CANDIDATE COMPLETION FOR EXACT OFF-POLICY CALIBRATION OF BEST-OF-M POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Choosing the highest-scoring action from several candidates changes which prediction errors a controller encounters. A bound calibrated on the original policy's data can therefore lose its intended coverage. We introduce candidate completion, which calibrates best-of-M policies using logged trajectories and samples from the preserved data-collection policy. It places each logged action among fresh competitors and tests whether the deployed rule selects it. Applying these tests along a trajectory yields exact target-policy samples without evaluating action densities or collecting new environment outcomes. Ordinary and weighted calibration then provide finite-sample prediction bounds. In repeated-selection experiments on directly observed control rewards, completion corrects Walker2d-high coverage from 78.7% to 90.4% under two-step best-of-four selection. Across all 27 setting/group comparisons, its mean absolute coverage difference from a target-data oracle is 0.69 percentage points, versus 2.12 for behavior calibration. In controlled equal-budget comparisons with 500 logged actions, completion achieves 90.4–91.7% coverage for 4–16 candidates, while score-CDF weighting achieves 86.6–87.3%. Additional selection sweeps, tied-score experiments, and complete-trajectory checks support the sampling and calibration guarantees.

1 INTRODUCTION

A learned controller needs more than a promising action: it also needs a reliable prediction of the outcome. A lower prediction bound summarizes that reliability. At 90% coverage, the outcome is at or above the bound with probability at least 0.9, averaging over calibration data and future outcomes. Calibration chooses how much to subtract from a prediction by examining its errors on held-out data.

Action selection changes which errors matter. In offline reinforcement learning, a policy can sample several candidate actions from a data-collection policy and execute the highest-scoring one (Fujimoto et al., 2019; Ghasemipour et al., 2021; Park et al., 2026). Errors measured on the original actions need not represent errors after selection. Repeating selection also changes the states visited. The calibration data must therefore reflect the deployed policy, called the *target policy*, rather than just the *behavior policy* that generated the log.

Off-policy calibration methods address this change using ratios of target and behavior action probabilities (Taufiq et al., 2022; Zhang et al., 2023; Foffano et al., 2023). For a sample-and-select policy, computing that ratio requires knowing how often an action beats its competitors. Drawing actions can be straightforward even when evaluating this probability is not. A preserved robot controller, for example, can propose actions at recorded states without executing them or repeating a physical experiment.

The key idea. Keep the action whose outcome is already known, and sample only its missing competitors. We place each logged action into a fresh candidate set and apply the deployed selection rule. For pure best-of-M selection among M candidates, we retain a trajectory when its logged action is selected at every active step, with additional tests after early termination. A randomized acceptance rule also handles mixtures of behavior and selected actions. The retained trajectories have exactly the target-policy distribution, including downstream states and termination times. We call this procedure *candidate completion* (CC). The new candidates are scored, never executed. Because we compare

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.

candidate labels rather than require a fresh action to equal the logged one, this construction works directly with continuous actions.

A simple example. Let a logged action a be uniform on $[0, 1]$, its prediction be $G(a) = 2a$, and its outcome be $Y(a) = a$. Subtracting a correction q gives the lower bound $2a - q$. This bound covers the outcome exactly when $a \leq q$, so $q = 0.9$ gives 90% coverage for behavior actions. After selecting the larger of two actions, both candidates must be at most 0.9 for that same bound to cover. Coverage becomes $0.9^2 = 81\%$. Completing each logged action with one fresh competitor recovers the selected-action distribution; its correct population correction is $\sqrt{0.9}$. Section 5.3 tests the finite-data version.

Contributions. **Sampling:** we construct exact target-policy trajectories from logged outcomes and candidate comparisons, including ties and early termination. **Calibration:** we obtain ordinary, higher-confidence, and weighted prediction bounds, with quantified error for approximate behavior samplers. **Evidence:** repeated-selection experiments calibrate directly observed control rewards, while equal-budget comparisons show when completion improves reliability over estimated score weights. Complete group-level results and known-distribution tests connect these gains to the sampling mechanism.

2 Related Work

Learning policies that select actions. Behavior-supported offline reinforcement learning uses candidate actions proposed by a behavior model (Fujimoto et al., 2019; Ghasemipour et al., 2021). Model-based methods also select sequences of actions (Park et al., 2026), and best-of-M analyses describe the resulting output distribution (Beirami et al., 2025). These methods motivate our setting. We take the learned score and sampling rule as fixed, then calibrate predictions for the selected actions and their subsequent outcomes.

Calibrating predictions after a policy change. Conformal prediction uses held-out errors to construct prediction bounds (Lei et al., 2018). Weighted conformal methods adjust for distribution changes when probability ratios are available (Tibshirani et al., 2019). Off-policy extensions address stochastic policies, continuous actions, and sequential decisions (Taufiq et al., 2022; Zhang et al., 2023; Foffano et al., 2023). In particular, Zhang et al. (2023) subsample logged observations using policy-probability ratios and discrete pseudo-action matching. Candidate completion instead compares the logged action's label with sampled competitors, obtaining exact target-distributed observations for continuous-action selection policies through sampling and scoring. The resulting samples support standard conformal ranks and higher-confidence tolerance bounds (Park et al., 2022; Wilks, 1941).

Reusing logged outcomes through exact sampling. Replay and rejection sampling evaluate policies using logged observations (Li et al., 2010; Dudík et al., 2012; Mandel et al., 2016). Bounded unbiased random weights can implement exact sampling when deterministic weights are unavailable (Schmon et al., 2019). Candidate completion supplies the required random weight from the deployed selection rule. Keeping the logged action in the competition preserves its connection to the observed outcome; applying the rule along the trajectory also accounts for changes in later states.

Predicting outcomes with learned models. Dynamics models and value functions support short-horizon predictions (Yu et al., 2020; Lu et al., 2022); Bayesian model-value methods construct conservative estimates during learning (Jeong et al., 2023). Our method separates fitting from calibration. A trained model supplies the score or predictor, and completion calibrates its errors under the deployed policy. The guarantees therefore do not depend on a particular training algorithm.

3 Problem Setup

Available data. The behavior policy $\mu(\cdot \mid s)$ samples an action at state s. We observe n independent episodes, each starting from the same distribution of initial states, denoted by ρ_0. An episode is one run from an initial state until termination. We use its first H transitions, or all available transitions if

it terminates earlier. This trajectory prefix is denoted by τ. Its state, action, and reward at step t are S_t, A_t, R_t; T is the number of transitions before natural termination. After termination we pad the prefix to length H with a fixed terminal state and zero rewards.

The behavior policy is preserved, so we can draw fresh actions at each logged state without executing them. These draws must use the same conditional distribution as the logged actions. The target policy shares the environment, reward rule, termination rule, and initial-state distribution. Models are fitted on separate data. The policy, predictor, outcome definition, and calibration settings are fixed before calibration. Episodes are independent; transitions within an episode can be dependent.

The policy to calibrate. Let $g(s, a)$ be the score used to rank actions, with larger scores preferred. At state s, draw $M \geq 2$ independent candidates $\widetilde{A}_0, \ldots, \widetilde{A}_{M-1}$ from $\mu(\cdot \mid s)$. Here the subscript indexes candidates, not time. Choose an index J uniformly among the highest-scoring candidates, and execute $\widetilde{A}_J$. Write $\nu_M(\cdot \mid s)$ for the distribution of this selected action.

The target policy π uses selection with probability $\epsilon \in [0, 1]$ and an ordinary behavior action otherwise:

$$\pi(\cdot \mid s) = (1 - \epsilon)\mu(\cdot \mid s) + \epsilon\nu_M(\cdot \mid s), \qquad C = 1 + \epsilon(M - 1). \tag{1}$$

Thus $\epsilon = 1$ selects at every step. The constant C normalizes the acceptance rule below and determines how many trajectories are retained.

The prediction to calibrate. Let $Y_H(\tau)$ be the outcome, such as the discounted reward over H steps, and let $G(S_0, A_0)$ predict it. The ranking score g and outcome predictor G can be different. Define the signed error

$$E_H(\tau) = G(S_0, A_0) - Y_H(\tau). \tag{2}$$

Positive error means overprediction. For an allowed error probability $\alpha \in (0, 1)$, calibration chooses a correction q so that the lower bound $G - q$ covers a new target-policy outcome with probability at least $1 - \alpha$. A smaller correction gives a higher, more informative lower bound when coverage is maintained.

4 Candidate Completion

The method has three steps: complete the candidate set around each logged action, retain trajectories that pass the selection tests, and calibrate using their prediction errors. Algorithm 1 gives the procedure. Exactness concerns the retained data distribution; calibration controls the predictor's remaining error. Competitions, tie choices, and acceptance uniforms use independent randomness across steps and episodes.

4.1 Complete the candidates around a logged action

At a logged state s, place the observed action a at candidate index zero and draw $M - 1$ fresh actions from $\mu(\cdot \mid s)$. Apply the same score and tie-breaking rule as the target policy. All candidate labels have the same sampling role; only the logged action already has an observed outcome.

Let $I = 1$ when selection chooses the logged action and $I = 0$ otherwise. Its random weight W and acceptance probability κ are

$$W = (1 - \epsilon) + \epsilon MI, \qquad \kappa = \frac{W}{C}. \tag{3}$$

A selected logged action always passes, because $W = C$. An unselected one passes with probability $(1-\epsilon)/C$, accounting for the behavior branch of the target policy. Under pure selection, only selected actions pass. In the introductory example, a logged action a passes exactly when the fresh competitor is smaller, so larger actions are retained more often.

Why the accepted actions have the right distribution. Let $p_s(a)$ be the probability that action a is selected against $M - 1$ fresh competitors. Since all M candidate labels play the same role, selection

increases an action's relative frequency by $Mp_s(a)$. The target-to-behavior probability ratio $\ell_s(a)$ is therefore

$$\ell_s(a) = \frac{d\pi(\cdot \mid s)}{d\mu(\cdot \mid s)}(a) = (1-\epsilon) + \epsilon M p_s(a) = \mathbb{E}[W \mid s, a]. \tag{4}$$

The derivative denotes a density ratio when densities exist, and the corresponding distribution ratio more generally. The expectation $\mathbb{E}$ averages the fresh competition. Completion samples the bounded weight W, rather than evaluating $\ell_s(a)$. For a logged action $A \sim \mu(\cdot \mid s)$ and any action set B,

$$\Pr_{\mu}(A \in B, \text{ passes} \mid s) = \frac{1}{C}\pi(B \mid s). \tag{5}$$

Thus accepted actions follow the target distribution. Their recorded next states and rewards already have the correct distribution conditional on those actions.

4.2 Retain complete trajectories

Test each logged action at its actual state, and retain a prefix only when every test passes. This preserves the entire sequence of states and outcomes. After early termination, add independent tests with probability $1/C$ at the remaining padded steps. These supply the same total scaling for short and long prefixes, preserving the target policy's termination distribution.

Algorithm 1 Candidate completion: logged trajectories to prediction bounds

Require: n logged episodes; sampler μ; score g; predictor G; outcome Y_H; M, ϵ, H, α

Set $C \leftarrow 1 + \epsilon(M-1)$; initialize the retained-error list $\mathcal{E} \leftarrow \varnothing$.
for each of the n logged trajectory prefixes τ **do**
  for $t = 0, \ldots, H-1$ **do**
    if the episode is active at step t **then**
      Add $M-1$ fresh behavior candidates to the logged action A_t.
      Set $I_t \leftarrow 1$ if selection chooses A_t, and $I_t \leftarrow 0$ otherwise.
      Set $\kappa_t \leftarrow [(1-\epsilon) + \epsilon M I_t]/C$.
    else
      Set $\kappa_t \leftarrow 1/C$.
    end if
    Draw U_t uniformly on $[0,1]$; reject this prefix if $U_t > \kappa_t$.
    On rejection, skip the remaining steps and continue with the next prefix.
  end for
  Append $G(S_0, A_0) - Y_H(\tau)$ to $\mathcal{E}$.
end for
Compute q from $\mathcal{E}$ using Equation (7); **return** the bound $G - q$.

Theorem 1 (Exact target trajectories)**.** *Under the conditions of Section 3, Algorithm 1 retains target-policy trajectories. Let P_π^H be the distribution of a length-H, terminal-padded target trajectory. For every trajectory event D,*

$$\Pr_{\mu}(\tau \in D, \textit{ retained}) = C^{-H} P_\pi^H(D). \tag{6}$$

The retained count N follows $\mathrm{Binomial}(n, C^{-H})$*. Conditional on N, the retained trajectories are independent samples from P_π^H.*

Proof idea and data budget. Equation (5) supplies a target action and a factor $1/C$ at each active step. The following physical transition is shared by both policies. Repeating the argument along the logged states, then including padded tests, gives exactly H factors. Appendix C provides the proof. The expected retained count is nC^{-H}, known before processing the data. Each prefix needs at most $H(M-1)$ extra action samples and no new environment transitions.

Selection can also vary by step. If M_t and ϵ_t are the candidate count and selection probability fixed for step t, let $C_t = 1 + \epsilon_t(M_t - 1)$. The same construction retains trajectories at rate $\prod_{t=0}^{H-1} C_t^{-1}$.

4.3 CALIBRATE THE RETAINED PREDICTION ERRORS

Suppose $N = m$ errors are retained. Sort them as $E_{(1)} \le \cdots \le E_{(m)}$, where parentheses indicate rank. Choose the rank k_m and correction q by

$$k_m = \lceil (m+1)(1-\alpha) \rceil, \qquad q = \begin{cases} E_{(k_m)}, & k_m \le m, \\ +\infty, & k_m > m. \end{cases} \tag{7}$$

The ceiling rounds upward. The extra observation in $m+1$ accounts for the future outcome: 99 retained errors at 90% coverage require the 90th smallest error. If the rank is unavailable, the method returns no finite bound. Replacing a negative q by zero lowers the bound and preserves coverage.

Theorem 2 (Calibrated prediction)**.** *For the fixed predictor and target in Section 3, Equation* (7) *gives*

$$\Pr(G(S_0^*, A_0^*) - q \le Y_H(\tau^*)) \ge 1 - \alpha, \tag{8}$$

where τ^* is an independent target-policy trajectory, with initial state S_0^* and action A_0^*. Probability averages over calibration and the future trajectory.

Why the rank works. The retained errors and future error have the same distribution. The future error is equally likely to occupy any rank among the $m+1$ observations when there are no ties; ties preserve the inequality. This holds at every retained count, so averaging over the random N gives Theorem 2.

A higher-confidence option. Ordinary calibration controls coverage on average across calibration datasets. A *tolerance bound* additionally requires that a specified fraction p of target outcomes be covered with confidence $1-\delta$ over those datasets. Here δ is the probability of failing that requirement. Choose the smallest $k \in \{k_m, \ldots, m\}$ satisfying

$$\Pr(\text{Binomial}(m, p) \ge k) \le \delta, \tag{9}$$

and set $q_{\text{tol}} = E_{(k)}$, or $+\infty$ if none exists. We use $p = 0.9$ and $\delta = 0.025$: at least 97.5% of calibration datasets produce a bound covering at least 90% of target outcomes. Appendix C proves this statement.

4.4 WEIGHT TRAJECTORIES TO REUSE THE FULL LOG

Instead of retaining a subset, weighted CC assigns a weight to every logged trajectory. Repeat each active-step competition $K \ge 1$ times. Let $W_{i,t,r}$ be the weight in Equation (3) for trajectory i, step t, and repetition r; let T_i be its termination time. Define

$$\omega_i = \prod_{t < \min(T_i, H)} \left(\frac{1}{K} \sum_{r=1}^{K} W_{i,t,r} \right). \tag{10}$$

Average at each step, then multiply across steps. Each trajectory contributes one error and one weight. The weight lies in $[0, C^H]$, and its mean conditional on the trajectory is the target-to-behavior trajectory ratio.

For logged error $E_i = E_H(\tau_i)$, choose the smallest threshold x satisfying

$$\sum_{i=1}^{n} \omega_i \mathbf{1}\{E_i \le x\} \;\ge\; (1-\alpha) \left(\sum_{i=1}^{n} \omega_i + C^H \right), \tag{11}$$

where $\mathbf{1}$ is one when its condition holds and zero otherwise. Set the weighted correction to this threshold, or $+\infty$ if none exists. The extra C^H reserves the largest possible weight for the future observation. Appendix C.3 shows that this rule preserves Theorem 2. Ordinary CC also supplies the independent target samples required by the tolerance option.

Reuse a shared candidate pool. With $R \ge M-1$ fresh competitors at a state, average the selection probability over all groups of $M-1$ competitors. Every group is a valid completion, so their average has the same exact mean. Counting lower and tied scores computes it without enumerating the groups (Appendix C.3). Here R counts fresh actions and K counts independent competitions; neither increases the number of observed trajectories.

Approximate behavior samplers. For a fitted replacement sampler, let η bound the total-variation distance at every active state: the largest difference in probability assigned to an action event. Appendix C.4 converts η into a trajectory discrepancy bound Δ, giving coverage at least $1 - \alpha - \Delta$. It distinguishes replacing the sampler only during completion from also replacing it at deployment. An implementation check uses the fact that the logged action is selected with probability $1/M$ under exact completion; first-step counts across independent episodes can be compared with this binomial prediction.

5 EXPERIMENTS

We first test prediction bounds for directly observed rewards under repeated action selection. We then isolate the benefit of exact completion against estimated weighting at the same sampling budget. All experiments target 90% coverage.

5.1 EVALUATION SETUP AND HOW TO READ THE RESULTS

Tasks, policies, and outcomes. We use HalfCheetah-v5, Hopper-v5, and Walker2d-v5 (Todorov et al., 2012; Towers et al., 2025). Three soft actor-critic (SAC) checkpoints per environment define low-, medium-, and high-quality behavior policies (Haarnoja et al., 2018). These nine environment/quality groups, each with three model-fitting seeds, give 27 fixed configurations. A five-model dynamics ensemble and an implicit Q-learning (IQL) value function define a three-step selection score $g = G_{\min}$ (Kostrikov et al., 2022).

The reward study keeps this score fixed and selects the best of M actions at *every* step. It uses $(M, H) \in \{(2,3), (4,2), (2,5)\}$, always with $\epsilon = 1$. Its prediction target contains only observed rewards:

$$Y_{R,H}(\tau) = \sum_{t=0}^{H-1} \gamma^t R_t, \qquad G_{R,H}(s,a) = \min_{e \in \mathcal{F}} \sum_{t=0}^{H-1} \gamma^t \widehat{R}_t^e(s,a). \tag{12}$$

Here $\gamma = 0.99$, $\mathcal{F}$ is the set of five fitted dynamics models, and $\widehat{R}_t^e$ is model e's reward prediction after initial action a. Both sums stop at termination. The predictor has no final-state value term; the selection score is unchanged. Appendix A.2 specifies the model continuation and data collection.

Each group supplies three independently generated pools of 2,000 behavior episodes. Its three fits and three settings share those source pools, with independent completion randomness. Each of the resulting 243 evaluations uses 2,000 separate target-test paths and 2,000 oracle-calibration paths. This gives 486,000 target-test paths; all models and experimental settings are fixed before collection.

Comparisons and metrics. *Behavior calibration* uses logged errors without correcting the policy change. *Score-CDF weighting* estimates the cumulative distribution function (CDF) of candidate scores and converts it into trajectory weights. Both weighting methods use the same competitor scores. The *target oracle* calibrates on extra target-policy outcomes; CC uses only the logged outcomes. A learned-ratio comparator and its training diagnostics are included in Appendix B.1.

Coverage is the fraction of outcomes at or above the lower bound; 90% is the target, not the largest possible value. A smaller *correction* raises the bound. Lower *pinball loss* balances missed outcomes against unnecessarily low bounds. We report all groups because opposite calibration errors can cancel in an overall average. Appendix A.2 defines the normalization and pool-level uncertainty calculations.

5.2 CALIBRATING OBSERVED REWARDS UNDER REPEATED SELECTION

CC corrects reward miscalibration within individual environment/quality groups. With best-of-four selection at both steps, Walker2d-high coverage rises from 78.67% with behavior calibration to 90.45% with CC, close to the target oracle's 90.24% (Table 1). Score-CDF weighting reaches 88.71% at the same three-competitor budget. CC's gain over behavior calibration is positive in each of the three independent pools, ranging from 8.73 to 13.38 percentage points after averaging fits.

The same mechanism also reduces overcoverage. In the three-step best-of-two setting, Hopper-low changes from 94.27% to 91.24%, near the oracle's 91.09%. For five-step best-of-two, Walker2d-low

Table 1: **Calibration of observed rewards under repeated best-of-four selection.** Coverage targets 90%; both steps use selection. Each group averages three fixed fits and three independent pools of 2,000 logged episodes, with 2,000 test paths per run. CDF and ordinary CC use three competitors per active state. The oracle uses 2,000 extra target outcomes. The gap row averages absolute group-level coverage differences from that reference, in percentage points (pp). Bold marks the best displayed summary among methods without extra target outcomes.

Environment	Quality	Behavior	Score-CDF	**CC (Ours)**	Oracle
HalfCheetah	Low	90.54	90.47	90.01	89.99
	Medium	90.22	90.09	89.70	90.13
	High	92.59	90.67	90.95	89.69
Hopper	Low	96.33	94.86	94.29	94.19
	Medium	89.48	93.35	94.41	93.77
	High	89.04	91.32	91.68	93.12
Walker2d	Low	91.32	89.48	89.53	90.42
	Medium	90.91	91.18	91.82	90.77
	High	78.67	88.71	90.45	90.24
Mean coverage (%)		89.90	91.13	91.43	91.37
Mean gap to oracle (pp) ↓		2.96	0.81	**0.67**	–
Normalized pinball loss ↓		0.1655	**0.1597**	0.1603	0.1593

changes from 83.75% to 91.26%. Appendix B.1 reports every group for all three settings, rather than only these examples.

Across all 27 setting/group averages, the mean absolute coverage difference from the separate target oracle is 0.69 percentage points for CC, versus 2.12 for behavior calibration. This descriptive comparison measures agreement with the extra-data reference. The aggregate coverage alone is similar because undercoverage and overcoverage partly cancel. Within the two-step best-of-four setting, normalized pinball loss also falls from 0.1655 to 0.1603 relative to behavior calibration; the CDF baseline gives 0.1597.

Why completion follows the required correction. Selection moves the target error's 90th percentile both upward and downward relative to behavior. Across all 27 setting/group pairs, these shifts range from -0.52 to $+0.50$ behavior-error standard deviations. The shifts recovered from retained errors have correlation 0.987 with the target shifts and mean absolute difference 0.030 standard deviations (Figure 1). This explains why calibration must follow the selected policy, rather than always increasing the correction.

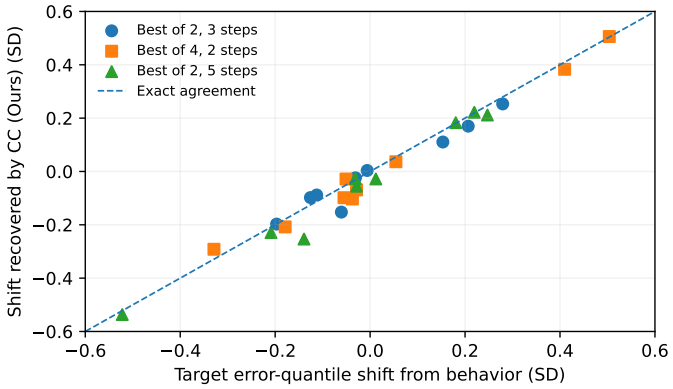


Figure 1: **Completion recovers selection-induced changes in reward prediction errors.** Each point averages three fits and three independent pools for one setting/group pair. Axes show the change in the 90th error percentile relative to behavior, scaled by behavior-error standard deviation (SD). All 27 pairs are shown; the dashed line is exact agreement. These empirical-quantile shifts explain the mechanism, while calibration itself uses the exact ranks in Section 4.

5.3 EXACT COMPLETION MATTERS AT FINITE CANDIDATE BUDGETS

At equal sampling budgets, completion avoids undercoverage from estimated score weights. In the introductory continuous-action example, weighted CC covers 90.37–91.67% for $M = 4, 8, 16$, while CDF weighting covers 86.56–87.29% (Table 2). Each comparison uses the same 500 logged actions, the same $R = M - 1$ competitors per action, and the same weighted rank. Only the weight construction changes.

Table 2: **Exact completion avoids finite-budget undercoverage.** In the continuous-score example, each row uses 1,000 independent datasets of 500 logged actions. Both weighted methods use the same competitors and rank. Coverage (%) and expected pinball loss are evaluated analytically. Bold gains have paired 95% Monte Carlo intervals above zero; bold losses are the smaller displayed values, including ties. All bounds are finite.

M	R	CDF cov.	**CC (Ours)**	CC–CDF gain (pp)	CDF loss ↓	CC loss ↓
2	1	90.19	90.19	+0.00 [0.00, 0.00]	**0.03089**	**0.03089**
4	3	87.29	90.37	**+3.08 [2.97, 3.18]**	0.01894	**0.01881**
8	7	86.56	90.80	**+4.24 [4.08, 4.40]**	0.01068	**0.01057**
16	15	86.95	91.67	**+4.72 [4.48, 4.97]**	0.00571	**0.00565**

Each row averages 1,000 independently generated calibration datasets. Actions are uniform on $[0, 1]$, the score and predictor are $2a$, and the outcome is a. The selected action has CDF x^M, so coverage and expected prediction loss are evaluated analytically at every fitted cutoff. CC's paired coverage gains are 3.08, 4.24, and 4.72 percentage points; its prediction loss is also slightly smaller. At $M = 2$, the same-pool estimators coincide.

The full comparison uses 16,000 independent underlying datasets: $n \in \{500, 2000\}$, $M \in \{2, 4, 8, 16\}$, continuous or eight-bin scores, and $R \in \{M - 1, 16, 64\}$. Each dataset is evaluated with errors a and $-a$. Tied scores use the correct tie-aware probability formula. Completion also corrects the opposite direction of shift; larger candidate budgets bring the two methods closer. Appendix B.2 gives both error directions, exact-ratio and target-data references, losses, and finite-bound counts.

5.4 SELECTION STRENGTH AND COMPLEMENTARY CONTROL RESULTS

A first-action sweep confirms the effect across learned scores. CC maintains 90.4–91.9% coverage as M increases from 2 to 16, while behavior calibration falls from 88.3% to 82.6% (Table 3). This experiment selects the first action and then follows behavior for two steps. It calibrates three observed rewards plus a fixed IQL final-state value, with 500 logged episodes per configuration and 54,000 target-test trajectories overall.

Table 3: **Completion maintains coverage as selection strengthens.** Coverage targets 90%; selection acts only at the first step. Each row averages all 27 configurations with 500 calibration episodes each. CC and score-CDF weighting use $M - 1$ fresh competitors. Gains over behavior are percentage points, calculated before rounding; bold gains have paired 95% confidence intervals (CI) above zero. All bounds are finite.

M	Behavior	Score-CDF	**Ordinary CC (Ours)**	Gain over behavior [95% CI]
2	88.32	89.93	90.41	**+2.10 [0.4, 3.7]**
4	86.64	89.81	90.40	**+3.76 [0.9, 6.4]**
8	84.19	89.61	90.87	**+6.69 [2.2, 10.7]**
16	82.61	91.42	91.86	**+9.25 [4.1, 14.6]**

At $M = 4, 8$, CC also lowers pinball loss relative to behavior calibration by 0.017 and 0.030, with paired intervals excluding zero. Appendix B.3 includes equal-budget CDF comparisons. A separate critic-based study gives smaller corrections in all 15 fits under two-step best-of-two selection, with a 14.6% pooled reduction (Table 16); Appendix A.6 specifies its predictor.

5.5 Data requirements and validation

In the observed-reward study, all 243 ordinary bounds are finite with 2,000 logged episodes. The three settings retain means of 252.0, 126.1, and 62.7 trajectories, close to the predicted 250, 125, and 62.5. With 500 episodes, the five-step setting yields finite ordinary bounds in 79 of 81 runs. Weighted completion with 16 competitors per step increases its effective sample size to 412.6 at $n = 2{,}000$. Table 4 shows the benefit of additional candidate comparisons; Appendix B.1 gives all budgets and methods. The known factor M^{-H} makes the required data budget explicit before collecting outcomes.

Table 4: **More candidate comparisons improve data reuse.** Both methods use the same 2,000 logged outcomes per run. Ordinary CC uses $M - 1$ competitors per step; weighted CC uses $R = 16$, with no additional environment outcomes. Coverage targets 90%. Loss is normalized pinball loss; bold marks the smaller displayed loss in each pair. Count is retained trajectories or weighted effective sample size (ESS). All bounds are finite.

		Ordinary CC (Ours)			Weighted CC (Ours), $R = 16$		
M	H	Count	Cov. (%)	Loss ↓	ESS	Cov. (%)	Loss ↓
2	3	252.0	91.25	0.1608	768.8	91.32	**0.1604**
4	2	126.1	91.43	0.1603	317.1	91.47	**0.1596**
2	5	62.7	92.72	0.1606	412.6	92.95	**0.1592**

Controlled trajectory experiments independently check exactness, including score ties and early termination. In the enumerated five-state problem, ordinary CC covers 91.0% across 2,000 independent calibrations, compared with 87.5% when post-termination tests are removed. Tolerance calibration misses its 90%-coverage requirement in 1.25% of calibrations, within its 2.5% allowance. Appendix B includes complete path-distribution checks and sampler perturbations.

6 Conclusion

Candidate completion keeps the action whose outcome is known and samples only its competitors. It converts logged trajectories into target-policy samples and calibrated bounds under independent reset episodes and the stated sampling conditions. The experiments show corrected reward miscalibration under repeated selection and a finite-budget reliability advantage over estimated score weights. Together, these results provide a practical route to policy-specific calibration using a preserved sampler and previously observed outcomes.

AI USE STATEMENT

Generative AI tools played a limited, assistive role in this work. They were used for language editing and polishing of the manuscript, to help find some related papers, and for partial assistance with writing experiment code. The research idea, the mathematical results and proofs, and the experimental design are the authors' own. All AI-assisted text and code were reviewed and checked by the authors, and every cited reference was verified against the original source. The authors take full responsibility for the content of the paper.

ETHICS STATEMENT

The experiments use simulated control tasks and explicitly specified synthetic models. They contain no human-subject data or personal information. The reported prediction bounds concern the stated trajectory outcomes and covered fraction of the target distribution. They should be interpreted at that level when used in downstream decision systems.

REPRODUCIBILITY STATEMENT

Appendix A describes the data, models, targets, and metrics. Appendix B provides complete comparisons, and Appendix C proves the stated guarantees. The anonymous supplementary archive contains source code, tests, experimental records, and scripts for the paper's tables and figures. Appendix D distinguishes analyses that run directly from the archive from full experiment regeneration, which also requires the generated trajectory and checkpoint collection.

REFERENCES

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3580–3602. PMLR, 2025. URL https://proceedings.mlr.press/v267/beirami25a.html.

Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Sample-efficient nonstationary policy evaluation for contextual bandits. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012. URL https://www.auai.org/uai2012/papers/254.pdf.

Daniele Foffano, Alessio Russo, and Alexandre Proutiere. Conformal off-policy evaluation in Markov decision processes. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 3087–3094. IEEE, 2023. doi: 10.1109/CDC49753.2023.10383469.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Seyed Kamyar Seyed Ghasemipour, Dale Schuurmans, and Shixiang Shane Gu. EMaQ: Expected-max Q-learning operator for simple yet effective offline and online RL. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3682–3691. PMLR, 2021. URL https://proceedings.mlr.press/v139/ghasemipour21a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Jihwan Jeong, Xiaoyu Wang, Michael Gimelfarb, Hyunwoo Kim, Baher Abdulhai, and Scott Sanner. Conservative Bayesian model-based value expansion for offline policy optimization. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=dNqxZgyjcYA.

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.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018. doi: 10.1080/01621459.2017.1307116.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670. ACM, 2010. doi: 10.1145/1772690.1772758.

Cong Lu, Philip J. Ball, Jack Parker-Holder, Michael A. Osborne, and Stephen J. Roberts. Revisiting design choices in offline model-based reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zz9hXVhf40.

Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. Offline evaluation of online reinforcement learning algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. doi: 10.1609/aaai.v30i1.10312. URL https://ojs.aaai.org/index.php/AAAI/article/view/10312.

Kwanyoung Park, Seohong Park, Youngwoon Lee, and Sergey Levine. Scalable offline model-based RL with action chunks. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1056d17e239b7f82fbfe47170cced506-Paper-Conference.pdf.

Sangdon Park, Edgar Dobriban, Insup Lee, and Osbert Bastani. PAC prediction sets under covariate shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=DhP9L8vIyLc.

Sebastian M. Schmon, Arnaud Doucet, and George Deligiannidis. Bernoulli race particle filters. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2350–2358. PMLR, 2019. URL https://proceedings.mlr.press/v89/schmon19a.html.

Muhammad Faaiz Taufiq, Jean-Francois Ton, Rob Cornish, Yee Whye Teh, and Arnaud Doucet. Conformal off-policy prediction in contextual bandits. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31512–31524. Curran Associates, Inc., 2022. doi: 10.52202/068431-2285. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cc84bfabe6389d8883fc2071c848f62a-Abstract-Conference.html.

Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candès, and Aaditya Ramdas. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/8fb21ee7a2207526da55a679f0332de2-Abstract.html.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standardized interface for reinforcement learning environments. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-4916. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d7ff1795e8527f6443371c3933bdb52b-Paper-Datasets_and_Benchmarks_Track.pdf.

S. S. Wilks. Determination of sample sizes for setting tolerance limits. *The Annals of Mathematical Statistics*, 12(1):91–96, 1941. doi: 10.1214/aoms/1177731788.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.

Yingying Zhang, Chengchun Shi, and Shikai Luo. Conformal off-policy prediction. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 2751–2768. PMLR, 2023. URL https://proceedings.mlr.press/v206/zhang23c.html.

A Experimental Details

Reading guide. Sections A.2–A.3 specify the reward application and the controlled comparison. Sections A.4–A.7 give the models shared by all MuJoCo experiments and the model-value protocols. Section B reports the reward application and controlled comparison first, then the complementary selection and calibration studies. Section C explains the sampling and calibration proofs; Section D gives the code and reproduction workflow.

A.1 Notation at a glance

Symbol	Meaning
μ, π	Behavior policy that collected data, and deployed target policy.
$s, a;\ S_t, A_t, R_t$	A state and action; the state, action, and reward at time t.
ρ_0, τ, T, H	Reset-state distribution, trajectory prefix, termination time, and fixed horizon.
$M, \widetilde{A}_j, J$	Number of candidates, candidate j, and selected candidate index.
ϵ, C	Selection probability and acceptance normalization $C = 1 + \epsilon(M-1)$.
g, G, Y_H, E_H	Selection score, outcome predictor, outcome, and error $E_H = G - Y_H$.
$G_{R,H}, Y_{R,H}$	Reward-only predictor and observed discounted reward sum over H steps.
$p_s(a), \ell_s(a)$	Probability that logged action a is selected, and its target/behavior ratio.
n, N, m	Number of logged episodes, retained count, and a realized value of that count.
I, W, κ	Indicator that the logged action is selected, random weight, and acceptance probability W/C.
α, q, k_m	Allowed uncovered fraction, correction, and rank among m retained errors.
$p, \delta, q_{\mathrm{tol}}$	Desired covered fraction, tolerance failure probability, and tolerance correction.
K, ω_i	Number of independent completions per step and weight of logged trajectory i.
R, b, u, ϕ_R	Shared-pool size; numbers of lower and tied competitor scores; averaged selection probability.
η, Δ	Bounds on sampler discrepancy and the resulting change in coverage.

A.2 Repeated-selection application with observed rewards

This study uses the same nine behavior checkpoints and 27 fitted dynamics/IQL bundles as the model-value experiments. All networks remain fixed. The selection score is the three-step $G_{\min}$ of Section 5.1, including its IQL terminal value; the *outcome being calibrated* is instead the fully observed reward sum $Y_{R,H}$ in Equation (12). Its predictor $G_{R,H}$ rolls each elite dynamics model forward for H steps from the observed initial action, uses the deterministic mean of the fixed IQL actor for later model actions, and sums predicted rewards with zero terminal value. Physical continuation selects the best of M fresh behavior actions at every active step. Model mismatch and the difference between model and physical continuation both enter the prediction error.

Independent data and fixed settings. The configuration fixes $(M, H) = (2, 3), (4, 2), (2, 5)$, $\epsilon = 1$, $\alpha = 0.1$, $\gamma = 0.99$, and three source pools before collection. Each environment/quality group provides three independently reset pools of 2,000 behavior prefixes of maximum length five. The three fitted models and three settings share these physical source prefixes. Thus there are 27 independent source pools and 54,000 unique logged episodes, not 243 independent source collections. Each fit/setting/pool evaluation has distinct target-test and target-oracle reset and action streams, with 2,000 paths in each stream. The complete grid contains 243 evaluations, 486,000 target-test paths, and 486,000 oracle paths. The master seed is 202609252; named streams separate source collection, test collection, oracle collection, and competition draws.

Rewards are discounted until natural termination, with zero rewards thereafter. Prefixes never cross a reset. No collected application prefix terminates naturally within the tested horizons; the separate tabular experiments exercise that case. Checkpoint hashes, normalization statistics, and 16 stored scorer predictions per evaluation are checked before use. The maximum observed scorer replay difference is 1.8311×10^{-4}, below the fixed 10^{-3} tolerance.

Matched competitors and calibration. At each active logged state, a shared pool of 64 competitors supplies budgets $R \in \{M-1, 16, 64\}$. Ordinary completion uses the first $M-1$ scores, uniform tie-breaking, and the trajectory acceptance rule. Shared-pool completion averages all size-$M-1$

competitions through Equation (25). Both weighting methods multiply step weights and use maximum future weight M^H. With ties, let $\widehat{F}_-$ and $\widehat{F}_+$ be the fractions of competitor scores strictly below and at most the logged score. The plug-in selected-action ratio is

$$\widehat{\ell}_{\mathrm{CDF}} = \sum_{j=0}^{M-1} \widehat{F}_+^{M-1-j} \widehat{F}_-^{j}. \tag{13}$$

Here j indexes the terms in the sum. Without ties this is $M\widehat{F}^{M-1}$, and for $M = 2$ it equals the shared-pool completion estimator, including ties. This correct tie-aware expression avoids attributing a tie-handling error to weight estimation.

Each rule is evaluated at the first 500 and all 2,000 logged episodes. Ordinary and weighted ranks are exact order statistics, followed by nonnegative correction clipping. An unsupported rank remains $+\infty$; finite-only loss and coverage summaries state their denominator. The oracle applies the same ordinary rule to an independent target sample of the corresponding size. Every run stores predictions, observed rewards, residuals, weights, retained masks, and random seeds.

Learned-ratio comparison. A separate balanced classifier distinguishes behavior actions from best-of-M actions at 8,192 states from the fitting split. Its inputs concatenate normalized states and actions. It has two width-128 ReLU layers, with Adam at 3×10^{-4}, weight decay 10^{-4}, batch size 256, 25 epochs, and a 20% holdout. If $D(s, a)$ is its predicted target-class probability, the estimated step ratio is $D/(1 - D)$, clipped to $[0, M]$. Ratios are multiplied over the trajectory and used with the same weighted rank. There are 54 fits: one per fitted configuration and candidate count; the $M = 2$ classifier is shared across horizons. This baseline needs additional action-only training data, with $8{,}192(M + 1)$ behavior draws and $8{,}192M$ selection-score evaluations per fit. These costs are separate from the equal-budget CDF comparison. No outcome label or target-test error trains the classifier.

Metrics and uncertainty. Coverage, loss, and correction are computed on paired target-test paths. Application corrections and losses are normalized by the standard deviation σ of the full 2,000 behavior errors for that run. Three fit results are averaged within each source pool. Pool-level bootstrap ranges resample three pools within each of the nine fixed groups, keeping fits together; they describe pool variation conditional on this benchmark and its trained models. There are 10,000 draws. Per-pool values are also supplied so the three independent repetitions are visible.

For the mechanism analysis, empirical 90th percentiles use linear interpolation. Their shifts are $(q_{\mathrm{target}} - q_{\mathrm{behavior}})/\sigma$ and $(q_{\mathrm{retained}} - q_{\mathrm{behavior}})/\sigma$, calculated per run and then averaged within a setting/group. Here these three q's denote empirical percentiles used only for diagnosis, not the conformal correction. The target shift uses the independent test distribution; the oracle remains a separate calibration stream. The correlation, mean absolute shift difference, and mean absolute coverage gap to the oracle are descriptive summaries added after the run. They do not select a policy, fit, or reported group.

A.3 Controlled comparison with known target distributions

The controlled comparison uses $n \in \{500, 2000\}$, $M \in \{2, 4, 8, 16\}$, and continuous or eight-bin scores. Each of these 16 settings has 1,000 independently generated calibration datasets. Actions are uniform on $[0, 1]$; the predictor is $G(a) = 2a$. Each dataset is evaluated with outcomes $Y(a) = a$ and $Y(a) = 3a$, giving errors a and $-a$. The two outcomes share the same actions and competitions. A score is either increasing continuously with a or equal to its bin index; selection breaks maximal-score ties uniformly.

For continuous scores the target action CDF is $F_A(x) = x^M$ on $[0, 1]$. For $B = 8$ bins, and $k/B \le x \le (k + 1)/B$, it is

$$F_A(x) = (k/B)^M + B(x - k/B)\big[((k + 1)/B)^M - (k/B)^M\big],$$

where $k \in \{0, \ldots, B - 1\}$ is the bin index. Thus exact coverage and expected pinball loss can be integrated for each fitted cutoff. Ordinary completion, shared-pool weighting, the tie-aware CDF plug-in, the analytically known ratio, and independent target calibration are compared on the same

datasets. Candidate budgets are $R \in \{M-1, 16, 64\}$. Corrections are not clipped, allowing both error directions. All 320,000 stored method/outcome cutoffs in this grid are finite. Paired 95% intervals are the mean difference plus or minus 1.96 Monte Carlo standard errors across the 1,000 independent datasets; they quantify simulation uncertainty, not a per-dataset tolerance guarantee.

A.4 Behavior policies and data splits

The SAC policies used for data collection have stochastic tanh-Gaussian actors. Table 5 identifies the checkpoints. Quality levels are relative to each environment and are not benchmark-normalized scores. Checkpoint returns were used to define the levels before downstream calibration.

Table 5: Behavior checkpoints. Each entry gives the training step and evaluation return.

Environment	Low	Medium	High
HalfCheetah	100k (4969.0)	300k (7561.4)	900k (11027.1)
Hopper	100k (1143.1)	200k (3083.9)	400k (3415.5)
Walker2d	100k (475.5)	500k (1863.5)	900k (3538.2)

Each checkpoint generated 1,000 complete reset episodes. For fitting seed $z \in \{0, 1, 2\}$, a partition with seed $2026 + z$ assigns 500 episodes to fitting. The remaining 500 are development partitions of sizes 50, 300, 50, and 100, for ratio, calibration, target, and evaluation development respectively. Only the fitting episodes train the models. A separate 500-episode pool per environment/quality cell supplies final calibration. All three fitted models use that pool with independent completion randomness; final target-test streams are separate for every fitted policy.

Collection records the checkpoint identifier, preprocessing, action noise, reset state, step identifier, and termination flags. Replaying recorded action noise with the preserved sampler has maximum action discrepancy 1.93×10^{-5}. This checks numerical reconstruction; the exact guarantee concerns sampling from the same behavior distribution, as stated in Section 3. No reported target-test outcome is used to select a model, policy, horizon, or calibration rule.

A.5 Predictors, outcomes, and termination

Let $\mathcal{F}$ be the set of five selected dynamics models. For a model $e \in \mathcal{F}$, hatted states, rewards, and activity indicators are predictions made by that model. Starting with $\widehat{S}_0^e = s$ and first action a, define

$$\widehat{Y}_3^e(s,a) = \sum_{t=0}^{2} \gamma^t \widehat{R}_t^e + \gamma^3 \widehat{B}_3^e V_{\mathrm{IQL}}(\widehat{S}_3^e), \tag{14}$$

$$G_{\min}(s,a) = \min_{e \in \mathcal{F}} \widehat{Y}_3^e(s,a), \qquad G_R(s,a) = \min_{e \in \mathcal{F}} \sum_{t=0}^{2} \gamma^t \widehat{R}_t^e. \tag{15}$$

The score and model-value predictor are both $G_{\min}$. The reward-only predictor is G_R. Subsequent actions inside the model use the deterministic mean of the fixed IQL actor; physical continuation follows the deployed policy. The complete difference between the prediction and the physical outcome enters the calibration error.

The terminating transition keeps its reward; all later rewards and the final-state value are zero. Time limits are distinct from natural termination, and prefixes never cross a reset. The 27-configuration continuous-control study has no termination within its first three transitions. The controlled sequential problem and implementation tests exercise early termination explicitly.

Observations are normalized with fitting-split means and standard deviations; rewards are not normalized. The dynamics model predicts normalized-state increments and raw rewards. Gymnasium-v5 health predicates applied to unnormalized observations determine Hopper and Walker2d termination. HalfCheetah has no natural termination.

Choosing the mixture and horizon before calibration. The model-value experiments set $C = 1.293995897451913$ using an outcome-independent count rule: it is the largest $C \leq 2$ for which a

Table 6: Training and calibration settings shared across the 27 model-value configurations.

Component	Settings
Dynamics ensemble	Seven Gaussian models; four width-200 SiLU layers; state-increment and reward outputs; Adam at 10^{-3}; batch size 256; at most 150 epochs; 2% fitting-data holdout; stop after ten epochs without a 1% holdout-MSE improvement; retain five best models. Layer weight decays: $(2.5, 5, 7.5, 7.5, 10) \times 10^{-5}$.
IQL	Twin critics, value network, and tanh-Gaussian actor; two width-256 ReLU layers; Adam at 3×10^{-4}; batch size 256; 10^5 updates; discount 0.99; expectile 0.7; advantage temperature 3; exponential-weight cap 100; target-update coefficient 0.005; cosine actor schedule.
Model-value policy	$M = 2$, $\epsilon = 0.293995897451913$, $H = 3$; uniform selection among tied maxima.
Calibration	500 episodes; $\alpha = 0.1$; exact order statistic; nonnegative clipping; tolerance $p = 0.9, \delta = 0.025$.
Testing	500 target-policy prefixes per configuration; independent test and oracle-calibration streams.

length-10 prefix retains at least 29 of 500 episodes with probability at least 0.95. The number 29 is sufficient for a finite 90%-content tolerance rank at confidence 0.95. At fixed $C \in (1, 2]$, $M = 2$ maximizes the mixture's expected score within this family: the expected improvement per additional candidate decreases with candidate count, while $\epsilon = (C - 1)/(M - 1)$. The reported experiment uses $H = 3$, the longest horizon in $\{1, 3, 5, 10\}$ satisfying its additional precision requirement $\Pr(N \geq 185) \geq 0.9999$. The actual probability is 0.999985. This precision check is separate from finite-sample validity; the rank rule remains defined at every retained count.

A.6 Repeated-selection critic protocol

The repeated-selection study uses HalfCheetah and Hopper SAC checkpoints at 100,000 training steps and Walker2d at 500,000 steps. Their files are named `medium_actor_clean`; the name does not identify the medium-quality checkpoint in Table 5. Each environment uses fitting seeds $0, 1, 2, 3, 4$, enumerated in Table 17. For seed z, split seed $2026 + z$ partitions 1,000 complete episodes into 500 fitting, 50 ratio-development, 300 calibration, 50 oracle, and 100 test-reset episodes. Fits use the shared architecture settings; source episodes overlap across fitting seeds.

The selector is $g(s, a) = \min\{Q_1(s, a), Q_2(s, a)\}$, where Q_1, Q_2 are the fixed IQL target critics. This critic study predicts the outcome with one independently randomized model rollout from the initial state; the 27-configuration study instead uses a deterministic predictor. For each horizon H, its prediction and physical outcome are

$$G_H(S_0; \xi) = \sum_{t=0}^{H-1} \gamma^t \widehat{R}_t(\xi) + \gamma^H V_{\mathrm{IQL}}(\widehat{S}_H(\xi)), \tag{16}$$

$$Y_H(\tau) = \sum_{t=0}^{H-1} \gamma^t R_t + \gamma^H B_H V_{\mathrm{IQL}}(S_H). \tag{17}$$

Here ξ contains the model rollout's sampled actions, Gaussian prediction noise, and randomly chosen elite model at each step. The model rollout uses the same frozen candidate policy from its first step; its first action is sampled independently of the physically observed action. It runs for H model steps. The physical outcome retains rewards only until natural termination, with B_H equal to one if the episode remains active at H and zero otherwise. Thus G_H is a state-based randomized predictor, not the deterministic $G_{\min}(S_0, A_0)$ used in the main 27-configuration study.

Prediction randomness follows the same fixed conditional law in calibration and testing and is independent of acceptance and physical continuation. Including ξ with the trajectory therefore leaves the transport and calibration arguments unchanged. Ten repetitions resample model predictions, completion, and target-policy actions on the same behavior and reset pools. The 100 test resets and 50 oracle resets are separate from calibration; the repetitions share these initial states.

For each configuration j, let $\bar{q}_{j,\mathrm{CC}}$ and $\bar{q}_{j,\mathrm{behavior}}$ be corrections averaged over its ten repetitions. We report both the pooled reduction and the mean relative reduction:

$$100\left(1-\frac{\sum_j \bar{q}_{j,\mathrm{CC}}}{\sum_j \bar{q}_{j,\mathrm{behavior}}}\right), \qquad \frac{100}{15}\sum_{j=1}^{15}\left(1-\frac{\bar{q}_{j,\mathrm{CC}}}{\bar{q}_{j,\mathrm{behavior}}}\right).$$

All behavior corrections in this comparison are positive. The second statistic is unchanged by multiplying a configuration's outcome scale by a positive constant. Paired bootstrap intervals use 10,000 resamples of three whole environment blocks, preserving all five fits and their ten-repetition means. These intervals describe variation among the three tested environments, treating each environment as one block. The available configuration means support correction and coverage comparisons. Individual residuals are not available for this study, so pinball loss is not reported for this study.

A.7 Comparison protocols and metrics

The selection sweep keeps the same 27 fitted scores and changes only the first-step candidate count. For each candidate count and fitted configuration, it collects 500 new target-test episodes and 500 separate oracle-calibration episodes. Its ordinary ranks are finite in every configuration. The repeated-selection protocol is specified above. Its 50-outcome oracle is an additional-information reference, not a matched-data baseline.

The matched model-value experiment applies all calibration rules to the same behavior pool and evaluates them on the same target-test set. Its size-matched behavior control averages 200 random subsets of size N. This is a descriptive sample-size control: N is derived from completion on the same pool. The oracle uses a separate set of 500 target outcomes. Weighted methods use Equation (11).

For a lower bound $L = G - q$ and outcome Y, pinball loss is $\rho_\alpha(Y-L)$, where $\rho_\alpha(u) = u(\alpha - \mathbf{1}\{u < 0\})$. Here u is outcome minus bound and $\mathbf{1}$ denotes an indicator. At $\alpha = 0.1$, an outcome below the bound incurs nine times the penalty of an equally large gap above it. This balances coverage with informative, high lower bounds. Correction sizes are normalized by the target-error standard deviation in the matched experiment and by the behavior-error standard deviation in the selection sweep. Weighted effective sample size is $(\sum_i \omega_i)^2 / \sum_i \omega_i^2$; it summarizes weight concentration, not additional independent episodes.

Confidence intervals use 10,000 hierarchical bootstrap draws: sample nine environment/quality cells with replacement, then three fitted seeds within each sampled cell, and average the selected values. Paired comparisons subtract methods before the same resampling. Seeds and every per-configuration result are recorded in the archive. The model-value analysis uses bootstrap seed 20260828; the other comparisons record their own seeds. Infinite ranks are reported explicitly. Finite-only summaries exclude them and state the resulting denominator.

Equal-budget score comparisons. For the sweep, the same stored pool of 256 fresh competitor scores per state supplies each $R \in \{M-1, 16, 64, 256\}$. Let b and u count competitor scores strictly below and equal to the logged action's score. The CDF baseline uses $\widehat{F}_R = (b + u/2)/R$ and weight $M\widehat{F}_R^{M-1}$. Shared-pool completion uses Equation (25). Both use the same behavior residuals, test trajectories, clipping, and maximum test weight M. The ordinary-CC curve uses a separate $M-1$-competitor stream; at the minimum budget the coupled weighted implementation equals it in distribution, apart from averaging rather than sampling tie-breaking. The shared-pool comparison reuses stored scores and evaluates every stated budget.

Both weighted methods use $500R$ new behavior draws and $500(R+1)$ candidate-score evaluations per configuration when the anchor is rescored. Table 15 separates these costs from extra physical outcomes. Generating and scoring the 256-competitor pools took a mean 1.07 seconds per configuration on the workstation described in Appendix D. Smaller budgets were obtained by taking prefixes of those pools; no separately measured per-budget network timing is claimed. The aggregation script runs on CPU and records its own timing separately.

The stored-data comparisons use bootstrap seed 2026092401 and include all 27 configurations and four candidate counts. Machine-readable outputs provide absolute metrics and paired intervals,

including normalized pinball loss. The reward-budget analysis measures data-size effects using the first episodes of each existing pool; the predictor and three-step policy stay fixed.

B Additional Results

B.1 Complete observed-reward application

Tables 7–9 report the full fixed grid. All methods calibrate the same fully observed reward target under pure selection at every step. At 2,000 logged episodes every ordinary and weighted bound is finite. Across the 27 setting/group averages, the mean absolute coverage difference from the target-data oracle is 2.12 percentage points for behavior calibration and 0.69 for ordinary CC. By setting, these differences are 1.80 versus 0.44 for best-of-two over three steps, 2.96 versus 0.67 for best-of-four over two steps, and 1.61 versus 0.95 for best-of-two over five steps. These summaries use all groups. They compare with an independently calibrated reference, not with an assumption that every realized bound must have exactly 90% coverage.

Table 7: **All reward-application groups.** Each row averages three fits and three independent source pools at $n = 2{,}000$. Coverage is in percent. The final two columns are ordinary CC's correction and pinball loss, each divided by the behavior-error standard deviation. All bounds are finite. CDF uses $M - 1$ competitors per active state; the oracle uses separate target outcomes.

Environment / quality	Behavior	CDF	**CC (Ours)**	Oracle	CC q/σ	CC loss$/\sigma$
Best of 2, 3 steps						
HalfCheetah low	91.77	90.66	90.66	89.58	1.208	0.1756
HalfCheetah medium	90.82	88.73	88.73	90.16	0.491	0.1740
HalfCheetah high	91.78	90.72	90.72	90.37	0.537	0.1763
Hopper low	94.27	91.24	91.24	91.09	0.220	0.1118
Hopper medium	93.96	94.70	94.70	94.61	0.037	0.1953
Hopper high	95.23	95.37	95.37	95.24	0.002	0.1432
Walker2d low	90.19	90.38	90.38	90.02	0.682	0.1317
Walker2d medium	87.22	89.79	89.79	90.17	0.327	0.1780
Walker2d high	84.71	89.66	89.66	89.67	1.705	0.1609
Best of 4, 2 steps						
HalfCheetah low	90.54	90.47	90.01	89.99	1.039	0.1739
HalfCheetah medium	90.22	90.09	89.70	90.13	0.593	0.1767
HalfCheetah high	92.59	90.67	90.95	89.69	0.714	0.1736
Hopper low	96.33	94.86	94.29	94.19	0.117	0.1188
Hopper medium	89.48	93.35	94.41	93.77	0.201	0.1997
Hopper high	89.04	91.32	91.68	93.12	0.151	0.1203
Walker2d low	91.32	89.48	89.53	90.42	0.520	0.1334
Walker2d medium	90.91	91.18	91.82	90.77	0.291	0.1716
Walker2d high	78.67	88.71	90.45	90.24	1.566	0.1746
Best of 2, 5 steps						
HalfCheetah low	92.99	90.49	90.49	89.76	1.149	0.1802
HalfCheetah medium	92.16	88.75	88.75	90.64	0.401	0.1702
HalfCheetah high	90.65	92.34	92.34	90.76	0.219	0.1649
Hopper low	95.63	94.82	94.82	95.36	0.076	0.1049
Hopper medium	95.28	95.44	95.44	95.28	0.007	0.1508
Hopper high	98.56	97.66	97.66	97.61	0.006	0.1741
Walker2d low	83.75	91.26	91.26	90.15	0.631	0.1533
Walker2d medium	89.36	93.12	93.12	91.33	0.556	0.1916
Walker2d high	89.81	90.56	90.56	89.83	1.761	0.1554

Prediction quality and sampling budget. At $(M, H) = (4, 2)$, ordinary CC's normalized pinball loss is 0.1603, compared with 0.1655 for behavior calibration and 0.1593 for the oracle. The paired CC-minus-behavior loss difference is -0.0052, with conditional pool-bootstrap range $[-0.0058, -0.0046]$. At $(2, 3)$, it is -0.0015, with range $[-0.0019, -0.0011]$. At $(2, 5)$, the difference is -0.0008, with range $[-0.0016, 0.0001]$. The resampling unit is a source pool with its three fits kept together, as specified in Appendix A.2.

Score-CDF weighting is a close comparator in this control application. At the minimum budget for $(4, 2)$, weighted CC has 0.30 percentage points higher mean coverage and 0.0006 higher normalized loss. At 64 competitors, normalized losses are 0.1595 for both methods to four decimals. For $M = 2$, the shared-pool estimators coincide. The controlled study below isolates the finite-budget reliability difference without the noise or predictor heterogeneity of the control tasks.

Table 8: **Complete reward-application comparisons.** Each setting averages 81 runs (nine groups, three fits, three pools), all with $n = 2{,}000$. Normalized metrics use each run's behavior-error standard deviation. Effective count is the retained count for ordinary CC and the weighted effective sample size otherwise. The learned-ratio model uses additional action-only training data; its fit diagnostics are reported in the text.

Method	Coverage (%)	q/σ	Loss$/\sigma \downarrow$	Effective count	Finite
Best of 2, 3 steps					
Behavior calibration	91.10	0.582	0.1623	2000.0	81/81
Ordinary CC (Ours)	91.25	0.579	0.1608	252.0	81/81
CDF, $R = 1$	91.25	0.579	0.1608	252.1	81/81
Weighted CC (Ours), $R = 1$	91.25	0.579	0.1608	252.1	81/81
CDF, $R = 16$	91.32	0.581	0.1604	768.8	81/81
Weighted CC (Ours), $R = 16$	91.32	0.581	0.1604	768.8	81/81
CDF, $R = 64$	91.34	0.581	0.1604	826.0	81/81
Weighted CC (Ours), $R = 64$	91.34	0.581	0.1604	826.0	81/81
Learned ratio (diagnostic)	91.23	0.590	0.1621	1981.2	81/81
Target oracle (extra data)	91.21	0.575	0.1602	2000.0	81/81
Best of 4, 2 steps					
Behavior calibration	89.90	0.533	0.1655	2000.0	81/81
Ordinary CC (Ours)	91.43	0.577	0.1603	126.1	81/81
CDF, $R = 3$	91.13	0.558	0.1597	334.3	81/81
Weighted CC (Ours), $R = 3$	91.43	0.577	0.1603	126.1	81/81
CDF, $R = 16$	91.40	0.564	0.1595	362.6	81/81
Weighted CC (Ours), $R = 16$	91.47	0.568	0.1596	317.1	81/81
CDF, $R = 64$	91.38	0.562	0.1595	378.4	81/81
Weighted CC (Ours), $R = 64$	91.42	0.564	0.1595	366.7	81/81
Learned ratio (diagnostic)	90.33	0.558	0.1656	1965.6	81/81
Target oracle (extra data)	91.37	0.556	0.1593	2000.0	81/81
Best of 2, 5 steps					
Behavior calibration	92.02	0.509	0.1614	2000.0	81/81
Ordinary CC (Ours)	92.72	0.534	0.1606	62.7	81/81
CDF, $R = 1$	92.72	0.534	0.1606	62.7	81/81
Weighted CC (Ours), $R = 1$	92.72	0.534	0.1606	62.7	81/81
CDF, $R = 16$	92.95	0.533	0.1592	412.6	81/81
Weighted CC (Ours), $R = 16$	92.95	0.533	0.1592	412.6	81/81
CDF, $R = 64$	92.98	0.534	0.1592	464.0	81/81
Weighted CC (Ours), $R = 64$	92.98	0.534	0.1592	464.0	81/81
Learned ratio (diagnostic)	93.05	0.568	0.1620	1943.9	81/81
Target oracle (extra data)	92.30	0.494	0.1586	2000.0	81/81

Table 9: **Observed reward calibration at both data budgets.** Each coverage and loss average uses finite bounds only; behavior and oracle bounds are finite in all runs. At $H = 5$, ordinary CC is finite in 79/81 runs for $n = 500$ and 81/81 for $n = 2{,}000$. These are measured results, not only binomial feasibility calculations.

M	H	n	Retained	CC finite	CC cov.	Beh. cov.	Oracle cov.	CC loss$/\sigma$
2	3	500	63.7	81/81	91.50	90.90	91.35	0.1627
2	3	2000	252.0	81/81	91.25	91.10	91.21	0.1608
4	2	500	31.5	81/81	92.05	89.74	91.43	0.1665
4	2	2000	126.1	81/81	91.43	89.90	91.37	0.1603
2	5	500	16.2	79/81	94.23	91.97	92.38	0.1722
2	5	2000	62.7	81/81	92.72	92.02	92.30	0.1606

Clipping and the learned-ratio diagnostic. Nonnegative clipping makes some bounds conservative even for the target-data oracle. For example, Hopper-high reaches 97.66% with ordinary CC and 97.61% with the oracle in the five-step setting. This is why the group-level oracle comparison complements distance from exactly 90%.

The classifier's final holdout cross-entropy ranges from 0.6640 to 0.69325, with mean 0.68725; 52 of 54 fits are below the balanced constant-classifier value $\log 2 \approx 0.69315$. Several fits therefore learn a class difference, while many remain close to the constant predictor. Table 8 reports this executed baseline, with its training losses and weights in the supplement. It is a diagnostic of the specified classifier, rather than a conclusion about all learned density-ratio methods.

All three independent pools. In the Walker2d-high, two-step best-of-four group, CC-minus-behavior coverage changes are 8.73, 13.38, and 13.22 percentage points for pools 0, 1, and 2, after averaging their three fixed fits. The corresponding ordinary-CC coverages are 88.43%, 91.07%, and 91.85%; the target-oracle coverages are 89.75%, 90.83%, and 90.13%. These values show the replication directly. The supplement provides the same pool-level records for every group and setting.

B.2 Complete finite-budget controlled comparison

Table 10 includes both error directions, both sample sizes, and tied and continuous scores at $R = M - 1$. Completion maintains coverage near the requested level in both directions, whereas the CDF plug-in can under-cover or over-cover depending on the error map. With continuous scores, $n = 2{,}000$, and $M = 16$, plug-in coverage is 85.53% for $E = a$, while completion gives 90.29%. For $E = -a$ in the same setting, the values are 97.31% and 90.31%. Thus increasing the number of logged outcomes does not by itself remove finite-competitor estimation error.

Table 10: **Both error directions and tied scores at minimum candidate budget.** Every row uses 1,000 independent calibration datasets and $R = M - 1$. The two error maps share each dataset. Coverage is evaluated exactly and averaged; all ranks are finite. Larger-budget results and all losses are included in the accompanying CSV files.

			$E = a$			$E = -a$		
n	Scores	M	CDF	CC (Ours)	Oracle	CDF	CC (Ours)	Oracle
500	Continuous	2	90.19	90.19	90.06	90.16	90.16	90.08
500	Continuous	4	87.29	90.37	90.02	96.18	90.41	90.03
500	Continuous	8	86.56	90.80	89.97	97.24	90.78	90.00
500	Continuous	16	86.95	91.67	90.08	97.75	91.84	90.07
500	Eight bins	2	90.20	90.20	89.98	90.11	90.11	90.05
500	Eight bins	4	88.10	90.50	90.03	95.71	90.45	90.07
500	Eight bins	8	88.18	90.88	90.06	96.82	90.92	90.05
500	Eight bins	16	90.25	92.12	89.99	97.14	92.27	90.06
2000	Continuous	2	90.03	90.03	89.99	90.06	90.06	89.99
2000	Continuous	4	86.93	90.05	90.01	96.02	90.09	90.01
2000	Continuous	8	85.79	90.09	89.99	96.93	90.20	90.00
2000	Continuous	16	85.53	90.29	90.00	97.31	90.31	90.00
2000	Eight bins	2	90.05	90.05	90.00	90.06	90.06	90.00
2000	Eight bins	4	87.61	90.06	90.02	95.51	90.08	90.02
2000	Eight bins	8	87.61	90.21	90.02	96.50	90.26	89.99
2000	Eight bins	16	88.71	90.49	90.00	96.40	90.56	89.99

Every reported method is finite across the complete grid. The larger budgets $R = 16, 64$, the exact-ratio reference, and direct target calibration appear in the supplied `summary.csv`; `paired.csv` gives coverage and loss differences with Monte Carlo uncertainty. These files retain all 320 setting/outcome/method rows, including cases where the empirical methods are close. A correct tie-aware plug-in is used in every tied-score case. The exact coverage and integrated pinball loss were independently recalculated for all 320,000 fitted cutoffs.

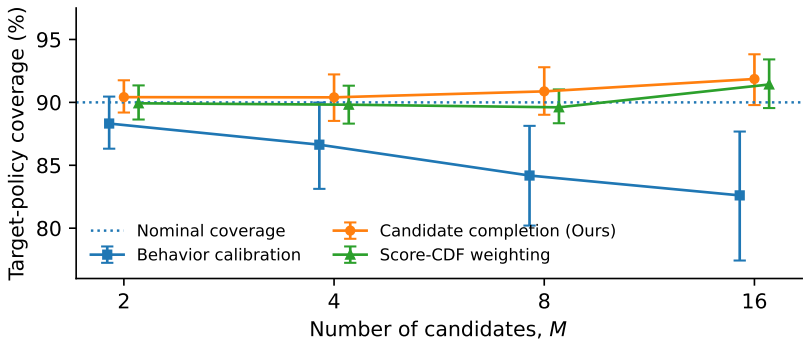


Figure 2: **First-action selection changes model-value calibration.** This sweep selects one action, followed by two behavior steps. Ordinary CC (Ours) and CDF weighting draw $M - 1$ competitors with separate random streams; behavior calibration uses none. Error bars are hierarchical 95% intervals over the 27 configurations.

B.3 Selection-strength and equal-budget comparisons

For the selection sweep, Table 11 gives correction sizes and the tolerance option. Corrections increase as the target policy favors higher-scoring actions. This adjustment improves pinball loss over behavior calibration by 0.017 at $M = 4$, with paired interval $[-0.035, -0.002]$, and by 0.030 at $M = 8$, with interval $[-0.052, -0.008]$. At $M = 16$, the loss reduction is 0.035; the CC-minus-behavior interval is $[-0.075, 0.000]$. Ordinary ranks remain finite in all configurations. The tolerance option at $M = 16$ has six finite bounds out of 27; their mean coverage is 95.2%. Non-finite bounds are reported separately rather than treated as informative predictions.

Table 11: Selection sweep details. Corrections are normalized by behavior-error standard deviation. Tolerance coverage is averaged only over configurations with finite tolerance ranks.

M	Behavior q	CC q	Oracle q	Tolerance finite	Finite tolerance coverage
2	1.31	1.41	1.39	27/27	93.9%
4	1.31	1.49	1.47	27/27	95.1%
8	1.31	1.62	1.54	27/27	96.2%
16	1.31	1.80	1.61	6/27	95.2%

Equal-budget score-CDF comparisons. Table 12 compares the specified CDF-power estimator with completion at minimum and 256-draw budgets. Table 13 reports the two intermediate budgets. Table 14 gives paired differences with uncertainty. The comparison shows similar empirical coverage and loss while distinguishing the estimators' guarantees. At $M = 2$, the mid-rank CDF and averaged-completion weights coincide exactly, including ties. For $M > 2$, taking a nonlinear power of the CDF estimate does not generally preserve the exact ratio mean. Unbiased score-based estimators, including shared-pool completion, satisfy the same exact weighted-calibration argument.

B.4 All repeated-selection configurations

Table 17 supplies all 15 fits and all four displayed settings. For the two-step pure-selection result, the pooled percentage reduction is 14.575%; the mean within-configuration reduction is 21.657%. Their descriptive environment-block intervals are [8.735, 43.236]% and [9.812, 42.841]%, respectively. All 15 mean corrections decrease. Table 16 reports all four settings with their uncertainty ranges.

For $(\epsilon, H) = (.514, 3), (.514, 5), (1, 2), (1, 3)$, mean retained counts are 85.71, 37.32, 74.13, and 36.91. The recorded ordinary-rank summaries are finite; the separately recorded tolerance finite counts are 150/150, 138/150, 150/150, and 135/150. The 50-target-outcome reference oracle has ordinary coverage 89.3%, 89.4%, 89.9%, and 89.9%, respectively. It uses new outcomes and fewer calibration observations than the 300-episode behavior pools.

Table 12: **Exact weighting at the same candidate budget.** Each pair uses the same 500 logged errors, R competitor scores per state, and test paths. Coverage (%) has a 90% target and 95% intervals. Here σ is the standard deviation of behavior errors; loss is pinball loss. Bold marks the smaller displayed loss in each pair, including ties. All bounds are finite in all 27 configurations.

		Score-CDF weighting			**Weighted CC (Ours)**		
M	R	Coverage [95% CI]	q/σ	Loss ↓	Coverage [95% CI]	q/σ	Loss ↓
2	1	89.9 [88.6, 91.3]	1.38	**0.441**	89.9 [88.6, 91.3]	1.38	**0.441**
2	256	90.4 [89.3, 91.6]	1.41	**0.439**	90.4 [89.3, 91.6]	1.41	**0.439**
4	3	89.8 [88.3, 91.3]	1.44	**0.439**	90.5 [89.0, 92.1]	1.48	0.440
4	256	90.7 [89.3, 92.2]	1.50	**0.440**	90.7 [89.3, 92.2]	1.50	**0.440**
8	7	89.6 [88.3, 91.0]	1.53	**0.430**	90.9 [89.5, 92.4]	1.62	0.439
8	256	91.0 [89.7, 92.3]	1.62	**0.432**	91.0 [89.7, 92.4]	1.62	**0.432**
16	15	91.4 [89.5, 93.4]	1.73	**0.433**	92.2 [90.3, 94.1]	1.81	0.445
16	256	91.5 [89.9, 93.3]	1.75	**0.438**	91.6 [90.1, 93.3]	1.76	**0.438**

Table 13: Intermediate competitor budgets for the same selection sweep. Each entry gives CC / CDF. All bounds are finite in all 27 configurations. The machine-readable table gives all four budgets and all per-configuration measurements.

M	R	Coverage (%)	Normalized q	Loss	Effective size
2	16	90.27 / 90.27	1.40 / 1.40	0.439 / 0.439	364.9 / 364.9
2	64	90.34 / 90.34	1.41 / 1.41	0.439 / 0.439	373.4 / 373.4
4	16	90.69 / 90.64	1.49 / 1.49	0.440 / 0.439	201.9 / 215.6
4	64	90.79 / 90.74	1.50 / 1.50	0.439 / 0.439	216.4 / 219.8
8	16	91.33 / 90.61	1.65 / 1.59	0.435 / 0.431	95.1 / 118.5
8	64	91.13 / 90.98	1.63 / 1.62	0.434 / 0.434	114.1 / 119.7
16	16	92.45 / 91.31	1.82 / 1.72	0.445 / 0.432	32.8 / 64.1
16	64	91.99 / 91.64	1.78 / 1.76	0.442 / 0.441	55.5 / 62.6

B.5 Complete matched comparisons

Table 18 reports the three-step model-value comparison with the size-matched control and three weighted-completion budgets. For $M = 2$ without score ties, the single-competitor selection indicator is also an unbiased estimate of the score CDF. Averaging K such indicators therefore gives the same weights as the corresponding fresh-CDF estimate. This equivalence explains the close performance of these implementations at the model-value operating point.

The ordinary correction's paired difference from the oracle is 0.04 normalized units, with interval $[-0.01, 0.09]$. The reduction relative to behavior calibration is about 0.06, with interval $[-0.01, 0.14]$. The intervals quantify uncertainty in the observed differences. Weighted completion increases effective sample size from 390 at one completion to 455 at 16 and 459 at 256. Corrections change

Table 14: Paired CC-minus-CDF differences with 95% hierarchical intervals. Coverage differences are percentage points. The same draw of cells and fitted seeds is used for both methods. Zero differences at $M = 2$ reflect identical estimators, not a failure to detect an effect.

M	R	Coverage difference	Normalized q difference	Loss difference
2	1	0.00 [0.00, 0.00]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]
2	256	0.00 [0.00, 0.00]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]
4	3	0.68 [0.02, 1.43]	0.041 [-0.003, 0.090]	0.001 [-0.001, 0.003]
4	256	0.00 [0.00, 0.00]	0.000 [0.000, 0.000]	0.000 [0.000, 0.000]
8	7	1.33 [0.33, 2.46]	0.090 [0.028, 0.160]	0.008 [0.000, 0.024]
8	256	0.03 [0.00, 0.12]	0.001 [0.000, 0.004]	0.000 [0.000, 0.000]
16	15	0.73 [0.25, 1.32]	0.078 [0.019, 0.186]	0.012 [-0.001, 0.042]
16	256	0.10 [0.00, 0.26]	0.008 [0.000, 0.023]	0.000 [-0.001, 0.000]

Table 15: Information and computation per selection-sweep configuration with $n = 500$. All methods calibrate the same outcome. "Exact" refers to a marginal finite-sample guarantee under the stated sampling conditions.

Method	Fresh draws	Extra outcomes	Guarantee
Behavior	0	0	For behavior, not generally the target
Ordinary CC (Ours)	$n(M-1)$	0	Exact; also tolerance ranks
Shared-pool CC (Ours)	nR	0	Exact weighted marginal
CDF-power weights	nR	0	Exact at $M = 2$; generally estimated at $M > 2$
Target oracle	–	500	Exact using new target data

Table 16: **Repeated selection yields tighter calibrated bounds.** Selection probability ϵ and horizon H specify each best-of-two policy. Each row averages ten runs within each of 15 configurations. Both reduction columns compare the correction with behavior calibration; mean relative reduction averages the within-configuration percentages. Descriptive 95% ranges resample three whole environments. Bold reductions have ranges above zero.

ϵ	H	Behavior coverage	**CC (Ours)** coverage	Pooled correction reduction	Mean relative reduction [95% range]
0.514	3	92.67%	90.55%	**5.8%**	**8.9 [2.1, 19.9]%**
0.514	5	93.05%	90.34%	2.0%	8.3 [-2.1, 23.3]%
1.000	2	94.80%	90.65%	**14.6%**	**21.7 [9.8, 42.8]%**
1.000	3	94.65%	91.15%	**8.2%**	**14.7 [2.6, 37.2]%**

little over those budgets. This comparison measures the benefit of reducing competition noise while keeping the physical data fixed.

Directly observed rewards. Table 19 uses Y_R and G_R from Appendix A. The policy, candidate comparisons, and retention decisions remain unchanged. Every three-step target was reconstructed from its recorded rewards, and the predictor was evaluated using the same fixed ensemble.

All 27 model-value configurations. Table 20 reports every model-value configuration. Zero corrections in two Hopper-high fits result from clipping a negative quantile at zero. The ordinary guarantee averages over calibration, so a single configuration's measured coverage can fall below 90%. The controlled tolerance experiment below evaluates reliability across independently sampled calibration datasets.

B.6 Experiments with known distributions

One-step construction. Actions are uniform on $[0, 1]$, the score is $g(a) = 2a$, and the predictor is $G(a) = 2a$. The two outcomes are $Y(a) = a$, giving error $E = a$, and $Y(a) = 3a$, giving error $E = -a$. If $F_A(x) = (1 - \epsilon)x + \epsilon x^M$, their target-error CDFs are $F_A(x)$ on $[0, 1]$ and $1 - F_A(-x)$ on $[-1, 0]$, respectively. Thus coverage is evaluated exactly for each correction without sampling test outcomes.

The grid uses $M \in \{2, 4, 8\}$, $\epsilon \in \{0.1, 0.293995897451913, 0.5, 1\}$, and $n \in \{100, 500, 2000\}$, with both outcomes and 5,000 independent calibrations per cell. Corrections are not clipped in these controlled experiments, allowing both signs of error. Table 21 reports representative settings; every grid result is provided in the archive. Monte Carlo standard errors in these displayed coverage estimates are at most 0.0006.

Sequential construction. The tabular problem has five active states, four actions, and an absorbing terminal state. Its reset probabilities are $(0.40, 0.35, 0.25, 0, 0)$. Each behavior-action probability is at least 0.05. Rewards are a fixed negative cost plus independent Gaussian noise with standard deviation 0.3; the discount is 0.99. The predictor is the exact three-step behavior value plus fixed Gaussian perturbations, and its rounded value supplies a score with ties. The complete matrices and random

Table 17: Every repeated-selection configuration. Entries are $\bar{q}_{\mathrm{CC}}/\bar{q}_{\mathrm{behavior}}$, each averaged over ten repetitions. The source checkpoints are 100k for HalfCheetah and Hopper, and 500k for Walker2d. Seeds identify distinct fits, not disjoint physical source datasets.

Environment	Fit seed	$\epsilon = .514, H = 3$	$\epsilon = .514, H = 5$	$\epsilon = 1, H = 2$	$\epsilon = 1, H = 3$
HalfCheetah	0	8.521 / 8.392	11.565 / 12.817	4.824 / 5.763	8.422 / 8.979
	1	6.946 / 7.765	11.870 / 12.145	4.909 / 6.010	7.932 / 8.707
	2	7.687 / 7.660	11.483 / 11.595	5.390 / 5.542	8.034 / 7.679
	3	7.701 / 8.653	13.181 / 12.841	6.124 / 7.126	8.990 / 8.969
	4	7.818 / 8.114	15.249 / 12.626	5.773 / 6.427	8.769 / 9.038
Hopper	0	1.681 / 2.016	1.545 / 2.220	0.844 / 1.578	1.454 / 2.273
	1	1.909 / 2.257	2.211 / 2.396	1.007 / 1.648	1.894 / 2.523
	2	1.747 / 2.071	1.675 / 2.381	1.020 / 1.706	1.635 / 2.308
	3	1.446 / 2.026	1.680 / 2.235	0.874 / 1.883	1.076 / 2.309
	4	1.223 / 1.594	1.462 / 1.923	0.973 / 1.497	1.054 / 1.835
Walker2d	0	5.070 / 5.663	3.999 / 4.544	5.788 / 6.663	4.924 / 5.962
	1	6.151 / 5.701	3.815 / 4.610	7.696 / 7.760	5.676 / 5.601
	2	6.191 / 6.335	7.689 / 7.752	6.526 / 7.265	6.226 / 6.630
	3	4.068 / 4.209	2.945 / 2.989	4.297 / 4.546	4.281 / 4.340
	4	3.786 / 3.880	5.938 / 5.216	2.745 / 3.409	4.146 / 4.054

Table 18: All matched model-value comparisons. Coverage and its hierarchical 95% interval are percentages. The normalized correction divides by each configuration's target-error standard deviation. All methods have finite bounds in all 27 configurations.

Method	Coverage [95% CI]	Normalized q	Pinball	Effective count
Uncalibrated	48.6 [31.5, 65.6]	0.00	0.986	–
Behavior calibration	91.6 [90.7, 92.5]	1.34	0.441	500
Size-matched behavior	91.8 [91.0, 92.8]	1.36	–	231
Ordinary CC (Ours)	90.8 [89.7, 92.0]	1.28	0.443	231
Tolerance CC (Ours)	94.3 [93.5, 95.1]	1.58	0.464	231
Weighted CC (Ours), $K = 1$	90.8 [89.8, 91.9]	1.28	0.441	390
Weighted CC (Ours), $K = 16$	90.8 [89.8, 91.9]	1.27	0.441	455
Weighted CC (Ours), $K = 256$	90.9 [89.9, 92.0]	1.27	0.441	459
Target oracle	90.3 [89.1, 91.5]	1.24	0.441	500

Table 19: Reward-only outcome on the same 27 configurations. Coverage intervals are hierarchical 95% intervals. All ranks are finite.

Method	Coverage (%) [95% CI]	Normalized q	Pinball
Uncalibrated	73.6 [61.3, 85.2]	0.00	0.063
Behavior calibration	91.3 [89.7, 93.2]	0.61	0.039
Size-matched behavior	91.6 [90.0, 93.5]	0.62	–
Ordinary CC (Ours)	91.3 [89.8, 93.2]	0.61	0.039
Tolerance CC (Ours)	94.1 [93.0, 95.3]	0.80	0.040
Target oracle	91.3 [89.7, 93.1]	0.60	0.039

Table 20: All 27 primary configurations. N: retained calibration paths; q_o, c_o: ordinary correction and coverage; q_t, c_t: tolerance correction and coverage; q_{nv}, c_{nv}: behavior-calibrated correction and coverage (all 500 behavior residuals); c_0: uncalibrated coverage ($q = 0$). Each coverage uses the same 500 fresh target paths.

Environment	Quality	Seed	N	q_o	c_o	q_t	c_t	q_{nv}	c_{nv}	c_0
HalfCheetah	low	0	214	4.7856	.932	5.8163	.956	4.5741	.928	.618
	low	1	228	5.5738	.934	7.0175	.968	4.6480	.902	.564
	low	2	240	3.8192	.880	5.0758	.940	4.1130	.902	.624
	mid	0	240	3.5833	.906	4.8887	.942	3.7735	.914	.542
	mid	1	236	4.2911	.878	5.2219	.918	4.7511	.892	.502
	mid	2	244	4.2583	.914	5.2754	.940	3.7602	.902	.602
	high	0	218	4.4166	.922	6.2108	.958	4.1987	.918	.730
	high	1	225	4.7502	.934	5.8966	.952	4.2049	.916	.682
	high	2	218	5.2161	.908	6.4820	.932	4.9246	.904	.670
Hopper	low	0	260	1.2848	.868	1.5538	.902	1.4380	.888	.212
	low	1	238	1.3250	.894	1.5462	.928	1.5275	.926	.178
	low	2	221	0.9655	.892	1.2644	.946	1.2311	.942	.354
	mid	0	225	0.5544	.880	0.6613	.936	0.6008	.906	.230
	mid	1	221	1.1778	.900	1.2897	.938	1.1778	.900	.298
	mid	2	224	1.6917	.904	1.9331	.954	1.8289	.938	.064
	high	0	220	0.0000	.978	0.0000	.978	0.0000	.978	.978
	high	1	233	0.0000	.942	0.0000	.942	0.0000	.942	.942
	high	2	223	0.0357	.864	0.1084	.926	0.0445	.866	.824
Walker2d	low	0	207	5.6797	.900	6.8630	.956	6.1518	.920	.158
	low	1	228	5.0497	.932	5.6438	.950	5.1188	.936	.164
	low	2	218	5.2934	.894	6.2590	.948	5.8069	.920	.192
	mid	0	246	3.7003	.926	4.3000	.954	3.7963	.930	.230
	mid	1	243	4.3994	.936	5.3597	.962	4.2655	.926	.224
	mid	2	244	3.6186	.892	4.8282	.938	3.8625	.908	.250
	high	0	239	1.1667	.892	1.6422	.924	1.3057	.908	.716
	high	1	229	0.8784	.900	1.4649	.936	0.9052	.902	.788
	high	2	242	1.1531	.908	1.6954	.946	1.1474	.906	.798

Table 21: Exact-distribution evaluation at $n = 500$. Entries are mean coverage percentages over 5,000 calibrations. "Exact weights" uses the analytically known ratio; the target oracle collects new target samples.

Error	M	ϵ	Behavior	CC	CC tolerance	Exact weights	Oracle
$E = a$	2	0.294	87.4	90.1	92.9	90.1	90.1
$E = a$	2	1	81.0	90.2	93.6	90.2	90.0
$E = a$	4	1	65.7	90.3	95.0	90.4	90.0
$E = -a$	2	1	99.0	90.2	93.6	90.3	90.0
$E = -a$	4	1	100.0	90.3	94.9	90.6	90.0

seed 20260923 are in the code. This diagnostic problem was constructed using known population quantities to make selection and termination effects observable.

The target distribution is enumerated over 48,420 trajectory skeletons at horizon three and 968,420 at horizon four. Gaussian reward noise is integrated analytically. Table 22 tests retained path distributions. Empirical total variation is compared with direct target samples of the same size, because a finite sample differs from a large discrete distribution even under correct sampling. Table 23 uses 2,000 independently generated calibration datasets, each with 500 behavior episodes.

Table 22: Path-law checks, 200,000 behavior proposals per row. Direct-sample TV is the expected empirical discrepancy of an equally sized sample drawn from the exact target law. TV denotes total-variation distance.

M	ϵ	H	Retention	Predicted retention	CC empirical TV	Direct-sample TV
2	.294	3	.4619	.4615	.1424	.1432
3	.600	3	.0937	.0939	.2800	.2808
4	1.000	3	.0153	.0156	.4199	.4201
3	.600	4	.0429	.0427	.7208	.7200

Table 23: Independent calibration repetitions in the three-step tabular problem. Mean coverage and the fraction of fitted bounds whose exact coverage is below 90% are percentages. Only tolerance calibration imposes the 2.5% failure requirement in the last columns.

	$M = 3, \epsilon = 0.6$		$M = 2, \epsilon = 1$	
Method	Mean coverage	Below 90%	Mean coverage	Below 90%
Behavior calibration	88.1	89.1	87.6	94.8
CC without padded tests	87.5	65.2	87.9	66.7
Ordinary CC (Ours)	91.0	35.6	90.6	37.2
Tolerance CC (Ours)	97.7	1.25	96.8	1.55
Weighted CC (Ours), $K = 1$	91.5	28.8	90.7	39.7
Weighted CC (Ours), $K = 16$	91.8	14.6	91.3	23.9
Exact-ratio weighting	91.8	13.9	91.3	23.1
Target oracle	90.1	48.8	90.0	47.8

The tolerance failure fractions have exact 95% binomial intervals of [0.8,1.8]% and [1.1,2.2]%. These experiments distinguish coverage averaged over calibrations from the tolerance guarantee on individual fitted bounds. The 2.5% failure target applies only to tolerance calibration.

B.7 Approximate behavior samplers

Controlled sampler changes. In the tabular problem, we perturb the behavior sampler by known statewise TV distances in $\{0.01, 0.02, 0.05, 0.10, 0.20\}$. Both deployment cases in Appendix C.4 are checked across the four trajectory settings. All 40 computed discrepancies satisfy the stated bounds. The minimum resulting population coverage is 87.9%. These tests quantify the effect of known sampler differences independently of learning a behavior model.

Learned behavior sampler. For each of the nine environment/quality cells at fitting seed zero, a tanh-Gaussian behavior clone is fitted by maximum likelihood to 500 fitting episodes. Training uses a 5% holdout, at most 400 epochs, and patience ten. Its estimated TV distance, averaged over visited states, is 0.42; this average is not a uniform bound on η. Table 24 distinguishes using this clone for completion from deploying it. With the original deployed policy unchanged, clone-based completion achieves 91.1% mean coverage. Changing the deployed sampler as well gives 88.0%. This distinction is exactly the one addressed by the two robustness bounds.

The first-step logged-selection rate is 0.516 with the clone versus 0.510 with exact completion. The corresponding two-sided binomial test probabilities are 0.04 and 0.19, testing the expected rate 0.5. This test can reveal inconsistency; passing it does not identify the entire conditional sampling distribution.

Table 24: Behavior-clone experiment. Coverage is in percent. "Original target" keeps deployment unchanged; "clone target" also uses the clone at deployment. HC, Hop, and W2d denote the three environments; L, M, H denote behavior quality.

		Original target		Clone target	
Cell	Mean TV	Original CC	Clone CC	Clone CC	Behavior calibration
HC–L	.63	93.2	93.2	90.4	89.0
HC–M	.69	90.6	91.4	82.2	81.6
HC–H	.70	92.2	92.8	85.2	83.4
Hop–L	.22	86.8	84.4	90.4	93.0
Hop–M	.28	88.0	87.6	80.4	84.0
Hop–H	.17	97.8	97.8	99.2	99.2
W2d–L	.35	90.0	90.0	90.2	92.8
W2d–M	.36	92.6	90.4	81.8	84.8
W2d–H	.37	89.2	92.4	92.0	90.0

B.8 Data requirements for model-value and reward bounds

Model-value outcomes. For the primary $C = 1.293995897451913$, horizons $H = 3, 5, 10, 20$ retain 230.8, 137.8, 38.0, and 2.9 trajectories in expectation from 500 episodes. These are predictions from nC^{-H}, not fitted performance measurements. More data or weaker repeated selection can be chosen before observing calibration errors. Table 25 recalibrates using the first episodes of each model-value calibration pool. Ordinary 90% ranks need at least nine retained errors; the specified tolerance rank needs at least 36.

Table 25: Calibration-pool size on the 27 primary configurations. Coverage averages only finite bounds; the finite count is stated for each method.

Behavior episodes	Mean retained	Ordinary finite	Coverage (%)	Tolerance finite	Coverage (%)
50	22.63	27/27	92.62	0/27	–
100	45.44	27/27	90.80	26/27	97.55
200	91.81	27/27	90.58	27/27	95.90
300	136.48	27/27	90.70	27/27	95.26
400	183.07	27/27	90.62	27/27	94.47
500	230.52	27/27	90.78	27/27	94.33

Directly observed rewards. Table 26 uses directly measured rewards and the fixed predictor G_R from Equation (15). Selection is repeated at all three steps with $M = 2$ and $\epsilon = 0.294$. The normalized correction uses the corresponding target-error standard deviation. At 500 episodes, CC's pinball loss is 0.03881, compared with 0.03889 for behavior calibration and 0.03875 for the target oracle. The three methods are close at this operating point. The complete record reports every data size, its finite-bound count, and hierarchical intervals.

Table 26: Observed three-step rewards under repeated selection ($M = 2$, $\epsilon = 0.294$). A row uses the first n episodes of each stored pool, without changing the predictor or policy. Every coverage and loss entry is CC / behavior calibration. Each method averages its own finite configurations; behavior ranks are finite in all 27. The CC finite count is explicit. The appendix data provide corresponding target-oracle results and uncertainty.

n	Mean retained	CC finite	Coverage (%)	Normalized q	Pinball loss
25	11.6	22/27	92.8 / 93.6	0.87 / 0.84	0.041 / 0.042
50	22.6	27/27	93.2 / 91.7	0.81 / 0.66	0.042 / 0.040
100	45.4	27/27	91.6 / 91.9	0.66 / 0.65	0.040 / 0.039
200	91.8	27/27	91.5 / 91.5	0.63 / 0.63	0.039 / 0.039
300	136.5	27/27	91.5 / 91.5	0.63 / 0.63	0.039 / 0.039
500	230.5	27/27	91.3 / 91.3	0.61 / 0.61	0.039 / 0.039

Table 27 varies horizon and data budget without using outcomes. Its ordinary feasibility probability is $\Pr(N \geq 9)$ for $N \sim \mathrm{Binomial}(n, C^{-H})$. Weighted completion has a different observable feasibility test: rearranging Equation (11) gives $\sum_i \omega_i \geq 9C^H$ at 90% coverage. This condition quantifies the total weight needed for a finite bound. The calculated grid also includes best-of-four and mixture probability 0.5. It forecasts finite-bound availability; Tables 25 and 26 measure prediction quality.

Table 27: Calculated data requirements for repeated selection at every step. Entries are expected retained count / probability (percent) of obtaining a finite ordinary 90% rank. These are binomial calculations, not observed prediction quality. Selection is best-of-two; the mixture row uses $\epsilon = 0.294$, all other rows use pure selection.

Policy	H	$n = 100$	$n = 300$	$n = 500$	$n = 2000$
Mixture	3	46.2 / 100.0	138.5 / 100.0	230.8 / 100.0	923.1 / 100.0
Pure	2	25.0 / 100.0	75.0 / 100.0	125.0 / 100.0	500.0 / 100.0
Pure	3	12.5 / 89.1	37.5 / 100.0	62.5 / 100.0	250.0 / 100.0
Pure	5	3.1 / 0.4	9.4 / 59.5	15.6 / 97.5	62.5 / 100.0
Pure	10	0.1 / 0.0	0.3 / 0.0	0.5 / 0.0	2.0 / 0.0

C Essential Proofs

The proofs use the data interface in Section 3: independent reset episodes; a fixed score, predictor, target, and selection rule; the same behavior distribution for logged and auxiliary actions; independent competition and acceptance randomness; and a shared environment and reset law. All statements may be read conditional on the fitted models. No independence between transitions within an episode is assumed.

C.1 Why completion produces the selected-action distribution

Idea. Every candidate label is equally likely to occupy any role in the selection procedure. The logged label's contribution is therefore one Mth of the selected action's total contribution.

Fix a state s. Let $f(a)$ be any bounded function of an action, let $\widetilde{A}_0, \ldots, \widetilde{A}_{M-1}$ be independent behavior draws, and let J be the selected label. Uniform tie-breaking gives

$$\mathbb{E}[f(\widetilde{A}_J) \mid s] = \sum_{j=0}^{M-1} \mathbb{E}[f(\widetilde{A}_j)\mathbf{1}\{J = j\} \mid s] \tag{18}$$

$$= M\mathbb{E}[f(\widetilde{A}_0)\mathbf{1}\{J = 0\} \mid s]. \tag{19}$$

For the weight W in Equation (3), an independent uniform acceptance test therefore satisfies

$$\mathbb{E}_\mu[f(\widetilde{A}_0)\mathbf{1}\{\text{passes}\} \mid s] = C^{-1}\mathbb{E}_\mu[f(\widetilde{A}_0)W \mid s] \tag{20}$$

$$= C^{-1}\big((1-\epsilon)\mathbb{E}_\mu[f(\widetilde{A}_0) \mid s] + \epsilon\mathbb{E}[f(\widetilde{A}_J) \mid s]\big) \tag{21}$$

$$= C^{-1}\mathbb{E}_\pi[f(A) \mid s]. \tag{22}$$

Choosing f to indicate membership in an action set proves Equation (5). Choosing $f = 1$ gives acceptance probability $1/C$, averaged over behavior actions. Conditioning on the logged action gives Equation (4). The argument extends to any fixed randomized selector whose choice respects a relabeling of the candidates.

From one step to a full trajectory. At each active state, the preceding identity replaces the accepted behavior-action distribution by π/C. Conditional on that action, the next state, reward, and termination follow the shared environment rule. Iteration over the $\min(T, H)$ active steps gives the target path distribution multiplied by $C^{-\min(T,H)}$. The remaining padded tests contribute $C^{-(H-T)}$ when $T < H$. The total is C^{-H} for every path length, proving Equation (6).

Independent behavior episodes and independent acceptance randomness give independent retention indicators, each with probability C^{-H}. Hence $N \sim \mathrm{Binomial}(n, C^{-H})$. Conditional on any

particular retained subset, the joint accepted-path distribution factors into copies of P_π^H. Conditioning only on its size $N = m$ leaves the same product distribution. This proves Theorem 1. Notice that an individual observed path has acceptance probability equal to its trajectory probability ratio divided by C^H; the constant rate is the average over proposals.

C.2 Why the ordinary and tolerance ranks cover

Ordinary rank: idea. After completion, calibration errors and the future error are interchangeable in distribution. A rank chosen to leave at most an α fraction above it therefore gives the desired coverage.

Condition on $N = m$. Theorem 1 makes the m retained errors and the independent future error identically distributed and independent. With random tie-breaking, the future error's rank among the $m + 1$ errors is uniform. The event that its rank is at most k_m implies that it is no larger than the k_mth retained error. Thus, whenever $k_m \le m$,

$$\Pr(E_H(\tau^*) \le E_{(k_m)} \mid N = m) \ge \frac{k_m}{m+1} \ge 1 - \alpha.$$

Using a non-strict comparison makes ties conservative. If $k_m > m$, the infinite correction covers automatically. Averaging over N proves Theorem 2. Clipping a negative correction to zero can only lower the prediction bound, preserving the inequality.

Tolerance rank: idea. A fitted error cutoff fails to cover fraction p only if unusually many calibration errors fall below the target distribution's p-quantile. A binomial tail controls that event.

Let $F(x) = \Pr_\pi(E_H(\tau) \le x)$ be the target-error CDF, and let $\Pr_{\text{cal}}$ denote probability over the calibration data. For a continuous CDF, the transformed errors $F(E_i)$ are independent uniforms. Therefore

$$\Pr_{\text{cal}}(F(E_{(k)}) < p \mid N = m) = \Pr(\text{Binomial}(m, p) \ge k).$$

When the error distribution has point masses (so ties can occur), $F(E_i)$ is stochastically no smaller than a uniform variable, so the same right-hand side is an upper bound. Selecting k by Equation (9) gives failure probability at most δ for every m. An infinite correction has zero failure probability. Consequently,

$$\Pr_{\text{cal}}\Big(\Pr_\pi(E_H(\tau^*) \le q_{\text{tol}} \mid \text{calibration}) \ge p\Big) \ge 1 - \delta. \tag{23}$$

The restriction $k \ge k_m$ also preserves the ordinary marginal guarantee. This is the classical order-statistic tolerance argument (Wilks, 1941), applied to the target samples supplied by completion.

C.3 Why weighted completion is valid

Idea. Include the random candidate comparisons as part of each observation. On this enlarged observation space, the random completion weight is an exact probability ratio. Ordinary weighted conformal reasoning then applies, and the deterministic bound C^H safely replaces the unknown weight of the future observation.

Let $z = (\tau, \zeta)$ consist of a behavior trajectory and all its auxiliary competition randomness ζ. Write P_0 for its joint distribution and $\omega(z)$ for Equation (10). Competitions are independent across steps and repetitions conditional on the trajectory. Equation (4) gives

$$\mathbb{E}[\omega(z) \mid \tau] = \prod_{t < \min(T,H)} \ell_{S_t}(A_t) =: \Lambda(\tau) = \frac{dP_\pi^H}{dP_\mu^H}(\tau). \tag{24}$$

Here Λ is the trajectory probability ratio, and P_μ^H is the padded behavior-trajectory distribution. Since the weight has expectation one under behavior, $P_1(dz) = \omega(z)P_0(dz)$ is a probability distribution. Its trajectory marginal is exactly P_π^H.

Consider n independent calibration observations from P_0 and one independent future observation from P_1. Conditional on their unordered collection, the probability that observation j is the future one is proportional to its weight ω_j. The weighted $(1 - \alpha)$-quantile of the full collection leaves at

most an α fraction of total weight strictly above it. Replacing the future observation's score by $+\infty$ can only increase that quantile. This is the weighted conformal rank argument (Tibshirani et al., 2019).

The actual future weight is at most C^H. Replacing that weight at $+\infty$ by C^H again only increases the cutoff, giving exactly Equation (11). Its covered probability is therefore at least $1-\alpha$. This proves the weighted guarantee on the target trajectory marginal without requiring an auxiliary test-time competition drawn from a different distribution.

Averaging competitions from one shared pool. Take $R \geq M-1$ independent behavior competitors at the logged state. Let b count scores below the logged action and u count ties with it. Among all size-$M-1$ subsets, the average probability of selecting the logged action is

$$\phi_R = \frac{\binom{b+u+1}{M} - \binom{b}{M}}{(u+1)\binom{R}{M-1}}, \qquad \overline{W}_R = (1-\epsilon) + \epsilon M \phi_R. \tag{25}$$

A binomial coefficient is zero when its lower argument exceeds the upper one. To see the formula, a subset with j tied competitors and $M-1-j$ lower competitors selects the logged action with probability $1/(j+1)$. There are $\binom{u}{j}\binom{b}{M-1-j}$ such subsets; summing their contributions gives Equation (25). A subset containing a higher score contributes zero.

Each fixed subset consists of $M-1$ independent behavior draws. Its completion weight therefore has conditional mean $\ell_s(a)$ by Equation (4). Averaging over subsets preserves this mean and the bound $0 \leq \overline{W}_R \leq C$, even though subsets overlap. With fresh independent pools across trajectory steps, their product satisfies Equation (24), so the preceding weighted proof applies unchanged. This averages valid competitions and the uniform tie draw, preserving their exact mean. For $M=2$, $\phi_R = (b + u/2)/R$, exactly the empirical mid-rank CDF.

A useful variance bound. For one completion per active step, label symmetry gives $\mathbb{E}[W_t^2 \mid S_t] = 1 + \epsilon^2(M-1)$ before the logged action is sampled. Repeated conditioning over at most H active steps therefore gives

$$\mathbb{E}_\mu[\omega_i^2] \leq \big(1 + \epsilon^2(M-1)\big)^H. \tag{26}$$

Equality holds when every trajectory has H active steps. The inequality also covers action-dependent early termination.

C.4 How sampler error changes coverage

Statement. Let $\widetilde{\mu}$ be the approximate behavior sampler, and suppose $\sup_s \mathrm{TV}(\mu(\cdot \mid s), \widetilde{\mu}(\cdot \mid s)) \leq \eta$. Total variation is $\sup_B |P(B) - Q(B)|$; for two discrete distributions it is half the sum of absolute probability differences. Define $d = \epsilon M \eta$ and the geometric sum $\mathcal{S}_H = \sum_{t=0}^{H-1}(1+d)^t$. If completion uses $\widetilde{\mu}$ but deployment retains the original target policy, its retained trajectory law differs from the target by at most

$$\Delta_A = \min\left\{1,\ \frac{M}{2}\left[(1+d)^H - 1\right]\right\}. \tag{27}$$

If deployment also uses $\widetilde{\mu}$ for both the behavior branch and selected candidates, the bound is

$$\Delta_B = \min\left\{1,\ (C + \epsilon M/2)\eta \mathcal{S}_H\right\}. \tag{28}$$

Provided retention has positive probability, ordinary coverage is at least $1-\alpha-\Delta_A$ or $1-\alpha-\Delta_B$, respectively. For a tolerance bound, replace the guaranteed content p by $p-\Delta_A$ or $p-\Delta_B$, with the same confidence $1-\delta$.

Idea. Changing one action sampler changes one selection test by a controlled amount. Add those changes across the trajectory, then account for normalizing the accepted observations. This avoids repeatedly applying the potentially much larger factor C.

Step 1: describe the changed acceptance rule. Let $\widetilde{p}_s(a)$ be the probability that logged action a is selected against $M-1$ competitors from $\widetilde{\mu}$. Define

$$b_s(a) = (1-\epsilon) + \epsilon M \widetilde{p}_s(a), \qquad X(\tau) = \prod_{t<\min(T,H)} b_{S_t}(A_t), \qquad Z = \mathbb{E}_\mu X.$$

The unnormalized accepted-path distribution is $U(d\tau) = X(\tau)P_\mu^H(d\tau)$; its mass is Z. The actual retained law is U/Z, and retention probability is $C^{-H}Z$.

Step 2: accumulate the sampling error. Write $\|P-Q\|_1$ for the absolute mass of the signed measure $P-Q$, which is twice TV when both are probabilities. At one active state, coupling the $M-1$ competitor draws gives

$$\|\mu_s b_s - \pi_s\|_1 \le \epsilon M(M-1)\eta.$$

Here $\mu_s b_s$ means multiplying the behavior-action distribution by the function b_s. Its total mass differs from one by at most d: selection symmetry under $\widetilde{\mu}$ gives $\int b_s \, d\widetilde{\mu}_s = 1$, and the constant $1-\epsilon$ cancels when comparing integrals under the two samplers. Thus

$$((1-d)^+)^H \le Z \le (1+d)^H, \qquad |Z-1| \le d\mathcal{S}_H, \tag{29}$$

where $x^+ = \max\{0, x\}$. Terminal padded steps have mass one and obey the same bounds.

Replace one step at a time by the desired target step. A prefix of t approximate steps has mass at most $(1+d)^t$, while the remaining target steps preserve mass. Summing gives $\|U - P_\pi^H\|_1 \le \epsilon M(M-1)\eta\mathcal{S}_H$. Also,

$$\mathrm{TV}(U/Z, P_\pi^H) \le \tfrac{1}{2}\|U - P_\pi^H\|_1 + \tfrac{1}{2}|Z-1|.$$

Substitution yields Δ_A, using $d\mathcal{S}_H = (1+d)^H - 1$.

Step 3: translate distribution error into coverage. For clone-based deployment, its one-step target law is $\widetilde{\mu}_s b_s$. Since $0 \le b_s \le C$, the corresponding one-step difference is at most $2C\eta$ in absolute mass. The same replacement and normalization steps give Δ_B. For any fixed error cutoff, probabilities under two trajectory laws differ by at most their TV distance. Applying this fact to the ordinary coverage event and to each fitted tolerance cutoff proves the stated coverage bounds.

C.5 The precise efficiency statement

Idea. An exact rejection procedure cannot retain a trajectory more often than once. The target trajectory with the largest behavior-to-target change therefore limits the overall acceptance rate.

Consider any procedure that either retains an unchanged logged trajectory or rejects it, and whose retained law is exactly P_π^H. Let p_{acc} denote its overall acceptance probability, and let Λ be the trajectory ratio in Equation (24). Its conditional acceptance probability must equal $p_{\mathrm{acc}}\Lambda(\tau)$. Since that probability is at most one,

$$p_{\mathrm{acc}} \le \frac{1}{\operatorname{ess\,sup}_\tau \Lambda(\tau)}. \tag{30}$$

The essential supremum is the largest attainable ratio after excluding probability-zero exceptions. Candidate completion attains this limit when that supremum equals C^H. One sufficient setting has full-length episodes and continuous score distributions that allow ratios arbitrarily close to C at every step. The equality condition must be checked when actions affect termination. This optimality statement is specific to exact rejection of unchanged whole trajectories.

D Reproduction and Released Materials

Released experiments. The supplement contains the implementation, data-collection, fitting, and evaluation runners, tests, protocol files, and numerical records. The `paper/` directory contains the editable manuscript and official style files. The model-value, selection-sweep, and tabular records are under `results/` and `extended/`. The `focused/` directory contains the code for the reward application and the controlled comparison, the complete controlled comparison, and the application export in `focused/returned/application/`. The export includes every prediction, observed reward, error, weight, retention mask, seed, and timing needed to recalculate the 243 evaluations, so these results can be inspected and recomputed without a simulator.

Verification and rebuilding. From the archive root, the following commands check file integrity, audit all stored application observations, recalculate the controlled distributions, regenerate the application and controlled-comparison tables and figure, and build the paper:

```
python verify_archive.py
python scripts/audit_application_export.py \
  --root focused/returned/application \
  --output final_analysis/application_audit.json \
  --addon focused
python scripts/audit_controlled_results.py
python scripts/analyze_application_results.py
python scripts/render_final_tables.py
python scripts/render_final_figures.py
cd paper && bash build.sh
```

The audits independently recover discounted outcomes, residuals, retention decisions, exact calibration ranks, and prediction losses. They also check competitor-count weight formulas, source-pool sharing, separate reset streams, and executed configuration hashes. Numerical comparisons allow only floating-point summation roundoff; discrete ranks and finite-bound counts must match exactly. The application export contains 1,458,000 prediction errors, including reuse of source episodes across fixed fits and settings. The controlled audit checks exact coverage and integrated loss at all 320,000 stored cutoffs. The README lists further tests.

Regenerating the experiments. The controlled experiment runs on CPU:

```
python focused/run_experiments.py controlled --repetitions 1000
```

For the application, run the supplied configuration against the project directory containing the trained policies and models, choosing a new output directory:

```
python focused/run_experiments.py application \
  --project-root /path/to/project \
  --output /path/to/new/application_run
```

This restores the preserved SAC policies, dynamics ensembles, IQL models, and normalization statistics rather than retraining them. Source, target-test, and oracle streams are collected independently. The classifier baseline is fitted using action-only data from the fitting split. Stage-level checkpoints permit resuming interrupted runs. The trained models and trajectory data occupy approximately 8.3 GB and are not bundled; the collection and fitting scripts regenerate them. The calibration runners check source-file paths and hashes, so a relocated run needs regenerated manifests.

Compute and release checks. The workstation has an Intel Core i7-13700K, approximately 45 GiB usable RAM, and an NVIDIA RTX 4070 SUPER with 12 GB VRAM. The application used Python 3.11.15, PyTorch 2.5.1 with CUDA 12.1, Gymnasium 1.3.0, MuJoCo 3.12.0, NumPy 2.4.6, and SciPy 1.17.1. All 243 jobs completed; the recorded sum of job times is 26.2 minutes and the workstation log reports approximately 29 minutes end to end, including setup and source collection. This run evaluates the trained models and fits only the small ratio classifiers; it does not retrain policies. Fitting the policies and models consumed approximately three GPU-hours, with 20–30 GPU-hours estimated for the whole project.

The archive includes audit reports and executable tests. Machine-specific paths in run metadata are anonymized, and numerical arrays are unchanged. `ARCHIVE_MANIFEST.sha256` lists a checksum for every distributed file. The research code is released under the supplied MIT license. Third-party dependencies retain their own licenses, and the simulated trajectories contain robot states and rewards rather than human-subject data.