DRESS: AUDITING REWARD-PRESERVING SAFETY-COST AMPLIFICATION IN OFFLINE RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline safe reinforcement learning is commonly evaluated under nominal dynamics and monitored through task return or binary constraint violations. Such evaluation can miss a critical deployment failure mode: a small environment shift may preserve task performance while substantially amplifying safety cost. We introduce DRESS (Directional Reward-preserving Environment Shift Screening), a post-training framework that converts this failure mode into a statistically controlled audit. DRESS retains perturbation direction to prevent cancellation, uses paired nominal and shifted rollouts, and tests cost increase jointly with reward noninferiority. A discovery-confirmation split, hierarchical effect intervals, exact training-seed tests, and family-wise correction separate candidate selection from reproducible evidence while respecting the hierarchical structure of RL evaluation. Experiments across multiple offline safe RL algorithms, tasks, and deployment shifts reveal a previously hidden delay-induced safety-cost amplification that is invisible to a saturated binary violation metric and persists on unseen policy checkpoints. Broader stress tests and nominal-placebo analyses further demonstrate that DRESS produces selective, well-calibrated conclusions, while its adaptive acquisition strategy improves screening efficiency under limited evaluation budgets.

Keywords: offline safe RL; distribution shift; policy auditing; statistical evaluation.

1 INTRODUCTION

Offline reinforcement learning (RL) can learn a policy without online exploration, and constrained formulations can explicitly trade task return against safety cost. These properties make offline safe RL attractive when unsafe data collection is expensive. Yet deployment rarely reproduces the training simulator exactly: sensing is noisy, actuators are delayed, and physical parameters drift. A policy that appears safe under the nominal environment may consequently fail under a small shift.

A particularly consequential failure mode arises when task performance remains operationally acceptable while constraint cost increases. We term this phenomenon *reward-preserving safety-cost amplification*. Reward-only monitoring overlooks this regime, and symmetric aggregation can conceal it when opposite perturbation directions induce opposing cost responses. Existing offline-safe-RL benchmarks standardize datasets and nominal evaluation (Liu et al., 2023d; Ji et al., 2023); deployment assessment additionally requires directional analysis and post-selection statistical control.

We address this gap with DRESS-Directional Reward-preserving Environment Shift Screening-a post-training framework that evaluates interpretable shifts on frozen policies with episode-level pairing, separates discovery from confirmation, and tests increased cost jointly with reward noninferiority. Retaining perturbation direction eliminates symmetric cancellation, while training-run variation supplies the upper-level replication for reproducibility analysis.

Our main contributions are:

- We formulate reward-preserving cost amplification as a directional, dual-criterion testing problem and distinguish newly induced limit crossings from severity amplification in policies that are already unsafe nominally.
- We develop a two-level inferential design: paired hierarchical intervals quantify mean effects, while exact training-seed tests and Holm correction assess directional reproducibility without episode-level pseudo-replication.

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.

- We identify a reward-preserving DroneRun cost amplification and reproduce its positive mean effect on five unseen checkpoints, including under zero reward tolerance. A 5,400-rollout task and severity expansion maps the effect's generalization boundary.
- We benchmark the optional sequential acquisition rule against equal-budget grid and random screening, and evaluate calibration with a 10,000-repeat nominal placebo. This separates the core audit from the acquisition heuristic and makes its empirical scope explicit.

2 Related Work

Offline and safe RL. Offline RL learns from a fixed dataset and must control extrapolation beyond its behavioral support. BCQ restricts selected actions toward the dataset (Fujimoto et al., 2019), while CQL regularizes learned values conservatively (Kumar et al., 2020). Safe RL casts the problem as a constrained Markov decision process (CMDP) (Altman, 1999); CPO is a prominent online constrained method (Achiam et al., 2017). For offline safety, CPQ penalizes out-of-distribution actions in the cost critic (Xu et al., 2022), and CDT uses sequence modeling to target reward-cost trade-offs (Liu et al., 2023a). DSRL/OSRL supplies datasets, implementations, and benchmarks for 38 offline-safe-RL tasks (Liu et al., 2023d). DRESS complements these learning methods at the evaluation layer by auditing their frozen policies after training.

Robustness and environment shift. Robust MDPs optimize against uncertainty sets (Iyengar, 2005), while adversarial perturbations probe state- or action-level failures (Huang et al., 2017). Safe-RL attacks can explicitly maximize cost or preserve reward while inducing unsafe behavior (Liu et al., 2023b); robust training with benign off-policy data addresses the corresponding defense problem (Liu et al., 2023c). Offline-RL action perturbations provide a non-adversarial deployment view (Ayabe et al., 2025). Complementing training, attack, and defense, DRESS evaluates frozen checkpoints under interpretable shifts, separates selection from confirmation, and distinguishes newly induced limit crossings from severity amplification in an already-unsafe regime.

Evaluation and statistical reliability. Deep-RL conclusions are sensitive to seeds, metrics, and experimental design (Henderson et al., 2018; Jordan et al., 2020; Patterson et al., 2024); power analyses emphasize training-run replication (Colas et al., 2018). Stratified bootstrap intervals and performance profiles improve reporting across tasks (Agarwal et al., 2021). We combine paired rollouts and hierarchical intervals for effect estimation with a training-seed exact sign test for directional reproducibility; Holm's procedure controls the prespecified family (Holm, 1979). In contrast to benchmark ranking, the decision criterion is conjunctive: cost increases while reward remains noninferior.

DRESS integrates these strands around a distinct decision object: directional, reward-noninferior cost amplification. It separates candidate selection, mean-effect qualification, and exact directional confirmation to provide a deployment-focused assessment of frozen policies.

3 Problem Formulation

Consider a finite-horizon CMDP $\mathcal{M}_\theta = (\mathcal{S}, \mathcal{A}, P_\theta, r, c, H)$, where $\theta = 0$ denotes the nominal environment and $\theta \in \Theta$ denotes a deployment shift. A fixed offline policy π is learned before our audit. Its expected return and cumulative safety cost are

$$R_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{H-1} r(s_t, a_t)\right],\ C_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{H-1} c(s_t, a_t)\right]. \tag{1}$$

For a directional candidate θ, define

$$\Delta R_\theta = R_\theta(\pi) - R_0(\pi),\ \Delta C_\theta = C_\theta(\pi) - C_0(\pi). \tag{2}$$

The nominal and shifted rollouts use the same episode seed. Thus each empirical observation is a within-seed difference, not the difference between two independently sampled means.

Let $\epsilon_R > 0$ be a frozen operational tolerance. We call θ a reward-preserving safety-cost amplification when

$$\Delta C_\theta > 0 \text{ and } \Delta R_\theta \geq -\epsilon_R. \tag{3}$$

The first condition identifies increased safety cost. The second is a noninferiority condition: reward loss, if any, is smaller than the agreed tolerance. Our experiments set ϵ_R to 5% of the absolute nominal return for each task-policy group, before inspecting shifted final results.

This definition is deliberately directional. For example, mass multipliers 0.9 and 1.1 are distinct candidates. Direction-specific analysis preserves harmful responses that a symmetric global average could cancel. Let κ be the task cost limit and C_0, C_θ the corresponding nominal and shifted mean costs. An amplification is a *limit-crossing failure* when $C_0 \leq \kappa < C_\theta$, and an *already-unsafe amplification* when $C_0 > \kappa$. We further distinguish *joint degradation* ($\Delta C > 0$ with material reward loss), *trade-off* (cost decreases with reward), and statistically uncertain cost effects under Eq. (3).

The audit takes a fixed collection of policies, checkpoints, shift candidates, training runs, and paired episode seeds. It returns the candidates for which both components of Eq. (3) are statistically supported, together with their limit-crossing category and the outcomes for all remaining candidates. The resulting claims are indexed explicitly by task, policy, and shift, yielding a precise empirical characterization of deployment robustness.

4 DRESS: Directional Reward-Preserving Environment Shift Screening

4.1 Design objective and scope

DRESS operates after training on frozen policies. Its input is a task e, an algorithm a, independently trained checkpoints $\{\pi_{a,j}\}_{j=1}^{J}$, and a finite panel Θ_e of deployment shifts supported by the environment. Its output is an auditable set of task-policy-shift claims, each labeled as confirmed, interval-qualified, reward-degrading, cost-inconclusive, or unavailable. The design targets four failure modes of common robustness evaluation:

1. **Reward-only blindness:** a policy can retain return while incurring more constraint cost.
2. **Directional cancellation:** low and high parameter changes can have opposite effects, so a symmetric mean can hide the harmful direction.
3. **Selection bias:** the largest discovery effect is optimistically biased if the same rollouts also supply its confidence interval.
4. **Pseudo-replication:** many environment seeds do not replace independently trained policy seeds.

The audit assumes access to a nominal simulator and interpretable shift wrappers. The offline dataset may contain only nominal transitions, and the policy remains frozen throughout evaluation. This setting corresponds to pre-deployment simulation or digital-twin testing with fixed policy, reward, cost, and shift definitions.

4.2 Directional candidate representation

A candidate is a tuple

$$\theta = (f, \lambda, d, w),\ \theta \in \Theta_e, \tag{4}$$

where f is the shift family, λ is its magnitude, d is an optional direction, and w identifies its deterministic implementation. Examples are $(\text{mass}, 0.1, \text{high}, w_{\text{mass}})$ and $(\text{delay}, 1, \varnothing, w_{\text{delay}})$. This representation uniquely identifies a candidate throughout evaluation and analysis.

We use an ordinal within-family severity coordinate $q_f(\theta) \in \{1, 2, 3\}$ for visualization and budget allocation. Because "1% observation noise" and "one delayed action" lack a task-independent physical distance, cross-family comparisons are performed directly in measured reward-cost effect space. Opposite directions remain separate elements of Θ_e throughout selection, inference, and reporting.

For each environment, a capability matrix verifies that w modifies the intended simulator quantity. Supported candidates enter the evaluation panel, and unavailable combinations remain explicitly marked. Step-level traces record the applied transformation, including delayed-action indices, noise scales, and dynamics multipliers, thereby excluding ineffective perturbations from the analysis.

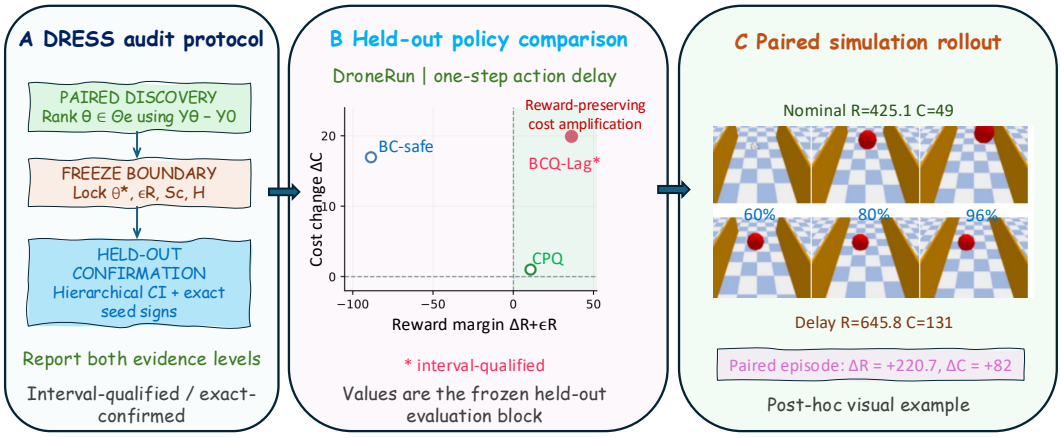

Figure 1: **DRESS overview and empirical evidence. Left:** paired discovery and independent held-out confirmation. **Center:** the held-out DroneRun one-step-delay comparison across the three audited policies; the dagger denotes interval qualification and exact family values quantify training-run consistency. **Right:** aligned nominal and shifted frames from the BCQ-Lag episode with the largest post-hoc paired cost increase, providing a qualitative view complementary to the quantitative analysis.

Fig. 1 summarizes DRESS and relates its inferential stages to aggregate held-out results and an illustrative paired rollout. Candidate identity, checkpoint set, tolerance, episode-seed blocks, and hypothesis family are fixed before confirmation outcomes are observed.

4.3 Paired Rollouts and Validity Conditions

For checkpoint j, episode seed i, metric $m \in \{C, R, V\}$, and candidate θ, let $Y_{ji}^{m,\theta}$ be the shifted episodic observation and $Y_{ji}^{m,0}$ its nominal counterpart. DRESS stores the paired difference

$$d_{ji}^{m,\theta} = Y_{ji}^{m,\theta} - Y_{ji}^{m,0}. \tag{5}$$

The same episode seed is used on both sides, aligning the initial-state and simulator random-number streams wherever the wrapper permits. Pairing reduces variance when the two outcomes are positively correlated:

$$\mathrm{Var}(d^{m,\theta}) = \mathrm{Var}(Y^{m,\theta}) + \mathrm{Var}(Y^{m,0}) - 2\mathrm{Cov}(Y^{m,\theta}, Y^{m,0}). \tag{6}$$

More importantly, it makes the estimand an episode-level effect under matched exogenous randomness.

Validity requires nominal and shifted observations to match in task, algorithm, training run, checkpoint identity, cost limit, and ordered episode seeds. Each observation includes the candidate specification and per-episode return, cost, and violation outcome, with unique seeds drawn from the designated evaluation split. Enforcing these conditions preserves the matched estimand throughout analysis.

4.4 Effect Estimand and Operational Margin

For metric m, we estimate an equal-checkpoint average

$$\widehat{\Delta m}_\theta = \frac{1}{J}\sum_{j=1}^{J}\left(\frac{1}{n_j}\sum_{i=1}^{n_j} d_{ji}^{m,\theta}\right). \tag{7}$$

Equal weighting prevents a checkpoint with more successfully logged episodes from dominating the policy-level claim. In the balanced experiments $n_j = n$, but Eq. (7) defines behavior under incomplete cells as well.

Reward noninferiority is specified through an operational margin

$$\epsilon_R = \eta \left| \frac{1}{J} \sum_{j=1}^{J} \bar{R}_j^0 \right|, \ \eta = 0.05, \tag{8}$$

where $\bar{R}_j^0$ is nominal mean return for checkpoint j. The candidate margin vector is

$$\widehat{\boldsymbol{g}}_\theta = \left(\widehat{\Delta C}_\theta, \widehat{\Delta R}_\theta + \epsilon_R \right). \tag{9}$$

A reward-preserving cost amplification lies in the positive orthant of this two-dimensional margin space. Fixing η before shifted final outcomes is crucial: changing it after observing $\widehat{\Delta R}$ would redefine which effect was sought. We nevertheless report a prespecified sensitivity grid after the primary analysis so readers can see the dependence on this operational choice.

4.5 Hierarchical paired bootstrap

Our uncertainty target includes policy-training randomness and environment randomness. For bootstrap replicate b, DRESS samples J checkpoint indices with replacement. For each selected checkpoint j^*, it independently resamples n_{j^*} paired differences from $\{d_{j^*i}^{m,\theta}\}_{i=1}^{n_{j^*}}$ and takes their mean. Averaging the J resampled checkpoint means yields $\widehat{\Delta m}_\theta^{*(b)}$. The 2.5% and 97.5% empirical quantiles over $B = 10{,}000$ replicates form the reported interval.

This procedure preserves within-episode pairing and treats the training seed as the sampling cluster. The two-stage construction captures both within-checkpoint evaluation uncertainty and between-checkpoint training variation while retaining the equal-checkpoint estimand in Eq. (7).

For the one-sided null $H_0 : \Delta m \le m_0$, we center the bootstrap distribution at the observed estimate and compute

$$p_m = \frac{1 + \sum_{b=1}^{B} \mathbb{1}\left[\widehat{\Delta m}_\theta^{*(b)} - \widehat{\Delta m}_\theta \ge \widehat{\Delta m}_\theta - m_0 \right]}{B + 1}. \tag{10}$$

The $+1$ correction avoids a zero Monte Carlo value. With $B = 10{,}000$, the minimum attainable value is $1/10{,}001$; training-seed resolution remains governed by the number of independent checkpoints.

Training-seed directional confirmation. Bootstrap intervals estimate the equal-checkpoint mean and retain episode-level uncertainty, whereas confirmatory resolution is determined by the number of independent training runs. We therefore reduce each checkpoint to its paired mean and count positive directions,

$$S_m = \sum_{j=1}^{J} \mathbb{1}[\bar{d}_j^m - m_0 > 0], \ p_m^{\text{sign}} = 2^{-J} \sum_{k=S_m}^{J} \binom{J}{k}, \tag{11}$$

where $m_0 = 0$ for cost and $m_0 = -\epsilon_R$ for return; ties count as nonpositive. Under independent training runs and the directional null $H_m^{\text{sign}} : \Pr(\bar{d}_j^m - m_0 > 0) \le 1/2$, this is an exact one-sided binomial test. The test quantifies directional reproducibility across training runs, while hierarchical intervals estimate the equal-checkpoint mean. Formal confirmation evaluates the conjunction of the two directional sign alternatives and applies Holm correction to multi-candidate families.

4.6 Discovery acquisition and candidate freezing

Discovery can use a deterministic grid, uniform random allocation, or an optional sequential heuristic. Let n_θ paired episodes have been observed for candidate θ, with sample means $\bar{d}_\theta^C, \bar{d}_\theta^R$ and standard errors $s_\theta^C/\sqrt{n_\theta}, s_\theta^R/\sqrt{n_\theta}$. The heuristic scores

$$U_C(\theta) = \bar{d}_\theta^C + 1.96 \frac{s_\theta^C}{\sqrt{n_\theta}}, \tag{12}$$

$$P_R(\theta) = \Phi\left(\frac{\bar{d}_\theta^R + \epsilon_R}{s_\theta^R/\sqrt{n_\theta}} \right), \tag{13}$$

$$A(\theta) = U_C(\theta) P_R(\theta), \tag{14}$$

where Φ is the standard normal CDF. U_C favors potentially large cost increases; P_R softly discounts candidates likely to violate reward noninferiority. Zero standard error maps deterministically to feasibility 0 or 1, while a one-sample standard error maps to 0.5.

Every candidate receives one randomly ordered batch before adaptive allocation. Thereafter the next batch goes to the available candidate with maximal $A(\theta)$, with seeded random tie-breaking. At each budget checkpoint, the currently maximal candidate is frozen. Eq. (14) defines a lightweight ranking heuristic that prioritizes large, reward-feasible cost effects. The equal-budget benchmark in Sec. 5 evaluates its finite-budget behavior independently of the inferential procedure.

4.7 Independent dual qualification, confirmation, and multiplicity

Discovery rollouts rank and freeze candidates; confirmatory labels are assigned exclusively from a disjoint evaluation range containing at least 20 paired episodes per checkpoint. Let $L_C(\theta)$ and $L_R(\theta)$ be the hierarchical 95% lower bounds. DRESS labels a candidate *interval-qualified* when

$$L_C(\theta) > 0 \ \wedge \ L_R(\theta) + \epsilon_R \geq 0. \tag{15}$$

The component nulls are $H_C : \Delta C \leq 0$ and $H_R : \Delta R < -\epsilon_R$. Because the amplification alternative is their intersection, the valid intersection-union value is

$$p_{\text{joint}}(\theta) = \max\{p_C(\theta), p_R(\theta)\}. \tag{16}$$

Eq. (16) is formed separately from bootstrap component values for descriptive interval qualification and from exact sign-test component values for training-run confirmation. For a prespecified family of M candidates, DRESS applies Holm step-down adjustment to the M exact joint values. If $p_{(1)} \leq \cdots \leq p_{(M)}$ are ordered, hypothesis (k) is rejected only while $p_{(k)} \leq \alpha/(M-k+1)$. A separately specified targeted confirmation has $M = 1$. This controls family-wise error without assuming independence among candidates.

The end-to-end procedure, including validation failures and sample-split control flow, is given as Algorithm 1 in Appendix A.1.

4.8 Confirmatory modes and complexity

The held-out AntRun/DroneRun study uses a strict final-panel mode: tasks, policies, all three candidates, episode seeds, and the 18-test family are frozen before opening any shifted result. Thus it evaluates every panel member without outcome-dependent selection. The three-seed panel supports broad effect estimation and candidate nomination. The outcome-blind five-seed replication excludes the nomination checkpoints and evaluates reproduction of both the selected mean effect and checkpoint-level directions, providing an independent effect estimate and a direct assessment of cross-checkpoint consistency.

For each candidate, DRESS reports point estimates and intervals for cost, return, and violation, together with the reward tolerance, component and joint p values, inferential status, and limit-crossing category. The resulting evidence profile distinguishes reward degradation, uncertain cost responses, and reward-preserving amplification. Complete family reporting establishes the specificity of each detected effect across policies and shifts.

With $K = |\Theta|$, J checkpoints, n episodes per condition, horizon H, and B bootstrap replicates, rollout cost is $O(KJnH)$ and stored episode summaries are $O(KJn)$. A direct hierarchical implementation costs $O(KBJn)$ time and $O(B+Jn)$ working memory per metric; candidates and metrics can be processed independently. In our experiments rollout generation dominates aggregation.

5 Experiments

Our evaluation addresses five questions. **RQ1**: What nominal reward-safety operating points do the compared offline safe-RL policies provide? **RQ2**: Can a mild deployment shift induce reward-preserving safety-cost amplification on held-out tasks? **RQ3**: How do these effects vary across policies, tasks, training runs, and safety metrics? **RQ4**: Do the observed effects persist across unseen checkpoints, additional tasks, and delay severities? **RQ5**: Does sequential acquisition identify qualifying amplifications more efficiently than grid or random screening?

5.1 TASKS, DATASETS, AND EVALUATION PROTOCOL

We use the DSRL datasets and OSRL implementations (Liu et al., 2023d). AntRun and DroneRun contain 363k and 395k offline transitions, respectively; both use cost limit 10 and horizon 200. Tasks, shifts, and checkpoints were frozen before shifted outcomes were opened. Complete dataset and interface statistics are in Appendix A.2.

Compared policies and fairness. We apply DRESS to BC-Safe, BCQ-Lag, and CPQ (Fujimoto et al., 2019; Xu et al., 2022) under frozen seeds 0/10/20, identical shifts, and paired episode seeds. At the screening layer, equal-budget grid, random, and DRESS acquisition provide the efficiency comparison. Appendix A.2 details both comparison layers.

The final panel comprises the nominal environment, a one-step action delay, observation Gaussian noise at 1% of the training-split feature scale, and action Gaussian noise at 2% of the action scale. Each of the 2 tasks $\times 3$ policies $\times 3$ training runs $\times 4$ conditions is evaluated for 50 deterministic episodes, yielding 72 cells and 3,600 rollouts. Nominal and shifted conditions share episode seeds, and checkpoint identity and applied transformations are verified for every pair.

Effect estimation uses paired changes in episodic cost and return, 10,000 hierarchical-bootstrap replicates, and a 5% nominal-return noninferiority tolerance. Confirmatory inference uses exact one-sided training-seed sign tests and their intersection-union maximum. The final family contains $2 \times 3 \times 3 = 18$ task-policy-shift claims and uses exact Holm correction; the outcome-blind five-seed replication is one separately frozen targeted claim and excludes every nomination checkpoint. Normalized reward, binary violation rate, and upper-tail CVaR_{10} cost provide complementary diagnostic views, with mean cost and return defining qualification.

5.2 RQ1: NOMINAL POLICY OPERATING POINTS

BCQ-Lag has the highest nominal normalized reward on both tasks (0.775 AntRun; 0.730 DroneRun), but also costs 89.2/91.2 and violation rates 0.973/1.000. AntRun CPQ is nearly at the safety floor (cost 0.3, zero violations), but its normalized reward is only 0.029. BC-Safe is intermediate on AntRun and has low mean DroneRun cost 5.5, although its tail cost is 31.8. These are operating points rather than an overall ranking: DRESS compares each shifted policy with its own frozen nominal checkpoint. Appendix A.2 reports dataset interfaces in Table 3, the full nominal statistics in Table 4, and checkpoint-resolved reward-cost operating points in Fig. 4.

5.3 RQ2: HELD-OUT REWARD-PRESERVING COST AMPLIFICATION

Across the 18 held-out comparisons, DRESS isolates one reward-preserving amplification: DroneRun BCQ-Lag under one-step delay. Cost rises by 19.947 (95% CI $[11.540, 28.307]$), while return changes by 11.171 ($[-10.096, 31.042]$). With reward tolerance 25.052, the lower return bound remains within the noninferiority region. The result is therefore *interval-qualified*. At the checkpoint level, all three cost and reward-margin directions are positive, giving exact component and joint values of $1/8 = 0.125$; the 18-way Holm family consequently has no rejection. Appendix A.8 reports the complete estimation and confirmatory evidence.

The three Main-2 training-seed cost changes are $+13.72$, $+17.92$, and $+28.20$; their return changes are also positive ($+2.25/+14.40/+16.85$). Thus the group result is not driven by one sign-reversed checkpoint. Table 1 gives every delay comparison. Appendix A.8 reports the complete 18-item family.

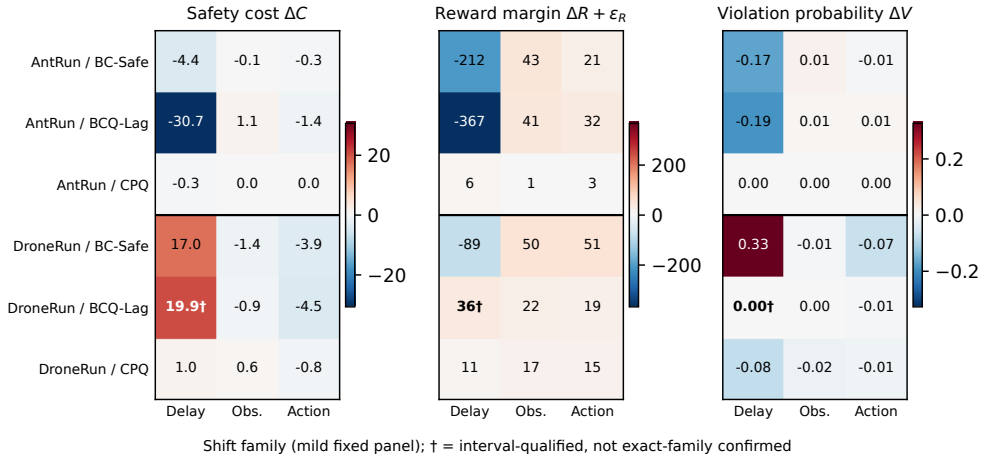

Figure 2: Cross-method final comparison. Cells show group mean changes for cost, reward noninferiority margin, and binary violation probability; red is an increase and blue a decrease. The dagger identifies the interval-qualified candidate; exact family-level results are reported in Table 7.

Table 1: Held-out one-step delay effects. Intervals are hierarchical over three training seeds and 50 paired episodes per seed. Interval-qualified passes both lower bounds; no row passes exact 18-way Holm confirmation.

Task	Policy	Shift	ΔC [95% CI]	ΔR [95% CI]	Outcome
AntRun	BC-Safe	Delay 1	-4.42 [-9.63, 1.13]	-241.06 [-303.96, -150.81]	Reward-degrading
AntRun	BCQ-Lag	Delay 1	-30.67 [-53.85, -12.42]	-404.16 [-430.09, -378.64]	Reward-degrading
AntRun	CPQ	Delay 1	-0.27 [-0.76, 0.00]	4.71 [-4.72, 14.81]	Reward-degrading
DroneRun	BC-Safe	Delay 1	16.98 [0.25, 39.59]	-99.28 [-187.59, -1.42]	Reward-degrading
DroneRun	BCQ-Lag	Delay 1	**19.95 [11.54, 28.31]**	**11.17 [-10.10, 31.04]**	**Interval-qualified**
DroneRun	CPQ	Delay 1	1.02 [-8.99, 11.89]	-3.32 [-18.57, 12.43]	Reward-degrading

5.4 RQ3: POLICY, TASK, SEED, AND METRIC HETEROGENEITY

Fig. 2 provides a common comparison grid for the three policies and three shifts, revealing pronounced task-policy specificity. Delay decreases AntRun BCQ-Lag cost by 30.7 while sharply reducing return (reward margin -367), whereas it increases DroneRun BCQ-Lag cost by 19.9 while preserving reward. DroneRun BC-Safe also exhibits a delay-induced cost increase of $+16.980$ ($[0.247, 39.587]$), accompanied by a strongly negative reward margin and therefore classified as joint degradation. CPQ and both noise families show cost intervals overlapping zero.

The metric comparison reveals a second monitoring blind spot. Nominal DroneRun BCQ-Lag already violates the cost limit in every episode (violation rate 1.000). Delay raises its mean cost from 91.2 to 111.2 on average across checkpoints, yet $\Delta V = 0$ because the binary indicator is saturated. Its mean per-checkpoint CVaR_{10} cost change is $+12.33$ (seed changes $+6.8/0.0/+30.2$). The detected effect is therefore an *already-unsafe amplification*: reward monitoring overlooks the preserved-performance cost increase, while binary violation monitoring overlooks its severity. Conversely, DroneRun BC-Safe delay raises violation probability by 0.327, consistent with visible joint degradation. The nominal-feasible combinations remain below the evidence threshold for a limit-crossing classification.

The checkpoint-resolved forest plot in Appendix A.5 separates environment uncertainty from training uncertainty. BC-Safe DroneRun cost responses are all positive but disperse substantially, whereas AntRun BCQ-Lag is consistently negative. The interval-qualified row places all cost and reward-margin checkpoint means on the favorable side of zero; the exact value of $p = 0.125$ reflects the three independent training clusters, while the 150 paired episodes refine within-checkpoint estimates.

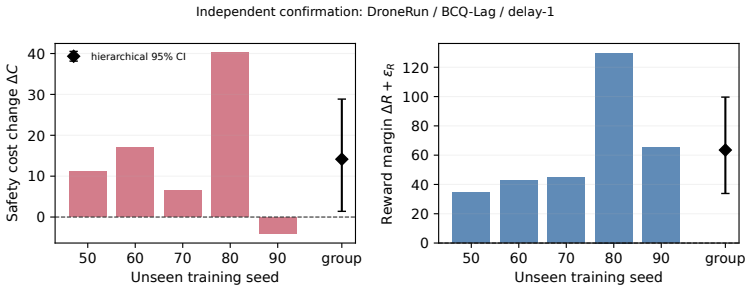

Figure 3: **Independent outcome-blind replication.** Bars are paired effects for five newly trained checkpoints; diamonds and intervals are equal-checkpoint hierarchical estimates. Cost increases in four checkpoints and reward increases in all five; the exact joint value is $p_{\text{joint}} = 0.1875$.

5.5 RQ4: TASK, SEED-BLOCK, AND SEVERITY EXPANSION

Across 5,400 rollouts, the outcome-blind four-task expansion identifies two interval-qualified effects among 28 candidates and maps their task dependence under multiplicity correction. The DroneRun mean effect persists across the delay-severity ladder, whereas Ant delays couple lower cost with substantial reward degradation and the Circle-task candidates remain outside the dual-qualification region. Replication on five entirely new checkpoints produces $\Delta C = 14.14$ ($[1.40, 28.85]$) and $\Delta R = 40.31$ ($[10.65, 76.45]$), with positive cost and reward directions in 4/5 and 5/5 runs, respectively. Thus, the positive mean effect reproduces even at zero reward tolerance; the exact joint value of $p = 0.1875$ quantifies the observed 4/5 cross-checkpoint consistency. Fig. 3 and Table 11 provide the checkpoint-resolved results under the prespecified tasks, candidates, and stopping rule.

Appendix A.9 reports all 28 hypotheses, sensitivity analyses, qualitative rollouts, and the accompanying video evidence; statistical confirmation is based on the prespecified quantitative evaluation.

5.6 RQ5: DEVELOPMENT SEARCH, COMPARATORS, AND PLACEBO

At a 50-episode budget, DRESS acquisition achieves the strongest point estimates across ten development problems: its qualification-hit rate reaches 0.300, compared with 0.200 for grid and 0.250 for random screening, while mean regret falls to 2.156 from 4.214 and 2.813, respectively. This corresponds to a 49% regret reduction relative to grid and 23% relative to random allocation. Paired intervals characterize the variability across development problems, and the methods converge as the budget increases. A 10,000-repeat nominal placebo yields a well-controlled 2.17% mean false-qualification rate and a 4.0% worst-problem rate. Appendix A.12 reports the budget curves in Fig. 16, numerical comparisons in Table 13, and calibration results in Table 14.

6 CONCLUSION

DRESS reframes post-training robustness evaluation as a directional test of whether safety cost increases while task performance is preserved. Paired rollouts, sample splitting, hierarchical effect estimation, and exact training-run inference jointly expose failure modes overlooked by reward and binary violation metrics. On DroneRun, DRESS identifies a one-step-delay amplification whose positive mean cost and return effects reproduce on five unseen checkpoints even at zero reward tolerance. The broader task and severity expansion further maps the effect's task dependence and cross-checkpoint heterogeneity, while placebo experiments verify controlled false qualification. By connecting interpretable deployment shifts to reproducible, policy-specific evidence, DRESS provides a principled foundation for detecting high-performing yet increasingly unsafe policies before deployment.

AI USE STATEMENT

We used AI-assisted tools to support literature search and background research, as well as to improve the clarity, grammar, and readability of the manuscript. All retrieved references and factual informa-

tion were independently verified by the authors. The AI tools were not used to generate scientific results, conduct experiments, or make research conclusions. The authors take full responsibility for the content of the paper.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 22–31, 2017.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021.

Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

Rintaro Ayabe, Yuki Yamagata, and Masashi Sugiyama. Robustness of offline reinforcement learning under action perturbations. *Transactions on Machine Learning Research*, 2025.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2052–2062, 2019.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruochen Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety-gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

Scott Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip S. Thomas. Evaluating the performance of reinforcement learning algorithms. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 4962–4973, 2020.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Zuxin Liu, Zijian Guo, Zhepeng Cen, Huan Zhang, Jie Tan, Bo Li, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 21611–21630, 2023a.

Zuxin Liu, Zijian Guo, Zhepeng Cen, Huan Zhang, Jie Tan, Bo Li, and Ding Zhao. On the robustness of safe reinforcement learning under observational perturbations. In *International Conference on Learning Representations*, 2023b.

Zuxin Liu, Zijian Guo, Zhepeng Cen, Huan Zhang, Yihang Yao, Hanjiang Hu, and Ding Zhao. Towards robust and safe reinforcement learning with benign off-policy data. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 21586–21610, 2023c.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and benchmarks for offline safe reinforcement learning. *arXiv preprint arXiv:2306.09303*, 2023d.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024.

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized Q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8753–8760, 2022. doi: 10.1609/aaai.v36i8.20855.

A Additional Experimental Results and Reproducibility

A.1 DRESS algorithm

Algorithm 1 DRESS sample-split screening and confirmation

Require: Frozen checkpoints $\{\pi_j\}_{j=1}^{J}$; panel Θ; discovery seeds S_D; confirmation seeds S_C; tolerance fraction η; budgets $\mathcal{B}$

Verify simulator support and uniquely identify every candidate
for $j \in \{1, \ldots, J\}$ and $\theta \in \Theta$ **do**
    Roll out nominal and θ with paired seeds S_D; verify identities and transformations
Allocate discovery batches using grid, random, or Eq. (14)
Freeze selected candidate(s), tolerance rule, and hypothesis family
for each frozen θ and checkpoint j **do**
    Collect paired nominal/shifted rollouts using disjoint S_C
    Exclude pairs that violate identity, seed, or transformation validity
Compute Eq. (7) and hierarchical bootstrap intervals
Form exact component values by Eq. (11) and p_{joint} by Eq. (16)
Apply Holm across the declared family; return confirmed amplifications and all non-confirmations

A.2 Evaluation comparisons

Table 2 makes explicit which objects are compared. DRESS is not trained against BC-Safe, BCQ-Lag, or CPQ as a control-policy competitor; it audits each of them under identical paired shifts. The algorithmic comparison occurs one layer above, where uniform grid, uniform random, and DRESS acquisition receive the same screening budget. This separation avoids attributing a checkpoint's nominal performance to the audit procedure.

Table 2: The three comparison layers, their outcomes, and fairness controls.

Layer	Compared entries	Outcome	Fairness control
Policy robustness	BC-Safe, BCQ-Lag, CPQ	Nominal reward-cost and paired shift effects	Same two tasks, three training seeds, shifts, episode seeds, and 50 episodes per cell
Screening search	Uniform grid, uniform random, DRESS acquisition	Held-out qualification-hit rate and regret at budgets 25/50/100	Same five candidates, discovery evidence, confirmation oracle, and episode budget
Safety monitor	Mean cost, CVaR_{10} cost, binary violation	Sensitivity to failure magnitude and threshold saturation	Computed from the same paired rollouts; only mean cost enters qualification

Table 3: Held-out datasets and task interfaces. MiB denotes the HDF5 dataset size.

Task	Transitions	Obs. dim	Act. dim	Horizon	Cost limit	MiB
AntRun	363,010	33	8	200	10	86.7
DroneRun	395,099	17	4	200	10	51.4

A.3 All-condition policy operating points

Fig. 5 extends the nominal comparison to every final condition and every checkpoint. Connecting each shifted point to its own nominal point shows both direction and absolute operating regime. This view complements paired deltas: for example, two policies can have similar ΔC while operating at very different absolute cost levels.

Table 4: Nominal policy operating points, averaged equally over three training seeds. CVaR_{10} is the mean of each checkpoint's worst 10% episode costs.

Task	Policy	Norm. reward	Cost	CVaR_{10} cost	Violation rate
AntRun	BC-Safe	0.613	16.8	38.5	0.653
AntRun	BCQ-Lag	0.775	89.2	126.6	0.973
AntRun	CPQ	0.029	0.3	1.0	0.000
DroneRun	BC-Safe	0.299	5.5	31.8	0.100
DroneRun	BCQ-Lag	0.730	91.2	126.9	1.000
DroneRun	CPQ	0.400	53.6	86.1	0.953

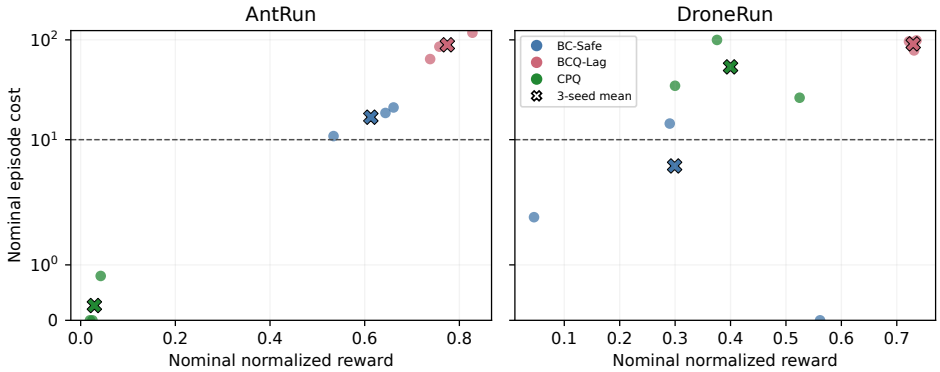

Figure 4: Nominal reward-cost operating points for every task, policy, and training seed. Crosses are policy means and the dashed line is the task cost limit. The symmetric-log cost axis retains near-zero policies.

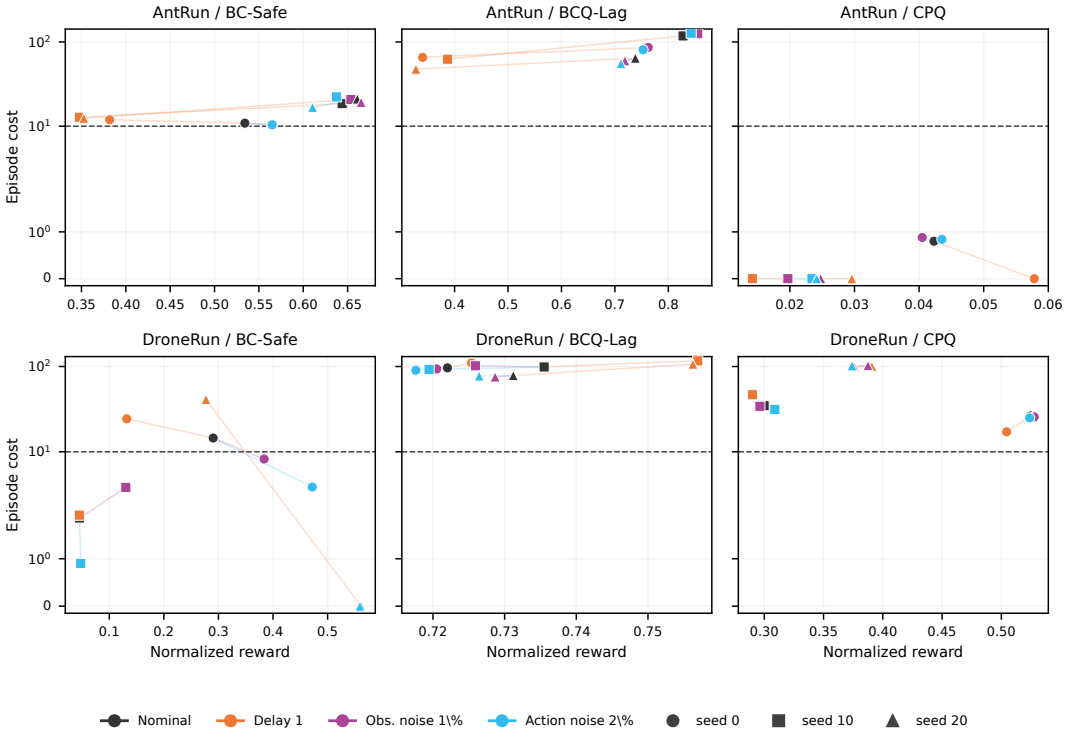

Figure 5: Absolute reward-cost outcomes for all 72 Final Main-2 cells. Color denotes condition, marker denotes training seed, and faint segments connect each shift to the matching nominal checkpoint. The dashed line is the task cost limit.

A.4 POLICY-WISE QUALIFICATION AND CONFIRMATION AUDIT

Table 5 decomposes the final decision into reward feasibility, positive cost effect, interval qualification, and exact family confirmation, exposing how the complete candidate set progresses through each evidence stage.

Table 5: Counts over the three final shifts for each task-policy pair. Cost-positive and interval-qualified use hierarchical lower bounds; exact-confirmed additionally requires the training-seed joint test and Holm correction.

Task	Policy	Reward-feasible	Cost-positive	Interval-qualified	Exact-confirmed
AntRun	BC-Safe	1/3	0/3	0/3	0/3
AntRun	BCQ-Lag	2/3	0/3	0/3	0/3
AntRun	CPQ	0/3	0/3	0/3	0/3
DroneRun	BC-Safe	2/3	1/3	0/3	0/3
DroneRun	BCQ-Lag	2/3	1/3	**1/3**	0/3
DroneRun	CPQ	2/3	0/3	0/3	0/3

A.5 CHECKPOINT-RESOLVED FINAL EFFECTS

The main text uses equal-checkpoint aggregation because training runs, not episodes, are the upper-level replication unit. Fig. 6 exposes all 54 shifted cells and four diagnostic views, while Fig. 7 adds checkpoint intervals for the delay family. Together they show that direction can change across checkpoints even when a group mean is near zero, while the interval-qualified DroneRun/BCQ-Lag delay row retains positive mean-cost changes for all three seeds.

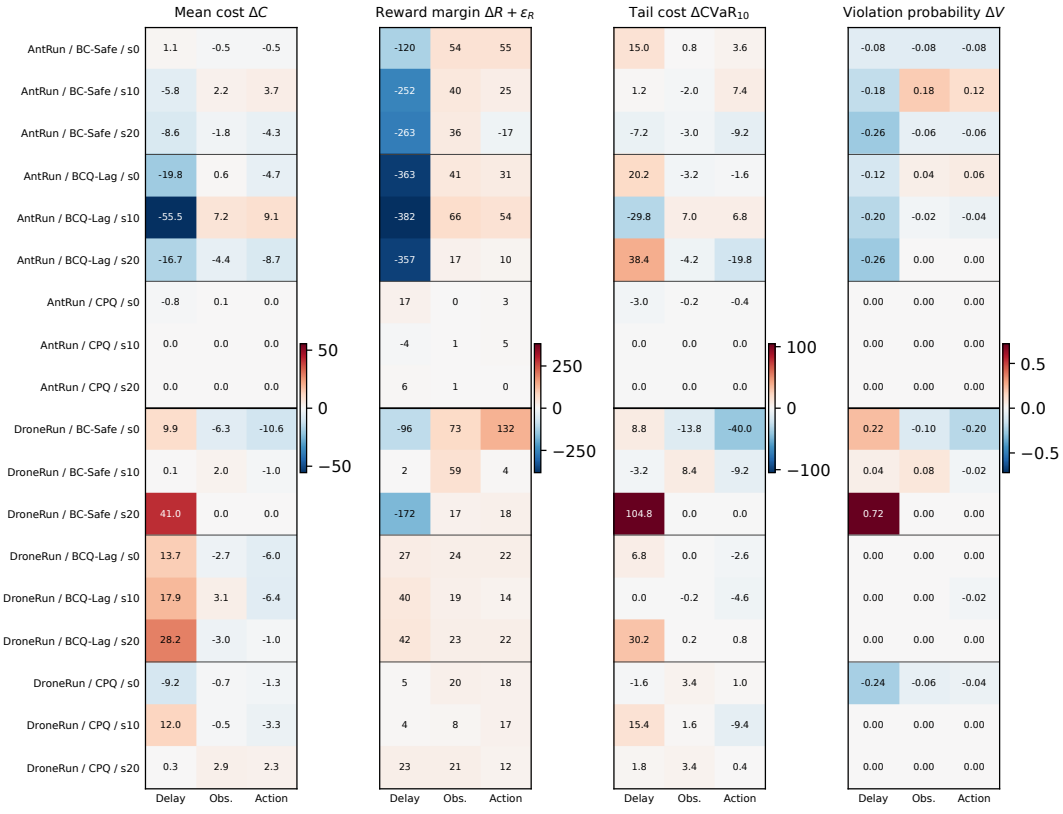

Figure 6: Checkpoint-resolved paired effects for every final task, policy, seed, and shift. Each row is one frozen checkpoint; columns are the three mild shifts. Color scales are centered independently at zero within each metric panel.

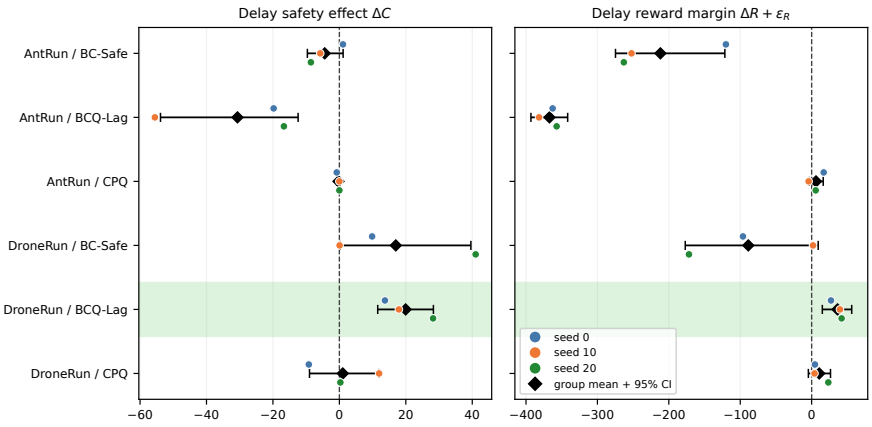

Figure 7: Original one-step-delay heterogeneity. Points are checkpoint means; diamonds/bars are hierarchical estimates and 95% intervals; green marks the interval-qualified row.

A.6 Episode-paired structure of the interval-qualified example

Fig. 8 shows the 50 paired episodes for each DroneRun BCQ-Lag checkpoint under delay. Individual pairs are heterogeneous, as expected in stochastic evaluation, while all three checkpoint means lie above zero cost change and on the reward-feasible side of their checkpoint-specific tolerance. The hierarchical interval evaluates the mean effect, and the exact analysis evaluates directional consistency across training seeds.

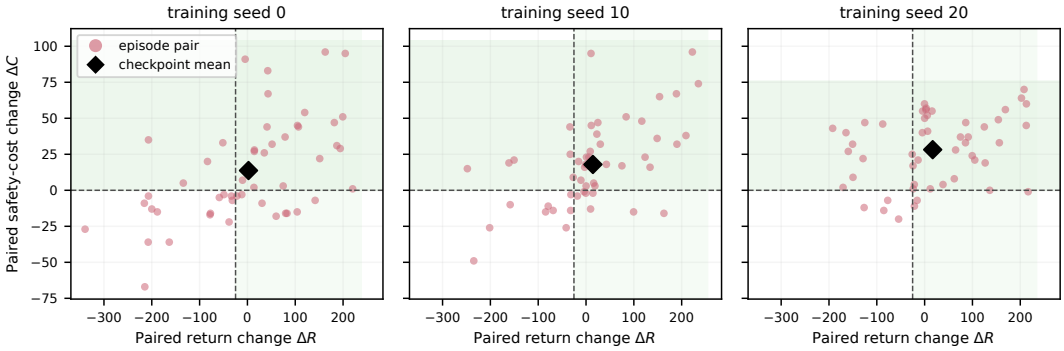

Figure 8: Episode-paired return and safety-cost changes for the interval-qualified task-policy-shift group. Dashed vertical lines mark $-\epsilon_R$ for each checkpoint, dashed horizontal lines mark zero cost change, and black diamonds are checkpoint means.

A.7 Complete checkpoint-level diagnostic table

Table 6 reports every shifted checkpoint cell, including absolute shifted performance and four paired changes. It is deliberately exhaustive so that readers can reconstruct the policy, task, shift, and seed heterogeneity without reading values from figures.

Table 6: Complete checkpoint-level Final Main-2 diagnostics. Norm. R and cost are absolute shifted-condition means; all deltas use the paired nominal condition.

Task	Policy	Seed	Shift	Norm. R	Cost	ΔR	ΔC	ΔCVaR_{10}	ΔV
AntRun	BC-Safe	0	Delay 1	0.382	11.9	-145.6	1.1	15.0	-0.08
AntRun	BC-Safe	0	Obs. noise 1%	0.564	10.3	28.6	-0.5	0.8	-0.08
AntRun	BC-Safe	0	Action noise 2%	0.565	10.4	29.4	-0.5	3.6	-0.08
AntRun	BC-Safe	10	Delay 1	0.348	12.8	-283.0	-5.8	1.2	-0.18
AntRun	BC-Safe	10	Obs. noise 1%	0.654	20.7	9.4	2.2	-2.0	0.18
AntRun	BC-Safe	10	Action noise 2%	0.637	22.3	-6.1	3.7	7.4	0.12
AntRun	BC-Safe	20	Delay 1	0.353	12.4	-294.6	-8.6	-7.2	-0.26

Continued on next page

Table 6 (continued)

Task	Policy	Seed	Shift	Norm. R	Cost	ΔR	ΔC	ΔCVaR_{10}	ΔV
AntRun	BC-Safe	20	Obs. noise 1%	0.665	19.2	4.2	-1.8	-3.0	-0.06
AntRun	BC-Safe	20	Action noise 2%	0.610	16.8	-48.2	-4.3	-9.2	-0.06
AntRun	BCQ-Lag	0	Delay 1	0.340	65.7	-398.9	-19.8	20.2	-0.12
AntRun	BCQ-Lag	0	Obs. noise 1%	0.762	86.1	4.4	0.6	-3.2	0.04
AntRun	BCQ-Lag	0	Action noise 2%	0.753	80.8	-4.8	-4.7	-1.6	0.06
AntRun	BCQ-Lag	10	Delay 1	0.387	62.4	-421.2	-55.5	-29.8	-0.20
AntRun	BCQ-Lag	10	Obs. noise 1%	0.855	125.1	26.2	7.2	7.0	-0.02
AntRun	BCQ-Lag	10	Action noise 2%	0.843	127.0	14.6	9.1	6.8	-0.04
AntRun	BCQ-Lag	20	Delay 1	0.328	47.5	-392.4	-16.7	38.4	-0.26
AntRun	BCQ-Lag	20	Obs. noise 1%	0.719	59.8	-18.7	-4.4	-4.2	0.00
AntRun	BCQ-Lag	20	Action noise 2%	0.711	55.5	-25.7	-8.7	-19.8	0.00
AntRun	CPQ	0	Delay 1	0.058	0.0	14.8	-0.8	-3.0	0.00
AntRun	CPQ	0	Obs. noise 1%	0.040	0.9	-1.7	0.1	-0.2	0.00
AntRun	CPQ	0	Action noise 2%	0.044	0.8	1.2	0.0	-0.4	0.00
AntRun	CPQ	10	Delay 1	0.014	0.0	-5.2	0.0	0.0	0.00
AntRun	CPQ	10	Obs. noise 1%	0.020	0.0	0.1	0.0	0.0	0.00
AntRun	CPQ	10	Action noise 2%	0.023	0.0	3.6	0.0	0.0	0.00
AntRun	CPQ	20	Delay 1	0.030	0.0	4.5	0.0	0.0	0.00
AntRun	CPQ	20	Obs. noise 1%	0.025	0.0	-0.1	0.0	0.0	0.00
AntRun	CPQ	20	Action noise 2%	0.024	0.0	-0.7	0.0	0.0	0.00
DroneRun	BC-Safe	0	Delay 1	0.132	24.4	-106.5	9.9	8.8	0.22
DroneRun	BC-Safe	0	Obs. noise 1%	0.384	8.2	62.8	-6.3	-13.8	-0.10
DroneRun	BC-Safe	0	Action noise 2%	0.472	3.9	122.0	-10.6	-40.0	-0.20
DroneRun	BC-Safe	10	Delay 1	0.045	1.9	-0.1	0.1	-3.2	0.04
DroneRun	BC-Safe	10	Obs. noise 1%	0.130	3.8	57.0	2.0	8.4	0.08
DroneRun	BC-Safe	10	Action noise 2%	0.048	0.9	1.5	-1.0	-9.2	-0.02
DroneRun	BC-Safe	20	Delay 1	0.277	41.0	-191.2	41.0	104.8	0.72
DroneRun	BC-Safe	20	Obs. noise 1%	0.558	0.0	-2.2	0.0	0.0	0.00
DroneRun	BC-Safe	20	Action noise 2%	0.559	0.0	-1.4	0.0	0.0	0.00
DroneRun	BCQ-Lag	0	Delay 1	0.725	110.1	2.3	13.7	6.8	0.00
DroneRun	BCQ-Lag	0	Obs. noise 1%	0.721	93.7	-1.0	-2.7	0.0	0.00
DroneRun	BCQ-Lag	0	Action noise 2%	0.718	90.4	-3.0	-6.0	-2.6	0.00
DroneRun	BCQ-Lag	10	Delay 1	0.757	116.7	14.4	17.9	0.0	0.00
DroneRun	BCQ-Lag	10	Obs. noise 1%	0.726	101.9	-6.4	3.1	-0.2	0.00
DroneRun	BCQ-Lag	10	Action noise 2%	0.719	92.4	-10.8	-6.4	-4.6	-0.02
DroneRun	BCQ-Lag	20	Delay 1	0.756	106.6	16.9	28.2	30.2	0.00
DroneRun	BCQ-Lag	20	Obs. noise 1%	0.729	75.4	-1.7	-3.0	0.2	0.00
DroneRun	BCQ-Lag	20	Action noise 2%	0.726	77.4	-3.2	-1.0	0.8	0.00
DroneRun	CPQ	0	Delay 1	0.504	17.1	-13.6	-9.2	-1.6	-0.24
DroneRun	CPQ	0	Obs. noise 1%	0.528	25.6	2.1	-0.7	3.4	-0.06
DroneRun	CPQ	0	Action noise 2%	0.524	25.0	-0.5	-1.3	1.0	-0.04
DroneRun	CPQ	10	Delay 1	0.290	46.6	-6.6	12.0	15.4	0.00
DroneRun	CPQ	10	Obs. noise 1%	0.296	34.1	-2.4	-0.5	1.6	0.00
DroneRun	CPQ	10	Action noise 2%	0.309	31.4	6.1	-3.3	-9.4	0.00
DroneRun	CPQ	20	Delay 1	0.391	100.1	10.2	0.3	1.8	0.00
DroneRun	CPQ	20	Obs. noise 1%	0.387	102.7	8.0	2.9	3.4	0.00
DroneRun	CPQ	20	Action noise 2%	0.374	102.1	-1.0	2.3	0.4	0.00

A.8 Complete held-out result family

Fig. 9 places all 18 claims in reward-margin-cost space, and Table 7 gives their complete numerical values. The joint presentation characterizes both the uniquely interval-qualified cell and the specificity of the remaining policy-shift responses. Exact training-seed Holm values provide the corresponding family-level evidence.

A.9 Prespecified robustness expansion

The extension was frozen before reading its outcomes. It adds AntCircle and DroneCircle to the original Run tasks, uses a disjoint episode block 40000:40049, and evaluates the original three-policy mild panel plus a BCQ-Lag delay ladder. Table 8 records the expanded dataset and interface scale. Treating the 28 hypotheses as a separate Holm family preserves the inferential boundary of the original 18-test panel. Fig. 10 isolates the delay ladder, while Fig. 11 and Table 9 place it alongside positive, negative, and unevaluated combinations. Gray heatmap cells denote combinations outside the prespecified extension.

Complete numerical family. Table 10 lists all 28 hypotheses, including the Circle-policy noise results and the negative Ant delay ladder.

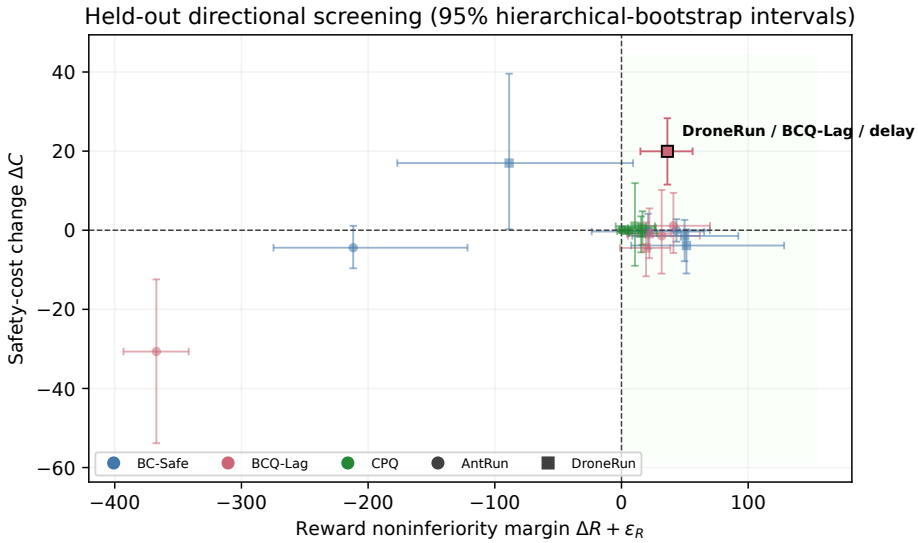

Figure 9: All held-out final claims. Horizontal position is reward noninferiority margin $\Delta R + \epsilon_R$ and vertical position is safety-cost change. Bars are 95% hierarchical-bootstrap intervals. The outlined DroneRun/BCQ-Lag/delay point satisfies dual interval qualification; Table 7 reports the family-level evidence.

Table 7: All held-out task-policy-shift effects. Intervals are hierarchical over three training seeds and 50 paired episodes per seed.

Task	Policy	Shift	ΔC [95% CI]	ΔR [95% CI]	Outcome
AntRun	BC-Safe	Delay 1	-4.42 [-9.63, 1.13]	-241.06 [-303.96, -150.81]	Reward-degrading
AntRun	BC-Safe	Action noise 2%	-0.33 [-4.66, 4.17]	-8.28 [-52.83, 36.00]	Reward-degrading
AntRun	BC-Safe	Obs. noise 1%	-0.05 [-2.89, 2.79]	14.07 [-9.75, 40.43]	Cost-inconclusive
AntRun	BCQ-Lag	Delay 1	-30.67 [-53.85, -12.42]	-404.16 [-430.09, -378.64]	Reward-degrading
AntRun	BCQ-Lag	Action noise 2%	-1.42 [-10.96, 10.19]	-5.29 [-31.80, 24.84]	Cost-inconclusive
AntRun	BCQ-Lag	Obs. noise 1%	1.13 [-5.75, 9.45]	3.98 [-21.89, 32.89]	Cost-inconclusive
AntRun	CPQ	Delay 1	-0.27 [-0.76, 0.00]	4.71 [-4.72, 14.81]	Reward-degrading
AntRun	CPQ	Action noise 2%	0.01 [-0.07, 0.11]	1.36 [-1.78, 4.56]	Reward-degrading
AntRun	CPQ	Obs. noise 1%	0.03 [-0.02, 0.11]	-0.60 [-3.30, 1.31]	Reward-degrading
DroneRun	BC-Safe	Delay 1	16.98 [0.25, 39.59]	-99.28 [-187.59, -1.42]	Reward-degrading
DroneRun	BC-Safe	Action noise 2%	-3.87 [-10.93, 0.20]	40.71 [-2.93, 117.83]	Cost-inconclusive
DroneRun	BC-Safe	Obs. noise 1%	-1.44 [-7.84, 2.62]	39.21 [-2.02, 81.64]	Cost-inconclusive
DroneRun	BCQ-Lag	Delay 1	**19.95 [11.54, 28.31]**	**11.17 [-10.10, 31.04]**	**Interval-qualified**
DroneRun	BCQ-Lag	Action noise 2%	-4.47 [-11.65, 1.97]	-5.65 [-26.24, 13.48]	Reward-degrading
DroneRun	BCQ-Lag	Obs. noise 1%	-0.88 [-7.07, 5.51]	-3.05 [-20.91, 14.15]	Cost-inconclusive
DroneRun	CPQ	Delay 1	1.02 [-8.99, 11.89]	-3.32 [-18.57, 12.43]	Reward-degrading
DroneRun	CPQ	Action noise 2%	-0.77 [-5.59, 3.50]	1.52 [-8.29, 14.17]	Cost-inconclusive
DroneRun	CPQ	Obs. noise 1%	0.55 [-3.69, 4.79]	2.58 [-8.89, 13.17]	Cost-inconclusive

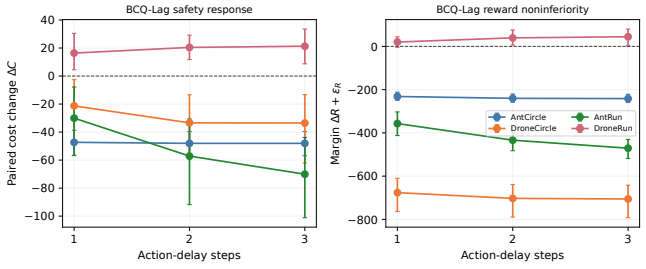

Figure 10: Independent BCQ-Lag delay expansion over four tasks. Points and bars are hierarchical means and 95% intervals across three checkpoints and 50 paired episodes per checkpoint.

Table 8: Dataset and interface scale after task expansion. Circle adds a distinct objective and substantially more offline transitions while retaining the Ant/Drone embodiment split.

Task	Transitions	Obs. dim	Act. dim	Horizon	Cost limit	MiB
AntRun	363,010	33	8	200	10	86.7
DroneRun	395,099	17	4	200	10	51.4
AntCircle	2,510,820	34	8	500	10	612.3
DroneCircle	576,243	18	4	300	10	77.4

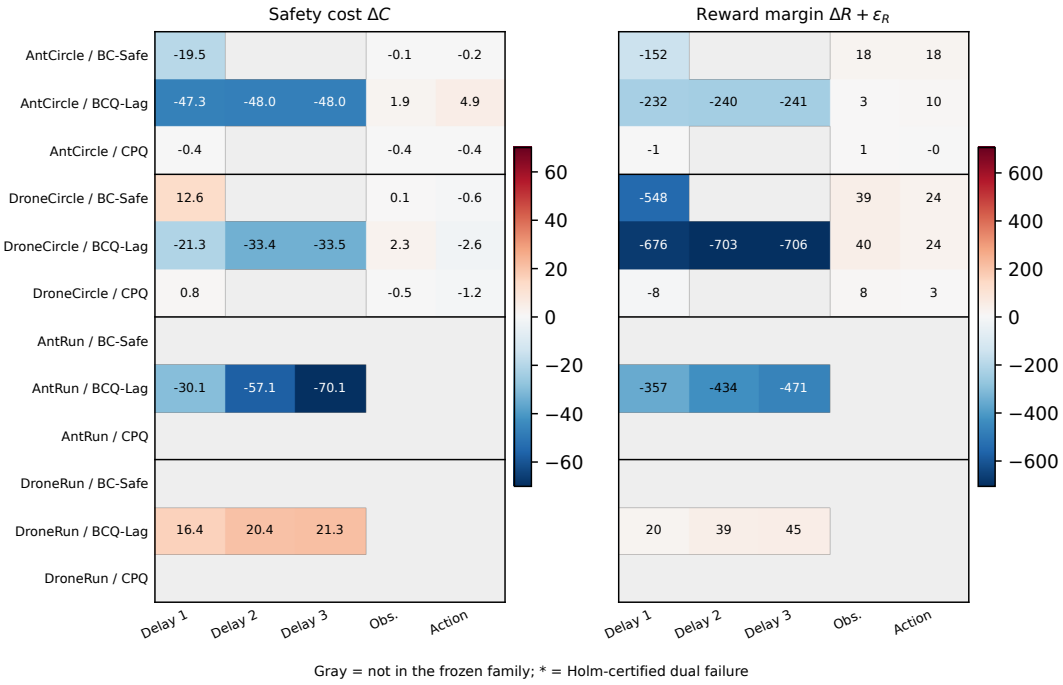

Figure 11: All 28 robustness-expansion hypotheses. Values are hierarchical mean cost changes and reward-noninferiority margins; the separate 28-way Holm analysis provides family-level evidence.

Table 9: Complete BCQ-Lag delay ladder in the independent expansion. Confirmation requires both interval qualification and rejection under the 28-way Holm correction.

Task	Delay	ΔC [95% CI]	ΔR [95% CI]	Holm p	Status
AntCircle	1	-47.31 [-56.49, -38.71]	-244.4 [-262.3, -224.6]	1.0000	Reward-unqualified
AntCircle	2	-48.05 [-56.85, -39.63]	-253.0 [-271.0, -232.9]	1.0000	Reward-unqualified
AntCircle	3	-48.05 [-56.85, -39.63]	-254.1 [-272.0, -234.1]	1.0000	Reward-unqualified
DroneCircle	1	-21.29 [-46.53, -2.57]	-713.7 [-800.8, -647.3]	1.0000	Reward-unqualified
DroneCircle	2	-33.39 [-59.21, -13.37]	-740.3 [-826.8, -676.5]	1.0000	Reward-unqualified
DroneCircle	3	-33.55 [-62.03, -13.28]	-743.1 [-829.6, -679.4]	1.0000	Reward-unqualified
AntRun	1	-30.05 [-56.80, -7.81]	-393.5 [-449.0, -339.7]	1.0000	Reward-unqualified
AntRun	2	-57.13 [-91.80, -30.68]	-470.3 [-519.1, -432.1]	1.0000	Reward-unqualified
AntRun	3	-70.07 [-101.12, -43.83]	-507.5 [-554.6, -466.9]	1.0000	Reward-unqualified
DroneRun	1	+16.40 [+4.47, +30.45]	-5.7 [-30.0, +18.2]	1.0000	Reward-unqualified
DroneRun	2	+20.43 [+11.81, +29.19]	+13.6 [-19.6, +50.4]	0.4779	Interval-qualified
DroneRun	3	+21.29 [+8.72, +33.55]	+18.7 [-22.4, +54.4]	0.0924	Interval-qualified

Table 10: All prespecified expansion hypotheses. R-margin is $\Delta R + \epsilon_R$.

Task	Policy	Condition	ΔC	C-LCB	ΔR	R-margin	Status
AntCircle	BC-Safe	delay-mild	-19.49	-25.89	-160.4	-151.8	Reward-unqualified
AntCircle	BC-Safe	obs-noise	-0.14	-6.05	+9.4	+18.1	Reward-unqualified
AntCircle	BC-Safe	action-noise	-0.21	-7.02	+9.4	+18.1	Reward-unqualified
AntCircle	BCQ-Lag	delay-mild	-47.31	-56.49	-244.4	-231.6	Reward-unqualified
AntCircle	BCQ-Lag	obs-noise	+1.87	-8.52	-9.7	+3.0	Reward-unqualified
AntCircle	BCQ-Lag	action-noise	+4.90	-4.25	-2.3	+10.5	Reward-unqualified
AntCircle	BCQ-Lag	delay-medium	-48.05	-56.85	-253.0	-240.3	Reward-unqualified
AntCircle	BCQ-Lag	delay-severe	-48.05	-56.85	-254.1	-241.4	Reward-unqualified
AntCircle	CPQ	delay-mild	-0.37	-1.49	-0.9	-0.9	Reward-unqualified
AntCircle	CPQ	obs-noise	-0.37	-1.49	+0.4	+0.5	Reward-unqualified
AntCircle	CPQ	action-noise	-0.37	-1.49	-0.5	-0.4	Reward-unqualified
DroneCircle	BC-Safe	delay-mild	+12.60	+9.67	-579.6	-548.4	Reward-unqualified
DroneCircle	BC-Safe	obs-noise	+0.11	-0.91	+7.7	+38.9	Cost-inconclusive
DroneCircle	BC-Safe	action-noise	-0.63	-1.75	-7.0	+24.2	Cost-inconclusive
DroneCircle	BCQ-Lag	delay-mild	-21.29	-46.53	-713.7	-676.2	Reward-unqualified
DroneCircle	BCQ-Lag	obs-noise	+2.29	-0.93	+2.9	+40.4	Cost-inconclusive
DroneCircle	BCQ-Lag	action-noise	-2.57	-7.76	-13.1	+24.4	Cost-inconclusive
DroneCircle	BCQ-Lag	delay-medium	-33.39	-59.21	-740.3	-702.9	Reward-unqualified
DroneCircle	BCQ-Lag	delay-severe	-33.55	-62.03	-743.1	-705.6	Reward-unqualified
DroneCircle	CPQ	delay-mild	+0.77	-7.65	-8.8	-7.5	Reward-unqualified
DroneCircle	CPQ	obs-noise	-0.49	-6.56	+6.4	+7.7	Reward-unqualified
DroneCircle	CPQ	action-noise	-1.18	-7.09	+2.2	+3.4	Reward-unqualified
AntRun	BCQ-Lag	delay-mild	-30.05	-56.80	-393.5	-356.9	Reward-unqualified
AntRun	BCQ-Lag	delay-medium	-57.13	-91.80	-470.3	-433.7	Reward-unqualified
AntRun	BCQ-Lag	delay-severe	-70.07	-101.12	-507.5	-470.8	Reward-unqualified
DroneRun	BCQ-Lag	delay-mild	+16.40	+4.47	-5.7	+20.1	Reward-unqualified
DroneRun	BCQ-Lag	delay-medium	+20.43	+11.81	+13.6	+39.4	Interval-qualified
DroneRun	BCQ-Lag	delay-severe	+21.29	+8.72	+18.7	+44.5	Interval-qualified

Outcome-blind five-checkpoint replication. The original panel nominated DroneRun/BCQ-Lag/delay-1 using seeds 0/10/20. Because an intermediate diagnostic on seeds 30/40 reused nomination checkpoints, independent replication uses newly trained seeds 50/60/70/80/90, paired episode seeds 50000:50049, and a stopping rule fixed before shifted evaluation. Hierarchical aggregation gives $\Delta C = 14.14$ ($[1.40, 28.85]$), $\Delta R = 40.31$ ($[10.65, 76.45]$), and 5% reward margin 63.50 ($[33.84, 99.64]$). Cost changes are $+11.08/+17.02/+6.52/+40.24/-4.14$; all reward changes are positive. The exact cost, reward, and joint values are 0.1875, 0.03125, and 0.1875, respectively. These results reproduce the positive equal-checkpoint mean and quantify directional consistency at four of five checkpoints. Nominal cost remains above the limit, categorizing the effect as already-unsafe amplification.

Table 11: Checkpoint-resolved independent replication. The group row reports hierarchical 95% intervals; per-seed rows are paired episode means.

Training seed	ΔC	ΔR	$\Delta R + \epsilon_R$
50	+11.08	+11.44	+34.63
60	+17.02	+19.73	+42.92
70	+6.52	+21.72	+44.91
80	+40.24	+106.58	+129.77
90	-4.14	+42.06	+65.25
Group	**+14.14 [+1.40, +28.85]**	**+40.31 [+10.65, +76.45]**	**+63.50 [+33.84, +99.64]**

Reward-tolerance sensitivity. The operational margin is varied from 0% to 10% of nominal return using the same rollouts. Fig. 12 and Table 12 show that the independent five-seed return interval is positive even at $\eta = 0$, establishing interval qualification throughout the tested range. The exact joint value remains 0.1875 at every tolerance because it is determined by the 4/5 cost direction count; the three-seed screen has exact resolution $1/8$.

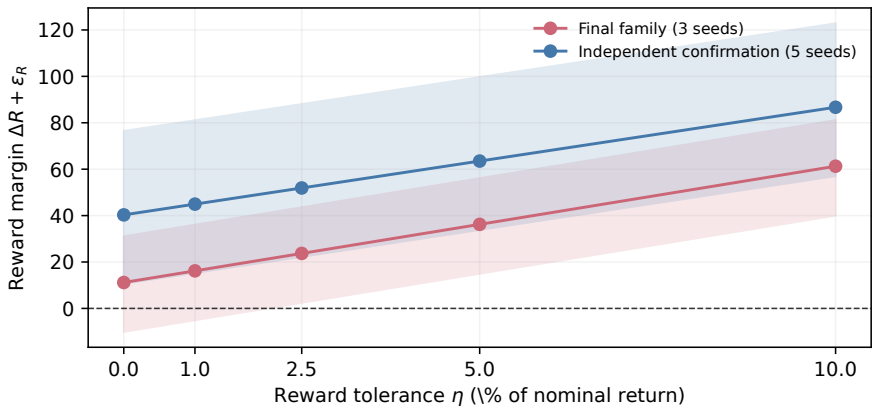

Figure 12: Reward-tolerance sensitivity for the interval-qualified three-seed screen and outcome-blind five-seed replication. Lines show the reward margin and shading shows the hierarchical 95% interval; exact directions and values appear in Table 12.

Table 12: Tolerance sensitivity. Interval qualification uses both hierarchical lower bounds; the exact column reports the one-sided training-seed joint value.

Panel	η	Reward margin [95% CI]	Pos. seeds	Exact p	Interval status
Final family (3 seeds)	0.0%	+11.17 [-10.10, +31.04]	3/3	0.12500	Reward-unqualified
Final family (3 seeds)	1.0%	+16.18 [-5.09, +36.05]	3/3	0.12500	Reward-unqualified
Final family (3 seeds)	2.5%	+23.70 [+2.43, +43.57]	3/3	0.12500	Qualified
Final family (3 seeds)	5.0%	**+36.22 [+14.96, +56.09]**	3/3	0.12500	**Qualified**
Final family (3 seeds)	10.0%	+61.27 [+40.01, +81.15]	3/3	0.12500	Qualified
Independent confirmation (5 seeds)	0.0%	**+40.31 [+10.65, +76.45]**	5/5	0.18750	**Qualified**
Independent confirmation (5 seeds)	1.0%	+44.94 [+15.28, +81.09]	5/5	0.18750	Qualified
Independent confirmation (5 seeds)	2.5%	+51.90 [+22.24, +88.05]	5/5	0.18750	Qualified
Independent confirmation (5 seeds)	5.0%	+63.50 [+33.84, +99.64]	5/5	0.18750	Qualified
Independent confirmation (5 seeds)	10.0%	+86.69 [+57.03, +122.83]	5/5	0.18750	Qualified

Qualitative rollout visualization. Fig. 13 samples four aligned time points from six clips fixed before recording. Each row reports the complete episode return and cost. The clips contrast nominal and delayed BCQ-Lag on DroneRun, delayed BC-Safe/CPQ controls, and the AntRun nominal/delay counterexample. For convenient review, the supplementary video `dress_comparison_dashboard.mp4` combines the unique clips and the post-hoc exemplar into one 1280×720 video with episode metrics and a final aggregate-evidence card.

Fig. 14 provides a complementary visualization of the amplification mechanism. Selected after evaluation as the largest paired ΔC among the 50 DroneRun/BCQ-Lag/seed-0 delay pairs, it illustrates how reward improves from 425.1 to 645.8 while cost rises from 49 to 131 under a shared initial-condition seed. Statistical conclusions are derived from the prespecified aggregate analysis.

A.10 Experimental partitions and seed isolation

The development panel contains action delays of one, two, and three steps and mass multipliers 0.9 and 1.1. Discovery uses episode seeds 10000-10049, confirmation uses 20000-20049, and the final evaluation uses 30000-30049. The final panel contains a one-step action delay, observation Gaussian noise at 1% of the training-split feature scale, and action Gaussian noise at 2% of the action scale. Nominal observations are reused only within the same task, checkpoint, and evaluation phase; discovery and confirmation remain disjoint.

The compact search benchmark uses five-episode batches and budgets 25, 50, and 100. Grid is deterministic and is evaluated once; random and DRESS acquisition use 30 seeded repeats. Selection uses only discovery-split paired differences. Held-out regret assigns risk L_C to an interval-qualified candidate and zero otherwise, and measures the gap between the confirmation oracle's maximum risk and that of the selected candidate.

A.11 Development-profile diagnostics

Fig. 15 records the discovery-stage perturbation profiles used before any final-panel outcome was inspected. Direction reversals across task-policy pairs and nonmonotone responses across adjacent delay lengths motivate the family-specific severity representation.

A.12 Search and placebo tables

Table 13: Compact search benchmark. Regret is lower-is-better; DRESS attains the strongest point estimates at budget 50.

Method	Budget	Qual. hit	Oracle hit	Mean regret
DRESS acq.	25	0.200	0.800	3.474
Grid	25	0.200	0.800	3.474
Random	25	0.187	0.787	3.334
DRESS acq.	50	**0.300**	**0.900**	**2.156**
Grid	50	0.200	0.800	4.214
Random	50	0.250	0.850	2.813
DRESS acq.	100	0.300	0.900	2.156
Grid	100	0.300	0.900	2.156
Random	100	0.303	0.903	2.018

Table 14: Empirical false-qualification rates under the nominal split/permutation placebo.

Statistic	False-qual. rate
Macro (10,000 permutations)	**0.022**
Minimum / maximum problem	**0.000 / 0.040**

A.13 Evaluation-record integrity

Each evaluation record retains the task, algorithm, training run, checkpoint identity, cost limit, episode seed, candidate tuple, and applied transformation. Estimation excludes mismatched nominal-shifted identities, duplicated or reordered seeds, non-finite metrics, and groups that mix experimental

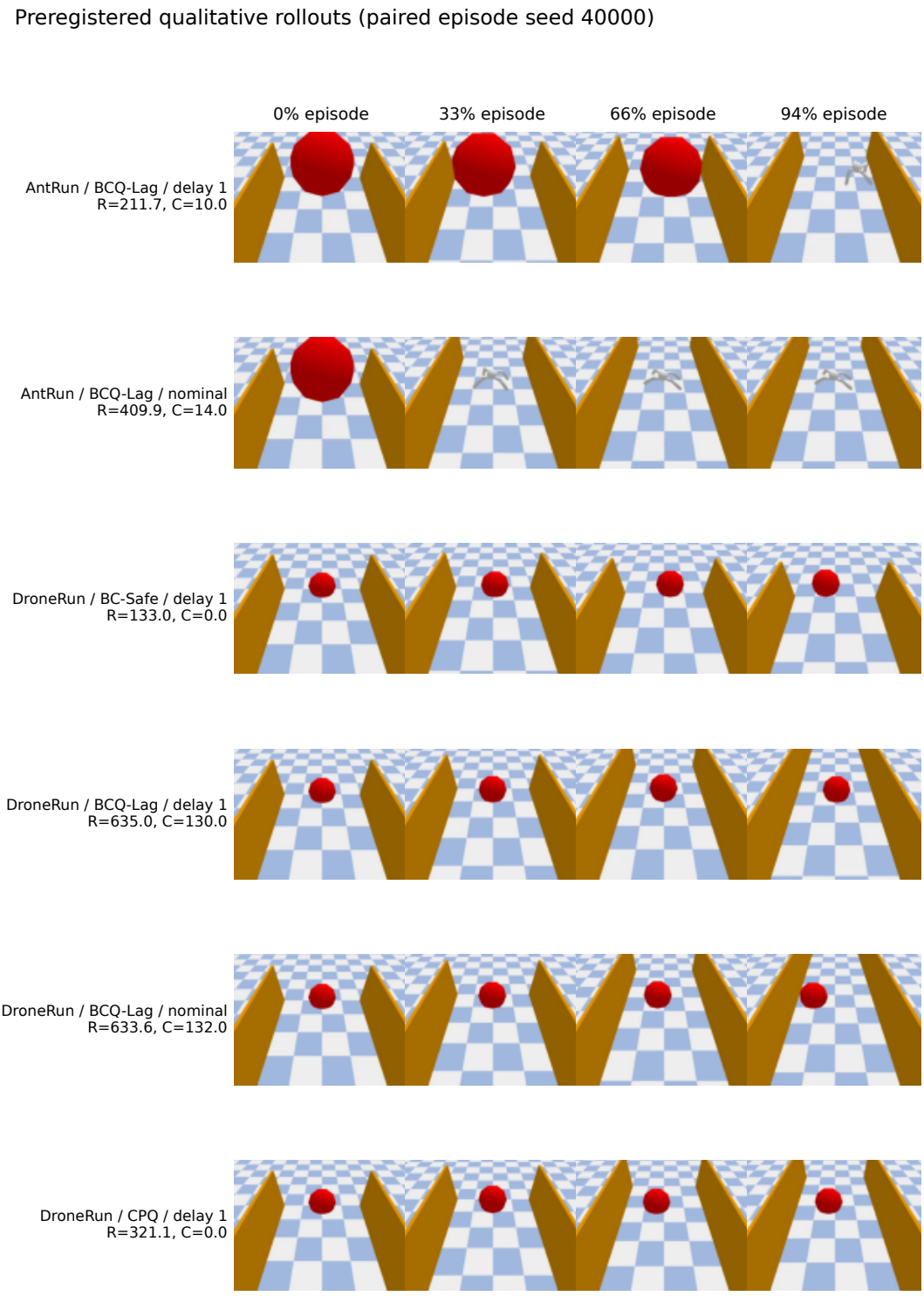

Figure 13: Prespecified paired and control rollouts at episode seed 40000. Columns show aligned episode times; labels report full-episode return and cost. The corresponding videos and evaluation metadata are included in the supplementary material.

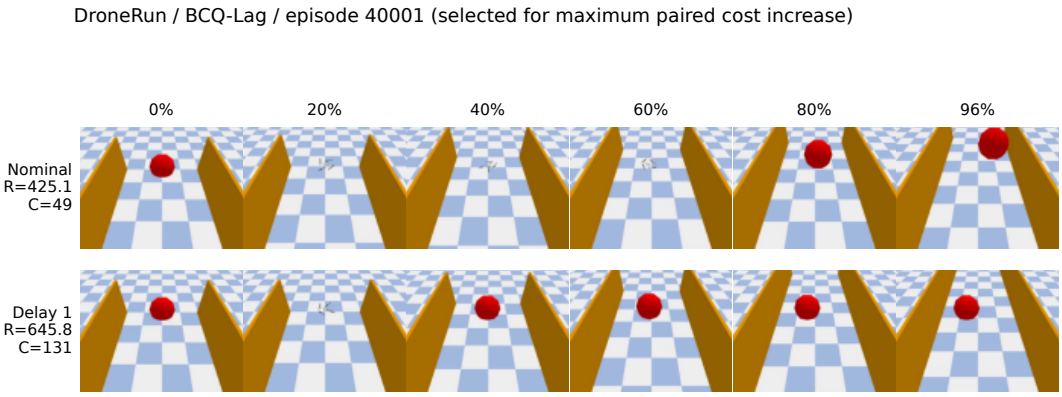

Figure 14: Qualitative visualization of reward-preserving cost amplification. Episode 40001 has the largest paired cost increase (+82) among 50 seed-0 pairs and a simultaneous return increase of 220.7.

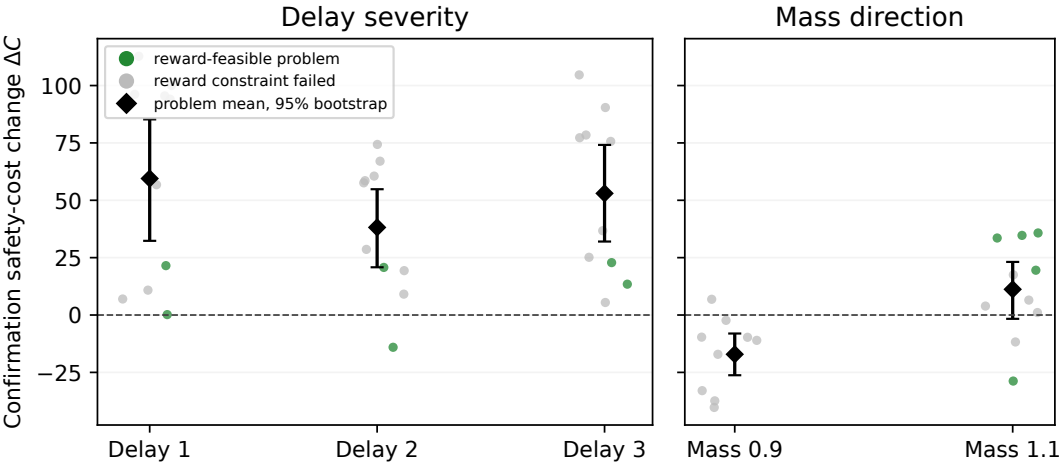

Figure 15: Discovery-stage paired reward and cost changes across action-delay length and mass multiplier. Lines summarize training-seed means and characterize the candidate profiles used before disjoint final evaluation.

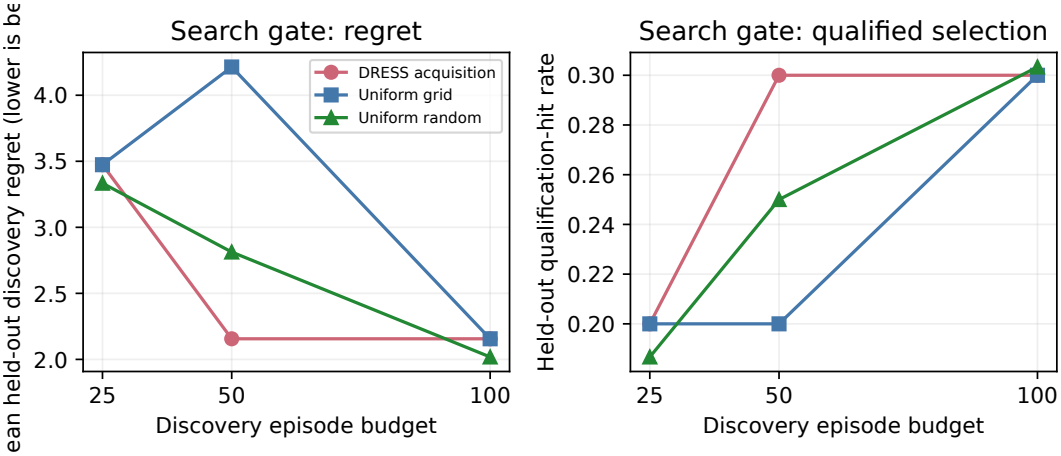

Figure 16: Equal-budget search comparison on held-out outcomes. DRESS attains the strongest qualification-hit and regret point estimates at budget 50, while the methods converge at budget 100; bands show paired variability.

conditions. Discovery observations remain ineligible for confirmation even when their numerical bounds satisfy Eq. (15); inferential status is determined by both the result and its evaluation partition.

B Statistical Properties

Sample splitting permits adaptive selection without reusing optimistic discovery noise in confirmation.

Proposition 1 (selection-valid confirmation). Let conditionally independent discovery and confirmation data D, E correspond to a fixed policy/checkpoint, and let $\hat{\theta} = g(D)$. If each fixed-θ confirmation test has type-I error at most α, so does testing $\hat{\theta}$ using only E.

Proof. Given D, $\hat{\theta}$ is fixed and E retains its confirmation distribution; thus $\Pr(\text{reject a true null} \mid D) \leq \alpha$. Averaging over D proves the claim. □

The proposition establishes selection validity under independence. Finite-sample effect uncertainty is quantified separately by the hierarchical bootstrap, and disjoint episode ranges with record-level validity conditions enforce the required separation.

Proposition 2 (dual-null validity). If p_C and p_R are valid for H_C and H_R, respectively, then $p_{\text{joint}} = \max(p_C, p_R)$ is valid for the union null $H_C \cup H_R$.

Proof. Under the union null some H_k is true, and $\{\max(p_C, p_R) \leq \alpha\} \subseteq \{p_k \leq \alpha\}$, whose probability is at most α. □

Exact sign components require independent training runs and a null probability of a positive checkpoint direction no greater than one half. Holm's step-down procedure on valid exact joint values controls family-wise error under arbitrary dependence (Holm, 1979). The guarantee applies to a prespecified shift family with valid component tests, as implemented by the frozen-panel design.

These results assume independent training runs, paired seeds with comparable nominal and shifted initial conditions, a tolerance fixed before shifted outcomes are observed, and representative resampling. Independent training-run count determines exact resolution: three runs yield $1/8$, while five runs yield $1/32$ when all directions agree and $6/32$ when four of five agree. Episode replication complements this design by refining within-checkpoint effect estimation.