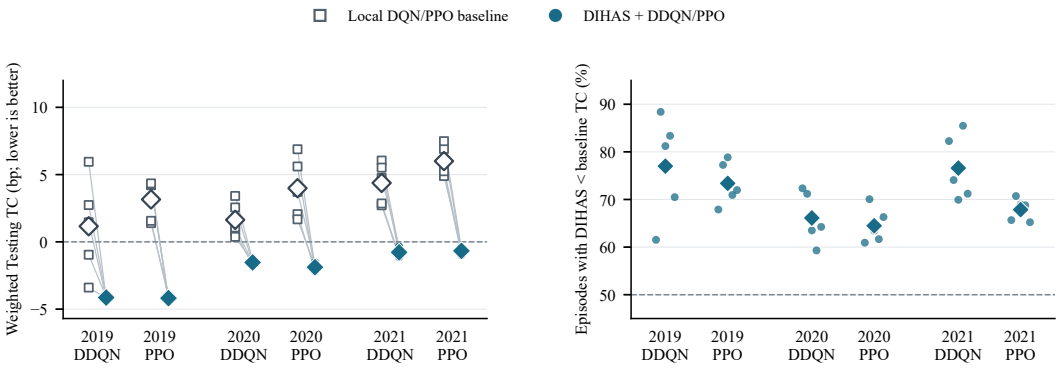


Figure 9: Matched-order Testing comparison of base algorithms. Left: local DQN/PPO baselines are open squares, DIHAS policies are filled circles; lines join matched seed identifiers and diamonds mark five-seed means. Right: the count-weighted episode share with lower DIHAS TC, shown for each seed.

B Protocol details and baseline adaptations

The shared execution contract does not remove method-specific differences in actions, information, and training work; RQ1 therefore does not isolate the effect of evidence feedback.

B.1 Panel construction and information access

The panel is formed by feed availability before methods are compared. Zhang et al. (2023) describe 100 liquid A-share stocks, while the released CODE_LIST in their implementation contains 98 names. We retain 52 Shenzhen names with both local trade and order feeds in every month of 2019. Of the other 46 names, 43 lack a local order feed and three have unavailable or incomplete trade feeds. The 52-name subset remains fixed in 2020–2021, with valid stock-days screened in each year. Table A3 distinguishes source-pack records from stock-days passing the window-level five-level-book screen. These counts do not equal the number of execution windows, since multiple starting times can belong to one stock-day.

To prevent diagnostic roles from sharing the same fitting dates, Training is partitioned internally. Fitting dates estimate the evidence model and control-variate coefficients; Dev dates support component diagnostics; guard dates check conditional-mean variance; and calibration dates estimate overstatement. These roles are disjoint. External Validation and Testing use their own manifests. A Dev result therefore measures an internally held-out diagnostic, not final out-of-sample execution cost.

Table A3: Stock-day records by year and split. The second group gives the separately recorded valid-book subset; a dash denotes an unlisted count.

Record stage and split	2019	2020	2021
Source pack, total	12,454	12,183	12,178
Source Training	8,279	8,055	8,030
Source Validation	1,955	1,917	1,820
Source Testing	2,220	2,211	2,328
Valid-book Training	—	7,990	6,902
Valid-book Validation	—	1,665	1,544
Valid-book Testing	—	1,934	2,023

B.2 Parent orders and executable actions

For each eligible stock-day, the parent order is fixed using only volume available by the preceding-day cutoff. At decision time the state is $s_t = (x_{t-L+1:t}, I_t/Q_{i,d}, \tau_t, h_t)$, where x is causal order-book context, I_t is remaining inventory, τ_t is remaining time, and h_t records earlier fills. An option $o_t = (\delta_t^p, \delta_t^q, \kappa_t)$ shifts the neutral quote and quantity for $\kappa_t \in \{1, 2, 3\}$ minutes. The 12 legal options span neutral, patient liquidity, patient size-up, catch-up urgency, and impact control. The option remaps its primitive command each minute, falls back to neutral at an invalid shifted index, and advances $\min\{\kappa_t, 30 - t\}$ minutes even after early inventory completion. All systems share depth-limited matching, capped passive fills, terminal completion, and sell-side TC accounting; raw nanosecond feed timestamps do not imply event-level queue simulation.

To connect the action identifiers in the case studies to executable decisions, Table A4 lists the option templates. The quote and quantity shifts are offsets in the primitive-action index grid, not price changes in basis points or fixed share quantities. The neutral command is recomputed from the current book, inventory, and remaining time at each minute. Option 0 is the neutral template, whereas the frozen reference is the reference policy's state-dependent choice; the reference options 6 and 10 in Figure 7 are therefore not alternative names for option 0.

Table A4: Executable option templates. Shifts are relative to the minute-level neutral command; duration is capped by the remaining horizon.

Option ID	δ^p	δ^q	κ (min)	Family
0	0	0	1	Neutral
1	1	0	1	Patient liquidity
2	1	0	2	Patient liquidity
3	2	0	2	Patient liquidity
4	1	1	2	Patient size-up
5	2	1	3	Patient size-up
6	0	1	1	Patient size-up
7	0	1	2	Patient size-up
8	-1	0	1	Catch-up urgency
9	-1	1	2	Catch-up urgency
10	-2	1	2	Catch-up urgency
11	0	-1	1	Impact control

B.3 Baseline adaptations and input checks

To compare available execution systems on the shared task, we retain their distinct action interfaces and document each adaptation. The local DQN and PPO-style Zhang baselines use 99 primitive actions under the same parent-order replay contract (Zhang et al., 2023). Our OPD adaptation trains a future-observing PPO teacher for 120K steps. A causal PPO student, trained for another 120K steps, receives labels at its Training states from the frozen teacher and is the only OPD policy used in Validation and Testing (Fang et al., 2021). The original OPD uses eight 30-minute decisions with action fractions of 0, 25, 50, 75, or 100% of the parent. Our 30 one-minute decisions use $\{0, 0.5, 1, 1.5, 2\}Q/30$ at an ask-one-relative -50 bp limit price and common terminal completion. This task-specific adaptation is neither an exact replication nor compute-matched to a single 120K-step policy. The released TWAP_Agent2 implementation accompanying Zhang et al. (2023) submits one-thirtieth of the parent at the same price offset. Our exploratory momentum rule submits two such

units if the current midpoint is below the midpoint five minutes earlier, and one otherwise. It does not reproduce the incompletely specified Momentum result in Zhang et al. (2023).

Because the OPD teacher observes future information during Training, its separation from the deployed student requires an explicit check. The OPD identity records specify no DIHAS method components and no external model API. Validation selects the causal student; the teacher is not used at Testing. The stored contract audit checks matching for one- and two-unit submissions and perturbs future observations at decision steps 0, 10, and 28 to check that student inference is unchanged. These finite checks support the implemented separation of teacher and student information; they are not an exhaustive proof of the data pipeline.

For OPD Testing, the audit contains 450,195, 362,983, and 390,217 unique windows in 2019, 2020, and 2021, representing 2,220,000, 1,806,203, and 1,928,728 weighted episode occurrences. Reconstructing costs with these repeat counts reproduces the stored student summaries. The distinction between windows, weighted occurrences, and stock-days is essential when checking the denominator of a result.

B.4 Evaluation denominators

One stock-day can contribute many overlapping 30-minute windows. Table A5 distinguishes source parent-order stock-days from stock-days represented in the final manifest, unique windows, and weighted evaluation occurrences. The source-parent counts precede final window validation. The represented/source ratios therefore describe retained stock-days, not the fraction of valid book observations. Windows that lack the required valid lookback or execution book are excluded; the retained manifests do not provide an exhaustive reason-by-reason rejection ledger. In particular, retaining all 2019 source stock-days does not imply that every possible window is valid.

Table A5: Final evaluation manifests. Source and retained counts refer to stock-days; window and occurrence counts are distinct denominators. Repeats restore evaluation frequency rather than order-size weights.

Year	Split	Dates	Source	Retained	Windows	Occurrences
2019	Training	162	8,279	8,279	127,482	132,464
2019	Validation	38	1,955	1,955	30,164	31,280
2019	Testing	43	2,220	2,220	450,195	2,220,000
2020	Training	160	8,055	7,989	122,182	126,697
2020	Validation	38	1,917	1,650	24,474	25,391
2020	Testing	44	2,211	1,934	362,983	1,806,203
2021	Training	161	8,030	6,842	100,581	104,401
2021	Validation	36	1,820	1,536	22,815	23,698
2021	Testing	45	2,328	2,023	390,217	1,928,728

B.5 Archived evidence-off comparisons and checkpoint schedules

To examine the closest available evidence-on/off comparison, we inspect the archived training schedules as well as the checkpoint step counts. The 2020 archive contains three seeds per algorithm for evidence-on and evidence-off policies. The latter disables the evidence bias and its unused fitting/PICV work while retaining the 12-option interface and the terminal paired verifier reward. Checkpoint hashes agree with the evaluation records, and saved effective configurations differ only in the evidence and PICV enable flags. However, all nine Full checkpoints and the first two evidence-off seeds per algorithm were captured at 120K steps from runs planned for 480K; the third evidence-off seed was trained independently to 120K. DDQN exploration uses the ratio of elapsed to planned steps, so equality of observed steps alone does not align its exploration schedule. The three-seed comparison is therefore an archived sensitivity check, not the missing fully matched baseline.

B.6 Common implementation settings

The positive-saving head uses the label $Y > 0.01$ bp; ensemble disagreement measures state–action model uncertainty. The raw $-$TC ledger is retained separately from the transformed terminal training reward for audit.

Table A6: Archived 2020 evidence-off minus Full Testing TC (bp). Negative values favor evidence-off. All checkpoints have 120K observed steps. Seed suffix 03 has different planned horizons; the other pairs share a 480K planned horizon. These are descriptive subset results.

Algorithm	Seed 01	Seed 02	Seed 03	Mean (paired SD)
DDQN	-0.433	-0.390	+0.020	-0.268 (0.250)
PPO	-0.941	+0.368	+0.017	-0.185 (0.678)
GRPO	-0.147	+0.441	+0.187	+0.160 (0.295)

Following the two routes in Figure 2, local paired saving supervises the evidence estimator, while terminal paired saving supplies the verified policy-training reward. PICV adjusts the conditional-mean target only; quantile and positive-saving heads retain the original paired outcomes. The bounded evidence offset modifies legal-action scores without replacing the native RL objective. The calibration and routing diagnostics do not remove otherwise legal options in the reported continuous-offset configuration, and the router offset is zero.

For a consistent interpretation of the scenario comparison, the learned interpolation grid must be distinguished from the replay evaluation nodes. Table A7 records both grids alongside the DIHAS learning settings. Both AA50 and CENTER0 consume the same three replay nodes, with the aggregation operator changed; this prevents attributing their contrast to a different number of scenario evaluations. A primitive-step budget does not include all counterfactual replay work and should not be interpreted as equal total compute across systems.

Table A7: Settings for the reported DIHAS runs and diagnostics. OPD has separate teacher and student budgets.

Setting	Value
RL primitive steps; fixed seeds	120,000; 20230901–20230905
Hidden width; evidence ensemble size	128; 3
Learning rate; discount	3×10^{-4}; 1
Batch size; update epochs	256; 4
Episodes per training block	64
PPO clip; GRPO group size	0.2; 4
Evidence refresh interval	20 blocks
Refresh states; draws per state	32; 4
Evidence fitting epochs; buffer capacity	4; 32,768
RL replay capacity	32,768
Evidence offset bound; scale	1; 0.05
Uncertainty temperature	0.5 bp
Downside; ensemble-disagreement weights	0.15; 0.1
Router offset scale	0
PPO/GRPO uniform exploration mixture	0.05 during training
Guard states; replay draws	24; 64
Calibration states; replay draws	128; 8
Mean-head interpolation knots	$-2, -1, 0, 1, 2$
AA50 audit nodes (fit-scale units)	$-0.5, 0, 0.5$
AA50 radius; CENTER0 radius	$0.5 s_{\text{fit}}$; 0
Terminal verifier scale	5 bp, with tanh transform
Fixed-checkpoint audit states; draws per node	128; 16
PICV guard bootstrap replicates; sampling unit	1,000; trading-date clusters
Fixed-checkpoint audit replicates; block length	1,000; 5 trading days
RQ1 date-bootstrap replicates; block length	4,000; 5 trading days

To preserve the behavior-policy probabilities used by PPO and GRPO, training mixes the evidence-biased softmax with a uniform distribution over legal options at weight 0.05. Update epochs reconstruct the same mixture using the stored, detached offsets and legal masks. This setting is distinct from DDQN's schedule-dependent exploration discussed above and is disabled for greedy evaluation.

To distinguish uncertainty sources, the online PICV guard resamples trading-date clusters, whereas the fixed-checkpoint scenario audit uses circular blocks of five trading days. The separate RQ1 post hoc analysis also uses five-day blocks, but with 4,000 rather than 1,000 replicates. These procedures

are not interchangeable with the sample SD across training seeds or with the 16-draw replay standard errors in the selected cases.

B.7 Compute accounting and missing measurements

To assess the work beyond policy learning, we separate the nominal primitive-step budget from evidence refresh, paired counterfactual rollouts, and the terminal reference verifier. Recorded run summaries provide counterfactual query/step counts and elapsed, RL, and refresh times where available. The source package exports these fields per run and reports the number of available measurements for each summary; imported milestones lacking complete timing records remain missing. The hardware and concurrency differ across runs, and neither comparable device-hours nor peak memory is available for the full panel. Accordingly, these records describe computational work without supporting an efficiency ranking. OPD additionally trains its teacher for 120K steps before the 120K-step student.

C Dev ablations and mechanism audits

C.1 Contrasts and diagnostic definitions

To trace the link between local evidence and policy cost, we separate two types of Dev comparison. The removal runs train new policies, so their contrasts include changed checkpoints, reference decisions, and visited states. A same-checkpoint audit instead keeps the learned model and state sample fixed while changing AA50 to CENTER0. These experiments answer different questions and must not be pooled as one ablation estimate.

For a replay outcome y and predicted quantile q_τ, pinball loss is $(\tau - \mathbb{1}\{y < q_\tau\})(y - q_\tau)$; it measures the quantile head against original paired outcomes. Residual action regret compares the realized saving of the mean-selected option with the best legal option on each replay draw, including the zero-saving reference. Ambiguity regret compares the selected option with the best conditional-mean saving at each scenario node. Worst-case saving is the selected option's minimum node mean. None of these quantities is the final policy's transaction cost.

C.2 Held-out variance guard (RQ2)

To check the estimator-level claim behind PICV, we examine the held-out guard rather than the cost of a retrained policy. The guard uses the variance ratio for the paired conditional-mean target, with a 95% upper bound from resampling trading-date clusters. Across the 45 stored year–algorithm–seed records, the largest upper bound is 0.9318 (2020 PPO, seed 20230904), consistent with Figure 6. Passing the threshold satisfies the prespecified empirical acceptance rule, not a universal variance bound or a guarantee of better Testing cost. The ablation variance summaries describe independently retrained policies and should not be substituted for this guard statistic.

For a reproducible acceptance decision, the implementation fits one coefficient per option family on fitting dates, using ridge factor 10^{-3} and clipping coefficients to $[-5, 5]$. Illegal and self-reference branches are masked out. On held-out guard dates, the ratio compares sums of per-option replay variances, averaged within dates and then across dates. At least eight distinct date groups are required. The complete coefficient vector is accepted only when the bootstrap 95% upper bound is below one; otherwise it is reset to zero. The recorded mean-shift statistic describes finite-replay correction precision, not a separate empirical proof of unbiasedness. The zero-mean argument instead depends on the predictable-control assumptions in Appendix D.

C.3 Fixed-checkpoint scenario sensitivity (RQ3)

To determine whether the aggregate scenario results conceal seed-level reversals, Figure 10 displays all 45 fixed-checkpoint records. Each comparison holds the checkpoint, reference, legal options, and 128 Dev states fixed, using 16 paired replay draws at each of the three mean nodes. AA50 and CENTER0 differ in their aggregation rule, not their evaluation states or replay budget. The intervals resample the 24 represented trading dates with five-day circular blocks.

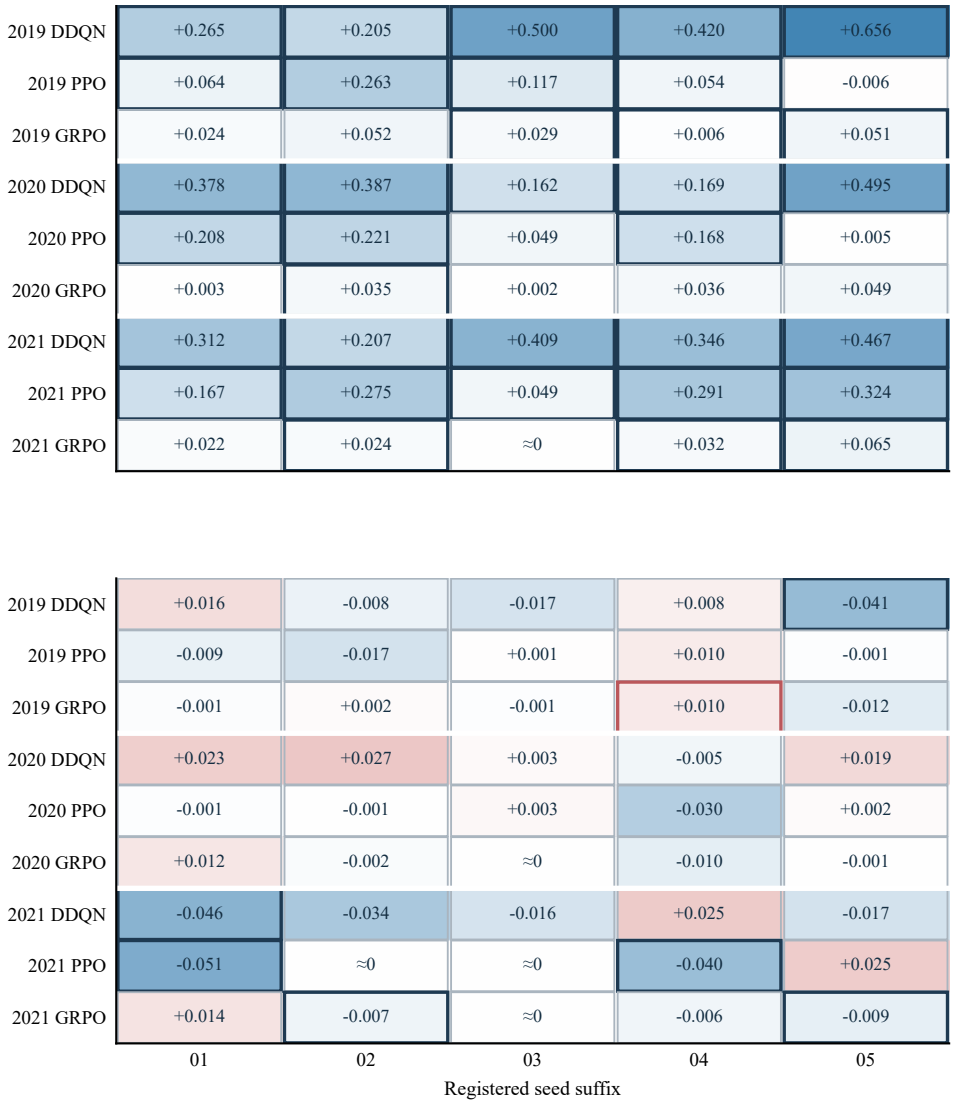


Figure 10: Fixed-checkpoint Dev contrasts, AA50 minus CENTER0 (bp). Upper: worst-node saving, higher is better. Lower: ambiguity regret, lower is better. Columns give seed suffixes 01–05. Blue/red outlines mark favorable/adverse 95% block-bootstrap intervals excluding zero; gray outlines include zero. Colors indicate effect direction and magnitude.

The saving advantage is more consistent than the regret advantage. Five-seed mean worst-node saving improves in all nine year–algorithm cells, but mean regret improves in eight, with 2020 DDQN the exception. Across the 45 individual seed records, regret decreases in 26, increases in 18, and is unchanged in one; only six favorable and one adverse marginal intervals exclude zero. These descriptive counts do not treat overlapping dates or repeated seeds as independent replications and are not adjusted for multiple comparisons. They support sensitivity of local decisions to mean scenarios, not a uniform regret improvement or a Testing-cost benefit.

C.4 Independent retraining and matched-probe interpretation (RQ4)

To examine whether local diagnostic gains translate into policy-level cost, we next compare independently retrained component removals. Figure 11 shows heterogeneous Dev TC changes across years and algorithms. The heatmap remains descriptive; the additional paired-seed summaries quantify training variation separately from date uncertainty.

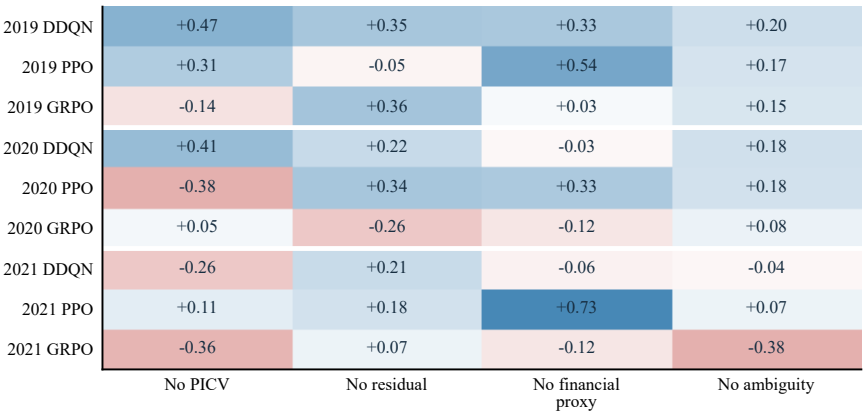


Figure 11: Dev TC after independent component removal. Cells show removed-component minus Full, in bp, for each year and algorithm; positive values favor Full. Values are differences of five-seed means without paired uncertainty intervals.

A matched-probe calculation clarifies the distinction between an operator switch and independent retraining. On a matched 96-state PPO probe, the same-checkpoint operator switch contributes -0.0090 bp; an arithmetic remainder of $+0.1164$ bp encompasses changes to the checkpoint, reference, evidence model, and continuation. Their sum is the retraining contrast, not a causal decomposition of Testing cost.

C.5 Paired-seed uncertainty after component removal

For each of the four removals, we pair Dev cost with Full by year, algorithm, and seed identifier. The following summaries use removal-minus-Full cost, its sample SD across five differences, and an exploratory Student-t interval with four degrees of freedom. These marginal intervals describe training-seed variation under the available probes; they are neither date-block intervals nor an isolated effect at a common checkpoint. No multiple-comparison adjustment is applied. The accompanying numerical records retain all 180 seed differences, including adverse and near-zero cells.

Table A8: 2019 Dev removal-minus-Full TC (bp): paired seed mean, sample SD, and exploratory marginal 95% Student-t interval over five seed pairs.

Algorithm	Removed component	Mean	Paired SD	95% seed interval
DDQN	PICV	+0.472	0.200	[+0.223, +0.721]
DDQN	Residual	+0.347	0.602	[-0.400, +1.095]
DDQN	Financial surrogate	+0.332	0.324	[-0.070, +0.735]
DDQN	Mean aggregation	+0.200	0.330	[-0.210, +0.610]
PPO	PICV	+0.312	0.567	[-0.392, +1.016]
PPO	Residual	-0.054	0.518	[-0.697, +0.589]
PPO	Financial surrogate	+0.542	0.484	[-0.060, +1.143]
PPO	Mean aggregation	+0.166	0.435	[-0.374, +0.706]
GRPO	PICV	-0.141	0.211	[-0.403, +0.122]
GRPO	Residual	+0.359	0.547	[-0.320, +1.038]
GRPO	Financial surrogate	+0.035	0.412	[-0.477, +0.546]
GRPO	Mean aggregation	+0.152	0.274	[-0.187, +0.492]

Table A9: 2020 Dev removal-minus-Full TC (bp): paired seed mean, sample SD, and exploratory marginal 95% Student-t interval over five seed pairs.

Algorithm	Removed component	Mean	Paired SD	95% seed interval
DDQN	PICV	+0.408	0.517	[-0.234, +1.051]
DDQN	Residual	+0.223	0.797	[-0.766, +1.212]
DDQN	Financial surrogate	-0.031	0.850	[-1.087, +1.025]
DDQN	Mean aggregation	+0.183	1.125	[-1.213, +1.580]
PPO	PICV	-0.382	0.937	[-1.545, +0.781]
PPO	Residual	+0.344	0.884	[-0.753, +1.442]
PPO	Financial surrogate	+0.329	0.987	[-0.897, +1.554]
PPO	Mean aggregation	+0.175	0.949	[-1.002, +1.353]
GRPO	PICV	+0.048	1.228	[-1.476, +1.573]
GRPO	Residual	-0.264	0.678	[-1.106, +0.577]
GRPO	Financial surrogate	-0.125	0.761	[-1.070, +0.820]
GRPO	Mean aggregation	+0.077	0.427	[-0.453, +0.606]

Table A10: 2021 Dev removal-minus-Full TC (bp): paired seed mean, sample SD, and exploratory marginal 95% Student-t interval over five seed pairs.

Algorithm	Removed component	Mean	Paired SD	95% seed interval
DDQN	PICV	-0.263	0.703	[-1.136, +0.610]
DDQN	Residual	+0.214	0.576	[-0.502, +0.929]
DDQN	Financial surrogate	-0.064	0.563	[-0.763, +0.634]
DDQN	Mean aggregation	-0.039	0.662	[-0.861, +0.783]
PPO	PICV	+0.107	1.121	[-1.285, +1.499]
PPO	Residual	+0.180	0.737	[-0.735, +1.095]
PPO	Financial surrogate	+0.729	1.098	[-0.634, +2.092]
PPO	Mean aggregation	+0.072	1.138	[-1.340, +1.485]
GRPO	PICV	-0.358	0.461	[-0.931, +0.215]
GRPO	Residual	+0.070	0.511	[-0.564, +0.705]
GRPO	Financial surrogate	-0.122	0.217	[-0.391, +0.148]
GRPO	Mean aggregation	-0.383	0.478	[-0.977, +0.210]

C.6 Selection and interpretation of the two Dev cases

The cases in Figure 7 were selected after the fixed-checkpoint audits were available. Eligible states had an AA50/CENTER0 option switch and a crossing of the independently replayed saving curves between the two mean endpoints. Within the favorable 2021 PPO and adverse 2020 DDQN strata, the selection rule chose the median absolute change in state-level ambiguity regret. There were 46 and 83 eligible states, respectively, among 640 audited states in each stratum. This rule avoids selecting the largest effect, but remains exploratory because the strata and rule were chosen after observing the audits.

The 2021 PPO case uses seed 20230904 and state index 43: AA50 selects option 6, CENTER0 selects option 3, and AA50-minus-CENTER0 regret is -0.1615 bp. The 2020 DDQN case uses seed 20230902 and state index 54: the selected options are 10 and 5, and the regret difference is $+0.1538$ bp. In both cases AA50 selects the frozen reference, so its paired saving is identically zero. The curves show separate replay estimates of the selected options, not the model scores used to select them; CENTER0 therefore need not have the highest plotted center estimate. This distinction explains why a selected option can look unfavorable even at the center in the diagnostic replay.

To interpret the two selected-option curves in the context of available alternatives, panels c and d show all 12 options on the same mean-saving scale. Worst-node saving evaluates the minimum of a selected option's node means, whereas ambiguity regret measures its gap to the best legal option at each node before averaging. In the adverse case, retaining the zero-saving reference protects the worst node but forgoes option 5's gains at the negative and central nodes. The different objectives explain why better worst-node saving need not imply lower regret; neither objective measures the retrained policy's TC.

For each plotted node, the error bar is the sample SD of 16 paired draws divided by $\sqrt{16}$. It quantifies replay noise for one state, not uncertainty across trading dates. The records do not include complete minute-by-minute inventory paths, so no such trajectory is inferred from the option identifiers. These

cases support a possible ranking mechanism and an adverse outcome, while a full radius sweep, an exposure-stratified test, and a placebo remain unperformed. In particular, no spread-, depth-, volatility-, or urgency-stratified result is inferred from these two cases. As in the main text, the resulting boundary concerns evidence under exogenous historical replay, not the response of a live market to the strategy.

D Conditional identities for paired evidence

These derivations state when the estimator has the properties used in the method; they do not assert that the historical simulator identifies live-market causal effects.

D.1 Shared innovations preserve the specified mean difference

Fix state, order, reference policy, and conditional mean. Write the simulated reference and candidate costs as $X(\omega)$ and $Z(\omega)$, with finite second moments. Coupling the two branches through the same innovation draw preserves their marginal distributions under the replay protocol. Consequently,

$$\mathbb{E}[X - Z] = \mathbb{E}[X] - \mathbb{E}[Z], \qquad \operatorname{Var}(X - Z) = \operatorname{Var}(X) + \operatorname{Var}(Z) - 2\operatorname{Cov}(X, Z).$$

The identity follows from linearity of expectation and expansion of the centered square. Relative to independent draws with the same marginals, shared innovations reduce difference variance when their induced covariance is positive; they do not guarantee variance reduction for every pair of options. Replacing the reference action by itself gives zero saving pathwise when the simulator and continuation are identical.

D.2 Predictable inventory control and its variance condition

Condition on the state, scenario mean, frozen reference, and fitting information. Assume that the finite-horizon innovations satisfy $\mathbb{E}[\xi_j \mid \mathcal{F}_{j-1}] = 0$, that ΔI_j is measurable before innovation j, and that the products are integrable. With s_{fit} fixed by the fitting sample, the tower property gives

$$\mathbb{E}[h] = s_{\text{fit}} \sum_j \mathbb{E}[\Delta I_j \mathbb{E}[\xi_j \mid \mathcal{F}_{j-1}]] = 0.$$

For a coefficient β fixed before the evaluation draws, $\mathbb{E}[Y - \beta h] = \mathbb{E}[Y]$. Under finite second moments,

$$\operatorname{Var}(Y - \beta h) = \operatorname{Var}(Y) - 2\beta \operatorname{Cov}(Y, h) + \beta^2 \operatorname{Var}(h).$$

If $\operatorname{Var}(h) > 0$, minimizing this quadratic yields $\beta^* = \operatorname{Cov}(Y, h)/\operatorname{Var}(h)$. An estimated coefficient need not achieve this reduction, motivating the held-out variance guard. If a fitted coefficient or its acceptance decision uses the same evaluation draws, the fixed-coefficient expectation argument does not automatically apply. Likewise, a misspecified conditional innovation mean invalidates the zero-mean identity. The reported family-level fit and separate guard are operational checks under these assumptions, not a proof of robustness to simulator misspecification.

D.3 Mean sensitivity of the financial surrogate

To clarify how a mean scenario can change an option ranking, consider the financial surrogate for two legal options. Their difference is

$$f_{\text{fin}}(s, o_1, \delta\mu) - f_{\text{fin}}(s, o_2, \delta\mu) = \Delta A + \Delta B\, \delta\mu,$$

where $\Delta A = A_\theta(s, o_1) - A_\theta(s, o_2)$ and similarly for ΔB. If $\Delta B \neq 0$, its ranking changes at $\delta\mu = -\Delta A/\Delta B$ when this point lies inside the scenario interval. If $\Delta B = 0$, the surrogate difference is constant across means. A scenario-dependent residual can still change the complete estimator's ranking, so the case plot alone does not identify inventory exposure as the unique cause of a switch. Also, minimizing a noisy mean estimate over scenarios is nonlinear: unbiased node means do not imply an unbiased minimum or a confidence guarantee for AA50.

E Reproduction scope

The paper and appendices describe the replay protocol, algorithm interfaces, implementation settings, and statistical summaries used in the reported experiments. These descriptions do not constitute a complete executable release. Rerunning the experiments requires access to the underlying market data, split and evaluation manifests, implementation code, model checkpoints, and training environment; these are not included in the manuscript PDF.

A reproduction should proceed in the following order:

1. Fix the stock universe, chronological splits, parent-order rule, matcher, and recorded repeat weights before comparing methods.
2. Train policies with the specified action interfaces and budgets. Keep OPD future features restricted to the teacher during training and evaluate only its causal student.
3. Freeze the selected checkpoints, evaluate the same manifests, and restore recorded window multiplicities before forming annual costs and equal-year seed summaries.
4. Run component removals separately from fixed-checkpoint diagnostics. Preserve the fit, Dev, guard, and calibration partitions and the scenario nodes used in each audit.
5. Recompute the reported summaries from the resulting evaluation records, preserving units, difference directions, and the distinction between seed SD, replay standard error, and trading-day block intervals.

Access to the underlying market data and permission to redistribute them must be established separately; neither is implied by the results reported here.