# DIAGNOSING AND REPAIRING WORLD-MODEL ERRORS FOR PLANNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

World models guide planning by predicting action consequences, yet lower average prediction error need not improve action selection. Relative errors in predicted returns can reverse candidate rankings even when average prediction error is small. We develop a support gate that couples penalties on positive predicted rewards to state and action support. A complementary pairwise action-effect loss supplements prediction training by matching differences between real next-state outcomes of alternative actions from the same state. Our analysis connects selection loss to relative score errors and explains how the two designs modify planning scores and relative outcome predictions. The gate reduces CartPole mean normalized regret from 0.676 to 0.077 without retraining, and improves return in 20 of 27 models across nine public-data settings. Component comparisons show further gains from pairwise training with the gate on DoorKey and CartPole, with the full combination performing best among the evaluated configurations. An independent acquisition study under matched interaction budgets finds that additional data can improve control beyond continued training.

## 1 INTRODUCTION

A planning agent uses a world model to compare actions before executing them. Probabilistic ensembles enable effective model-based control (Chua et al., 2018), and latent models extend it to larger observation spaces (Ha & Schmidhuber, 2018; Hafner et al., 2025; Hansen et al., 2024). However, accurate predictions need not improve action selection (Lambert et al., 2020). Consider a robot choosing between long strides (A) and short, stable strides (B), as illustrated in Figure 1(a). Its model predicts greater progress from A, but the longer strides may cause it to lose balance, making B the better choice in practice. This example suggests two ways to improve action selection: discount predicted gains with weak data support, and train the model to better distinguish action outcomes. As shown in Figure 1(b), changing the planning score while keeping the model fixed reduces CartPole's mean normalized regret from 0.676 to 0.077.

Conservative model-based methods use uncertainty or support to limit unreliable predictions (Yu et al., 2020; Kidambi et al., 2020; Zhan et al., 2022), while decision-relevant model learning emphasizes errors that affect downstream values (Farahmand et al., 2017; Grimm et al., 2020; Voelcker et al., 2022). These foundations motivate two inputs to action comparison: data support indicates where predictions may be unreliable, while real outcomes of alternative actions from the same state expose differences the predictor should preserve.

We develop a support gate and a pairwise action-effect loss. The gate couples penalties to lack of support and positive predicted reward without updating the model. The loss uses real branches from the same state to fit differences between action consequences. We analyze how relative candidate-score errors affect selection, how gating transforms rewards, and which errors pairwise supervision identifies. The gate improves 20 of 27 model means across nine Minari settings; the full combination achieves the highest DoorKey solved count and lowest CartPole regret among the six component configurations. An independent extension compares seven acquisition strategies with the training objective and planner fixed.

**Summary of Contributions: (1)** We develop support-gated planning and a pairwise action-effect loss, and analyze how they modify planning scores and relative outcome predictions, respectively.



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

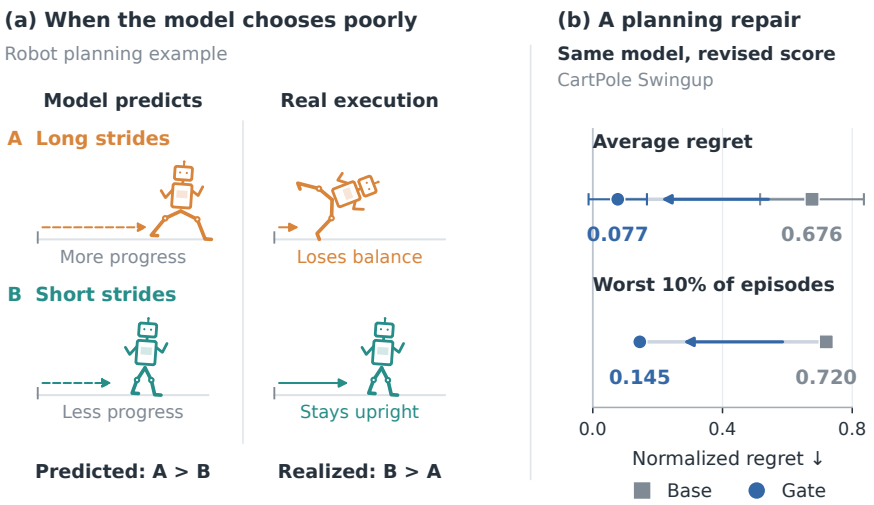



Figure 1: **Repairing model-based action selection.** (a) Robot schematic: predicted and realized stride rankings disagree. (b) Fixed CartPole models with original (Base) or gated (Gate) scores. Training data replace expert actions with random actions with probability $p = 0.5$. Normalized regret (↓) is $(J_{\text{ref}} - J)/(|J_{\text{ref}}| + 10^{-8})$, where $J$ is episode return and $J_{\text{ref}}$ the larger mean return of true-simulator planning and expert control on the same twenty resets. Points average five training runs sharing a dataset (twenty episodes/run); bars show across-run SD of mean regret. Worst-decile regret averages each run's two largest episode regrets, then runs. Full comparison: Table 1.

**(2)** Fixed-model and public-data experiments demonstrate control gains from gated planning; component comparisons show further gains from pairwise training with the gate on both tasks. **(3)** We demonstrate that acquiring additional real data can improve control beyond continued training, using matched-budget comparisons, training controls, and confirmation on independent sources.

## 2 Related Work

**Support and conservative planning.** MOPO penalizes uncertain model rewards, while MOReL introduces an absorbing pessimistic state for unknown queries (Yu et al., 2020; Kidambi et al., 2020). MOPP adapts offline planning through behavior-guided trajectories and uncertainty-based pruning (Zhan et al., 2022). Other methods use conservative value learning, adversarial models, or model–Bellman inconsistency (Yu et al., 2021; Rigter et al., 2022; Sun et al., 2023). Our gate combines state proximity with local action support and scales the penalty with claimed positive reward. Design-choice and search studies further motivate evaluating model use alongside prediction accuracy (Lu et al., 2022; Chang et al., 2026).

**Decision-relevant model learning.** Objective mismatch separates predictive training from control objectives (Lambert et al., 2020). Value-aware learning compares models through downstream values (Farahmand et al., 2017); value equivalence uses their supported value updates (Grimm et al., 2020), and VaGraM weights prediction errors by value gradients (Voelcker et al., 2022). Our pairwise loss directly fits normalized differences between real action outcomes at a shared state, without learning a value function for the loss. CoCo also uses same-state multi-action supervision to address statistical bias in visual modeling and planning (Shi et al., 2026). We study relative-effect supervision in state-space ensembles alongside planning-score and data controls.

**Rollout error and data acquisition.** Multistep and self-correcting models address rollout errors (Asadi et al., 2019; Talvitie, 2017), while reward learning under model misspecification distinguishes state accuracy from useful returns (Talvitie, 2018). Dataset aggregation targets visited states (Ross et al., 2011); short rollouts limit accumulated error (Janner et al., 2019). Active model learning selects informative trajectories (Zhang et al., 2021), continuous-time exploration adapts observation timing (Treven et al., 2023), and COPlanner combines conservative imagined rollouts with opti-

mistic collection (Wang et al., 2024). Our extension tests temporal diagnostics as a guide to label allocation, accounting for traversal and branch costs and comparing against continued training and local sampling.

## 3 Problem Formulation and Diagnosis

We consider a world model parameterized by $\theta$, with next-state predictor $f_\theta$, reward predictor $\widehat{r}_\theta$, and termination predictor. At state $s$, the planner predicts the outcomes of action sequences $\mathbf{a} = (a_0, \ldots, a_{H-1})$, executes its first selected action, and replans from the observed state. Our analysis fixes the candidate set $\mathcal{C}$; deployment also generates and refines candidates.

From a common initial state, let $G(\mathbf{a})$ denote true return from simulator replay and $\widehat{G}(\mathbf{a})$ predicted return. Define the best candidate $\mathbf{a}^* \in \arg\max_{\mathcal{C}} G$ and the model's choice $\widehat{\mathbf{a}} \in \arg\max_{\mathcal{C}} \widehat{G}$. Their selection loss is

$$L_{\text{sel}} = G(\mathbf{a}^*) - G(\widehat{\mathbf{a}}). \tag{1}$$

This loss is relative to the evaluated candidates; normalization divides by their nonzero true-return range. Closed-loop return instead sums rewards from repeated planning and execution, with changing states and candidates. Writing $e(\mathbf{a}) = \widehat{G}(\mathbf{a}) - G(\mathbf{a})$, the model's preference for its choice over $\mathbf{a}^*$ gives

$$0 \le L_{\text{sel}} \le e(\widehat{\mathbf{a}}) - e(\mathbf{a}^*) \le 2 \max_{\mathbf{a} \in \mathcal{C}} |e(\mathbf{a})|. \tag{2}$$

The rightmost term bounds selection loss by twice the worst absolute score error; the middle term measures optimism about the selected candidate relative to the best alternative. A common additive score offset leaves ordering unchanged, whereas lower average prediction error can coexist with an error difference that reverses an important pair. The fixed-candidate derivation and tie convention appear in Appendix A. This motivates reducing credit for weakly supported gains during planning and emphasizing same-state action differences during training. Gate scoring modifies predicted returns; pairwise training modifies predicted state consequences. Their connection depends on downstream rewards and rollouts, which motivates evaluating actual control alongside prediction accuracy.

## 4 Part I: Planner and Model Repair with Given Data

### 4.1 Available Supervision and the Repair Procedure

Let $D_\mu$ denote transitions collected under behavior policy $\mu$. A branch package $B = \{(s_i, a_i, s'_i, r_i, d_i, g_i)\}_{i=1}^{N_B}$ contains $N_B$ real simulator transitions: $s_i$, $a_i$, $s'_i$, and $r_i$ denote the starting state, action, observed next state, and reward. The binary $d_i \in \{0, 1\}$ indicates environment termination, excluding time-limit truncation. Group identifier $g_i$ links actions executed from the same full simulator snapshot and time input: $g_i = g_j$ implies $s_i = s_j$; different groups may start from different states. Branches originate at behavior-visited states or along planner-proposed paths. Ordinary prediction training uses $D_\mu \cup B$; pairwise training additionally uses the groups to compare actions. Gate-only planning can use the original model and support library from $D_\mu$.

Figure 2 summarizes the two interventions. Section 5 distinguishes shared-label comparisons from those that also change data and retraining.

### 4.2 A Support Gate for Planning Scores

Poorly supported predictions can give a candidate a high score. MOPO penalizes rewards using model uncertainty (Yu et al., 2020); subsequent analysis identifies interactions between uncertainty proxies, penalty strength, and rollout horizon (Lu et al., 2022). Ensemble agreement can coexist with shared errors, motivating training-data support as a coverage measure.

A common action can occur at an unfamiliar state, so we assess both action and state support. The action score $u_a(s, a) = -\log \widehat{q}(a \mid s)$ uses the conditional action probability (discrete) or density (continuous) estimated from neighboring training samples. The state score $u_s(s)$ averages distances to neighboring training states after standardization. Together they distinguish unusual actions at familiar states from unfamiliar states. Let $\mu_a$ and $\mu_s$ denote the calibration-input averages of $u_a$ and

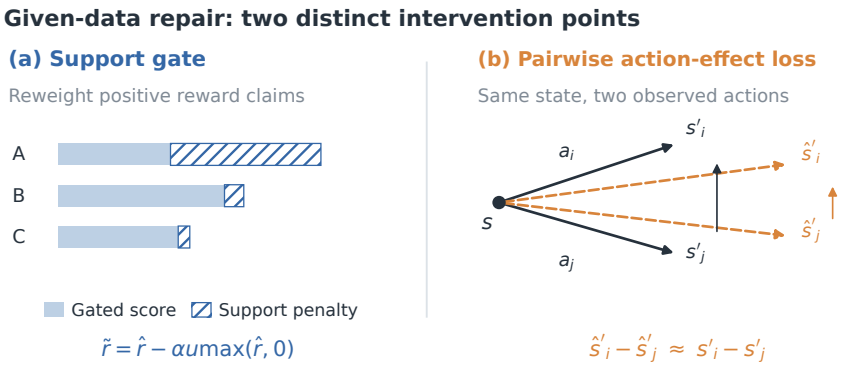



Figure 2: **Gate and pairwise supervision.** (a) For positive predicted rewards and $0 \leq \alpha u \leq 1$, full bars show original scores; solid and hatched segments show gated scores and penalties. Here $\alpha$ controls penalty strength and $u$ measures lack of support. (b) Pairwise supervision fits predicted differences between two same-state action outcomes to their observed differences.

$u_s$, respectively. These means normalize the two scores, which measure different quantities, in their implemented combination:

$$u(s,a) = \tfrac{1}{2}\left[\frac{u_a(s,a)}{\mu_a} + \frac{u_s(s)}{\max(\mu_s, 0.05)}\right]. \tag{3}$$

The state-scale floor limits amplification near the support library; Appendices A.2 and B.1 specify estimation and calibration, respectively. Positive additive penalties can accumulate even without predicted gains, potentially favoring shorter paths. We instead couple penalties to positive predicted rewards. With penalty strength $\alpha$, the planning reward is

$$\tilde{r}(s,a) = \hat{r}_\theta(s,a) - \alpha u(s,a) \max(\hat{r}_\theta(s,a), 0), \qquad \alpha \geq 0. \tag{4}$$

For fixed positive support score, larger positive rewards receive larger absolute penalties; zero rewards add none. Only planning scores change: model parameters, predicted states, and termination remain fixed for a given action sequence. The planner searches using gated scores and executes its first selected action. For scalar predicted reward $r$ and support score $u$, define the transformation $T_\alpha(r,u)$:

$$T_\alpha(r,u) = \begin{cases} r, & r \leq 0, \\ (1-\alpha u)r, & r > 0. \end{cases} \tag{5}$$

When $0 \leq \alpha u \leq 1$, positive rewards receive a proportional discount. At fixed $\alpha$ and $u$, positive reward scaling by $c$ gives $T_\alpha(cr,u) = cT_\alpha(r,u)$; this does not require interpreting support as a calibrated prediction-error probability. The implemented multiplier is unclipped: $\alpha u > 1$ makes positive rewards negative, while $u < 0$ amplifies them. Continuous-action density scores can be negative, so discounting is conditional. Appendix A details these cases and the calibration denominator.

Let $P(\mathbf{a})$ sum $\alpha u \max(\widehat{r}, 0)$ over a fixed candidate's model-active steps. Its gated score and error are $\widehat{G}_g = \widehat{G} - P$ and $e_g = e - P$. If $\widehat{\mathbf{a}}_g$ maximizes this score over the same $\mathcal{C}$, its selection loss $L^g_{\text{sel}} = G(\mathbf{a}^*) - G(\widehat{\mathbf{a}}_g)$ satisfies

$$L^g_{\text{sel}} \leq [e(\widehat{\mathbf{a}}_g) - P(\widehat{\mathbf{a}}_g)] - [e(\mathbf{a}^*) - P(\mathbf{a}^*)]. \tag{6}$$

Differences in support penalties adjust relative candidate scores, accounting for both data coverage and claimed positive gains. We evaluate this scoring intervention through realized return and model optimism.

### 4.3 Pairwise supervision for relative action effects

Objective mismatch motivates control-relevant training beyond average prediction accuracy (Lambert et al., 2020). Value-aware learning compares models through downstream value functions

(Farahmand et al., 2017); VaGraM weights consequential state errors using value gradients (Voelcker et al., 2022). We instead supervise relative action effects directly from observed outcomes, without requiring a learned value function or model-generated preferences.

Restoring the simulator to the same state for actions $a_i$ and $a_j$ yields true next states $s'_i$ and $s'_j$. Their difference $s'_i - s'_j$ is the action-effect target. An ordinary loss fits each outcome separately; the pairwise term explicitly fits the contrast predicted between alternative actions, using the same observed transition labels. Let $D = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$ contain next-state training scales. The normalized absolute difference loss is

$$\mathcal{L}_{\text{pair}} = \mathbb{E}_{(i,j)} \left\| D^{-1} \left[ f_\theta(s, a_i) - f_\theta(s, a_j) - (s'_i - s'_j) \right] \right\|_1 . \tag{7}$$

Pairs share the full initial state and time input and use real branch labels. Normalization prevents coordinates with larger physical units from automatically dominating the loss. Continuous models compare next-state means; discrete models compare expectations of the numeric state encoding. Writing $e_i = f_\theta(s, a_i) - s'_i$ exposes the targeted error:

$$\mathcal{L}_{\text{pair}} = \mathbb{E}_{(i,j)} \| D^{-1}(e_i - e_j) \|_1 . \tag{8}$$

Adding a common displacement $b(s)$ to both predictions leaves the loss unchanged: pairwise training targets relative action-effect errors, while ordinary prediction training anchors individual outcomes. This invariance does not make shared state errors harmless to nonlinear rewards or later rollouts. Relative state and return errors remain distinct; their connection depends on downstream scoring.

If the shared downstream score is linear, $V(s') = v^\top s'$, the predicted action-value difference error satisfies

$$\left| \widehat{\Delta V}_{ij} - \Delta V_{ij} \right| \le \| Dv \|_\infty \, \| D^{-1}(e_i - e_j) \|_1 . \tag{9}$$

Here $\Delta V_{ij} = v^\top (s'_i - s'_j)$ and $\widehat{\Delta V}_{ij} = v^\top [f_\theta(s, a_i) - f_\theta(s, a_j)]$. By dual norms, the residual $\| D^{-1}(e_i - e_j) \|_1$ also equals the largest absolute score-difference error over linear scores with $\| Dv \|_\infty \le 1$. This links the loss to comparisons under a common linear score. Nonlinear multistep rewards and termination require additional analysis; deployment experiments evaluate these effects.

### 4.4 Training and Deployment

The ordinary objective fits dynamics, reward, and termination using transition labels, including branches. Each member completes ordinary minibatches before additional Adam pair updates each epoch, using pairwise weight one and at most ten pairs per group. These are separate optimizer steps. Through the shared network, pairwise updates can also affect reward and termination predictions. Algorithm 1 gives the procedure; Appendices A.3 and B.1 specify pairwise normalization and training settings. Deployment replans with CEM for continuous actions or bounded BFS for DoorKey, scoring rollouts to the planning horizon or predicted termination using Equation 4.

**Algorithm 1** Gate and pairwise repair

**Given:** $D_\mu$, $B$, calibration inputs, $\alpha$
Build $D_\mu$ support; calibrate scales
Fit normalizers on $D_\mu \cup B$; sample pairs
**for** each member and epoch **do**
  Ordinary updates on $D_\mu \cup B$
  Additional Adam updates using $\mathcal{L}_{\text{pair}}$
**end for**
**for** each real decision **do**
  Search plans using gated rollouts
  Execute the best plan's first action; observe state
**end for**

Pairwise updates and gated planning can be enabled separately. Component comparisons retain branch labels when removing pairwise updates, so they evaluate recipes with different update budgets (Section 5.3).

## 5 Part I: Empirical Evaluation

### 5.1 Tasks, Models, and Evaluation

We use five-member state-space neural ensembles. The component study evaluates DoorKey, requiring a key and an unlocked door to reach the goal, and DMC CartPole Swingup (CP), requiring

pole swingup and balancing (Tunyasuvunakool et al., 2020). Gate-only evaluation uses Minari's simple, medium, and expert datasets for HalfCheetah (HC), Hopper, and Walker2d (Younis et al., 2024). These MuJoCo locomotion tasks (Todorov et al., 2012) test planning across data qualities without same-state branch supervision.

For the component data, $p \in \{0.05, 0.5\}$ is the probability of replacing an expert action with a random action during collection. It changes the data distribution rather than an isolated scalar measure of coverage. The component protocol uses five training seeds sharing the original dataset per task and mixture. Each run collects its own planner-path branches; training variants share that run's branch package. Minari uses three model seeds sharing each dataset. We average deployment episodes within each model before aggregation: twenty episodes for CP, one deterministic reset for DoorKey. Implementation and pairwise normalization appear in Appendices B.1 and A.3.

Closed-loop return evaluates real execution. For the component comparison, normalized regret is $(J_{\text{ref}} - J)/(|J_{\text{ref}}| + 10^{-8})$, where $J$ is the realized return. DoorKey uses a shortest-path reference; CP uses the stronger of true-simulator planning and an expert controller on the matched resets. The model exploitation gap (MEG) compares the unpenalized model prediction with the true return for the same chosen action sequence. Positive MEG indicates optimism, and negative MEG indicates pessimism.

Support scales use held-out IID test states and actions; the original-data support library remains fixed after branch retraining. We use $\alpha = 1.0$ for DoorKey and $\alpha = 0.3$ for CP. All Minari environments, dataset tiers, and model seeds share $\alpha = 0.3$. Calibration settings and gate strengths appear in Appendix B.1.

### 5.2 Support-gated planning

We first compare five planning scores on fixed CP checkpoints at $p = 0.5$: the base score, ensemble-disagreement (Ens.) and action-support (Sup.) penalties, their additive combination (Ens.+Sup.), and the gate (Table 1).

Table 1: **Planning scores on fixed CP models ($p = 0.5$).** Normalized regret (NRegret): mean ± SD across five model means, twenty episodes/model. Worst-decile averages each model's two largest episode regrets, then models. MEG is the signed, unpenalized prediction–return gap. Bold marks minimum regret.

| Metric | Base | Ens. | Sup. | Ens.+Sup. | Gate |
|---|---|---|---|---|---|
| NRegret ↓ | 0.676±0.16 | 0.682±0.27 | 0.524±0.06 | 0.496±0.07 | **0.077±0.09** |
| Worst-decile ↓ | 0.720 | 0.712 | 0.654 | 0.550 | **0.145** |
| MEG | +2.32 | +0.57 | +1.93 | +0.30 | +0.02 |

The gate achieves the lowest mean and worst-decile normalized regret (0.077 and 0.145), compared with base (0.676 and 0.720) and the additive combination (0.496 and 0.550), extending gains to the poorest episodes (Figure 1(b)). Paired 95% bootstrap intervals for gate-minus-base and gate-minus-combined-penalty regret both lie below zero (Appendix B.2). Signed MEG falls from $+2.32$ to $+0.02$, indicating less model overestimation. These comparisons jointly change state support and penalty form. On original DoorKey models at $p = 0.5$, the gate raises solved runs from zero to three of five; gains also occur with pairwise-trained planner-path models (Table 4).

The public-data evaluation uses 27 fixed models across nine task–tier settings, totaling 1,080 episodes. Figure 3 shows paired returns; Table 2 reports group means, variation across models, and MEG. Planner settings and model-level records appear in Appendix B, Table 7. The gate improves seven of nine task–tier means and 20 of 27 model means without retraining. All three HC tiers improve, with two of three models improving in each tier. Hopper simple and expert and Walker2d simple and expert each improve in all three models. Hopper expert, for example, rises from 4.55 to 103.63. The declining means are Hopper medium, from 67.98 to 5.29, and Walker2d medium, from 12.72 to $-0.79$. In the latter, one larger loss outweighs two gains, demonstrating why both individual pairs and group means matter.

Table 3 uses fixed checkpoints: IID NMSE remains unchanged, so control gains arise from planner behavior. Planner-path counterfactual error (CFE) also depends on each planner's visited states

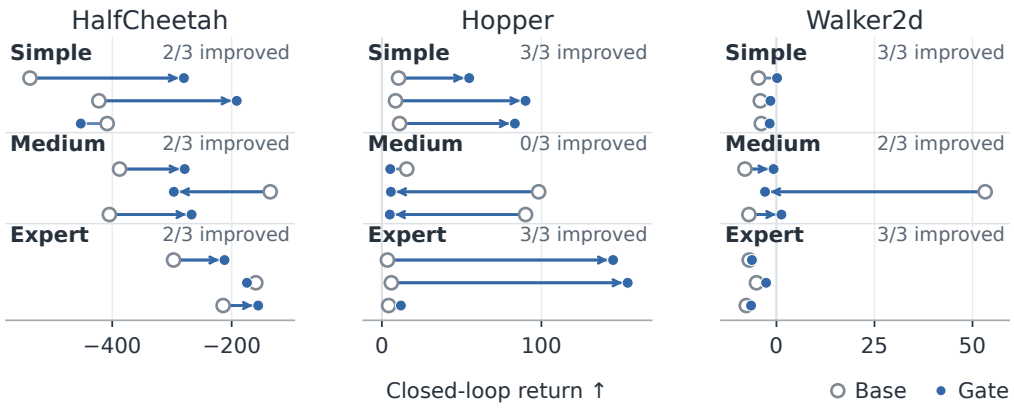



Figure 3: **Gate effects on Minari tasks.** Each row connects a model's Base (open gray) and Gate (filled blue) returns, averaged over twenty episodes per planner; rightward indicates improvement. Within tiers, rows follow seeds 0–2 from top to bottom, with improvement counts shown. Return scales differ across environments and are shared across tiers.

Table 2: **Gate performance on Minari.** Returns (↑): mean ± sample SD across three model means (twenty episodes/model). MEG is the unpenalized prediction–return gap: positive indicates optimism; negative, pessimism. Improved counts models with higher gated return; bold marks the higher mean within each row.

| Task | Tier | Base return | Gate return | Base MEG | Gate MEG | Improved |
|---|---|---|---|---|---|---|
| HalfCheetah | simple | $-456.19 \pm 70.96$ | $\mathbf{-308.05} \pm 132.82$ | 134.17 | 26.85 | 2/3 |
| HalfCheetah | medium | $-309.29 \pm 150.81$ | $\mathbf{-280.84} \pm 14.97$ | 150.85 | 2.92 | 2/3 |
| HalfCheetah | expert | $-223.74 \pm 69.35$ | $\mathbf{-180.64} \pm 28.69$ | 243.64 | 20.88 | 2/3 |
| Hopper | simple | $10.04 \pm 1.31$ | $\mathbf{76.04} \pm 18.76$ | 45.46 | 25.43 | 3/3 |
| Hopper | medium | $\mathbf{67.98} \pm 45.55$ | $5.29 \pm 0.33$ | 68.55 | 0.28 | 0/3 |
| Hopper | expert | $4.55 \pm 1.24$ | $\mathbf{103.63} \pm 79.58$ | 83.60 | 37.61 | 3/3 |
| Walker2d | simple | $-4.18 \pm 0.36$ | $\mathbf{-1.03} \pm 1.04$ | 173.44 | -3.36 | 3/3 |
| Walker2d | medium | $\mathbf{12.72} \pm 35.08$ | $-0.79 \pm 2.09$ | 66.02 | 56.22 | 2/3 |
| Walker2d | expert | $-6.57 \pm 1.34$ | $\mathbf{-5.11} \pm 2.16$ | 153.80 | 100.91 | 3/3 |

Table 3: **Prediction and selection diagnostics.** Entries average three model means; lower is better throughout. IID: held-out normalized mean squared prediction error, identical for Base and Gate on fixed checkpoints. CFE: scaled prediction errors in same-state action-effect differences, evaluated at each planner's visited states. NRegret: normalized regret relative to each environment's expert-dataset mean.

| Task | Tier | IID | Base CFE | Gate CFE | Base NRegret | Gate NRegret |
|---|---|---|---|---|---|---|
| HalfCheetah | simple | 0.0239 | 1.4667 | 1.2931 | 1.0281 | 1.0190 |
| HalfCheetah | medium | 0.0167 | 0.7098 | 0.5453 | 1.0190 | 1.0173 |
| HalfCheetah | expert | 0.0874 | 1.0959 | 0.7429 | 1.0138 | 1.0111 |
| Hopper | simple | 0.0034 | 0.1866 | 0.2172 | 0.9974 | 0.9803 |
| Hopper | medium | 0.0025 | 0.2517 | 0.2144 | 0.9824 | 0.9986 |
| Hopper | expert | 0.0031 | 0.1317 | 0.1320 | 0.9988 | 0.9731 |
| Walker2d | simple | 0.0289 | 1.5190 | 1.4481 | 1.0006 | 1.0001 |
| Walker2d | medium | 0.0273 | 1.4706 | 1.4252 | 0.9981 | 1.0001 |
| Walker2d | expert | 0.0088 | 0.8981 | 0.9165 | 1.0010 | 1.0007 |

(Appendix B). CFE decreases in six of nine settings. HC simple, medium, and expert reduce CFE by 11.8%, 23.2%, and 32.2%, respectively, while improving return. With the model fixed, this is consistent with gated planning reaching regions where relative action effects are predicted more accurately. Signed MEG also decreases in all nine group means and 26 of 27 models; HC medium falls from 150.85 to 2.92. MEG uses unpenalized predictions, so this reflects the plans selected rather than direct subtraction of the penalty. Realized return provides the primary evidence of control gains, with CFE and MEG describing prediction errors and optimism along the selected behavior. Hopper simple improves return even as CFE rises from 0.1866 to 0.2172. Changes in MEG likewise need not track changes in return, motivating a joint interpretation of these diagnostics and control outcomes.

### 5.3 Pairwise action-effect supervision and component comparisons

Table 4 compares six configurations at $p = 0.5$. Planner-path variants share labels, although pairwise minibatches add optimizer steps. The full combination performs best on both tasks. Adding pairwise updates to planner-path training with the gate raises DoorKey solved runs from one to four of five and lowers CP regret from 0.231 to 0.045. Adding the gate to the pairwise-trained model raises DoorKey's solved count from zero to four and lowers CP regret from 0.563 to 0.045. The combination also improves on gate-only (three solved runs; CP regret 0.077), supporting joint use of relative supervision and gated planning.

Table 4: **Component comparisons ($p = 0.5$).** Five runs per configuration; DK denotes DoorKey. Pools combine original data $D_\mu$ with behavior ($B_b$) or planner-path ($B_p$) branches. "Loss" adds pairwise optimizer updates to ordinary training; retraining changes the model and normalization. Bold marks the best value per metric.

| Configuration | Pool | Retrain | DK solved ↑ | CP NRegret ↓ |
|---|---|---|---|---|
| Base | $D_\mu$ | No | 0/5 | 0.676 |
| Base + gate | $D_\mu$ | No | 3/5 | 0.077 |
| Behavior + loss + gate | $D_\mu + B_b$ | Yes | 3/5 | 0.238 |
| Planner + gate | $D_\mu + B_p$ | Yes | 1/5 | 0.231 |
| Planner + loss | $D_\mu + B_p$ | Yes | 0/5 | 0.563 |
| Full recipe | $D_\mu + B_p$ | Yes | **4/5** | **0.045** |

We next decompose standardized next-state errors across actions into their mean (the common component) and deviations from it (the relative component). Pairwise supervision responds to the latter (Equation 8). Table 5 removes the common component from frozen-model predictions using simulator truth. In CP, it accounts for 98.11% of aggregate error energy, yet 98.98% of candidate choices remain unchanged after removal. In HC, it accounts for 81.85%; removal improves 15.51% of choices and worsens 13.69%. A large state-error component need not dominate selection, although common errors can affect choices. Pairwise supervision complements ordinary training by penalizing errors in predicted same-state outcome differences.

Table 5: **Common error and action selection.** Common energy is summed common-error energy divided by summed total error energy. Choice columns report lower, unchanged, or higher selection loss after oracle removal of the common component. Each task pools 5,120 dependent cases from twenty models and sixteen shared source episodes across four training mixtures.

| Task | Common energy | Choice improved | Unchanged | Worsened |
|---|---|---|---|---|
| CP | 98.11% | 0.45% | 98.98% | 0.57% |
| HC | 81.85% | 15.51% | 70.80% | 13.69% |

## 6 Part II: Model Repair through Budgeted Acquisition

Having examined how planning scores and action-effect supervision affect control, we next study whether additional real data can yield gains beyond continued training. State-correction diagnostics show that correcting predictions at different rollout depths can affect candidate selection (Ap-

pendix C), motivating collection along planned paths. We compare trajectory-based and local collection against continued training on existing data.

### 6.1 Acquisition and training controls

Local collects short rollouts from a shared starting state, whereas Uniform distributes branches along a planned trajectory. We first run the agent for 300 steps with the original model, saving six simulator states as collection starting points for both methods. From each state, Local collects fifteen eight-step rollouts, resetting to that state each time. Uniform follows a 40-step trajectory planned by the frozen model and collects two eight-step branches at each of five evenly spaced positions. Both methods collect 120 transitions per starting state, totaling $300 + 6 \times 120 = 1{,}020$ including navigation. All navigation, traversal, and branch transitions are used with existing data to update the dynamics model. The planner settings, reward function, and normalization stay fixed. Each ensemble member receives 500 updates using U loss, the squared prediction error in normalized one-step state changes. Frozen retains the original model; old-only matches updates using existing data. The common-label control retains only navigation and full-path transitions from the collected data to assess the contribution of additional branches.

### 6.2 Control gains on independent sources

Confirmation uses five new datasets and initializations per HC/CP and $p \in \{0.05, 0.5\}$ cell. Each of frozen, old-only, local, and uniform receives three paired 1,000-step deployments, totaling 240 (Figure 4). In HC, both acquisition strategies outperform frozen and old-only in mean return at both mixtures. At $p = 0.5$, local and uniform reach 1856.33 and 2403.63, versus 601.85 and 603.62 for frozen and old-only. Uniform has the highest mean at both mixtures. Comparison with old-only supports acquisition benefits beyond continued training, with variation across sources. In CP, local has the highest mean return at $p = 0.05$ (137.26), whereas old-only leads at $p = 0.5$ (377.08). Acquisition benefits over continued training therefore depend on task and initial data. Paired comparisons, development results, and training controls appear in Appendix D.

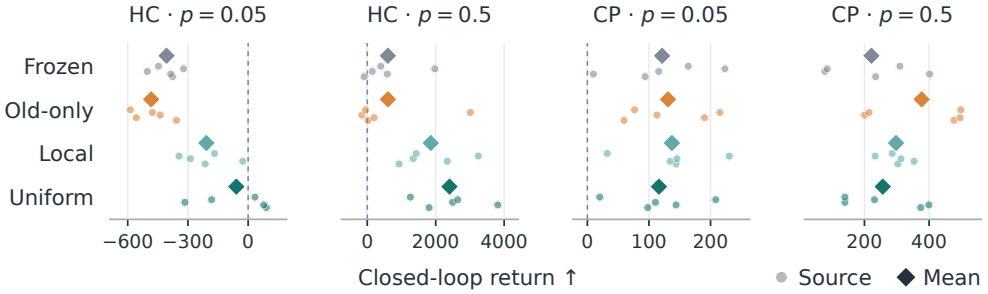



Figure 4: **Acquisition results on new sources.** HalfCheetah (HC) and CartPole (CP); $p$ is the probability of replacing an expert action with a uniformly random action during original data collection. Each condition shows five independent sources: dots average three paired 1,000-step resets; diamonds average the five sources. Horizontal return scales differ across panels.

## 7 Conclusion

We develop and analyze support-gated planning and pairwise action-effect supervision, connecting data support and relative prediction errors to action selection. Fixed-model and public-data evaluations show control gains from the gate, while the full combination performs best on both component-study tasks at $p = 0.5$. An independent acquisition study confirms HalfCheetah gains beyond continued training, with uncertain advantages from branch placement and different outcomes in CartPole. These results link explicit changes in planning and learning to their deployment effects. Broader model evaluations, independent gate calibration, and matched training controls can further assess which benefits generalize across tasks and data conditions.

## AI USE STATEMENT

We used AI tools to assist with code implementation and debugging, LaTeX formatting, and language polishing. The authors are responsible for the final content.

## REFERENCES


Kavosh Asadi, Dipendra Misra, Seungchan Kim, and Michael L. Littman. Combating the compounding-error problem with a multi-step model. *arXiv preprint arXiv:1905.13320*, 2019. URL https://arxiv.org/abs/1905.13320.

Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.21306.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

Amir-massoud Farahmand, André M. S. Barreto, and Daniel N. Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1486–1494. PMLR, 2017. URL https://proceedings.mlr.press/v54/farahmand17a.html.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5541–5552. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, pp. 47376–47405, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21810–21823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f7efa4f864ae9b88d43527f4b14f750f-Abstract.html.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 761–770. PMLR, 2020. URL https://proceedings.mlr.press/v120/lambert20a.html.

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.

Marc Rigter, Bruno Lacerda, and Nick Hawes. RAMBO-RL: Robust adversarial model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16082–16097. Curran Associates, Inc., 2022. doi: 10.52202/068431-1170.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Yuhong Shi, Zhenhao Chu, Jie Wei, Jun Hao, Jianyi Liu, and Jingwen Fu. Overcoming statistical bias in action-controllable world models. *arXiv preprint arXiv:2608.04653*, 2026. URL https://arxiv.org/abs/2608.04653.

Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-Bellman inconsistency for model-based offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33177–33194. PMLR, 2023. URL https://proceedings.mlr.press/v202/sun23q.html.

Erik Talvitie. Self-correcting models for model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pp. 2597–2603, 2017. doi: 10.1609/aaai.v31i1.10850.

Erik Talvitie. Learning the reward function for a misspecified model. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4838–4847. PMLR, 2018. URL https://proceedings.mlr.press/v80/talvitie18a.html.

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

Lenart Treven, Jonas Hübotter, Bhavya Sukhija, Florian Dörfler, and Andreas Krause. Efficient exploration in continuous-time model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 42119–42147. Curran Associates, Inc., 2023. doi: 10.52202/075280-1825.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. doi: 10.1016/j.simpa.2020.100022.

Claas A. Voelcker, Victor Liao, Animesh Garg, and Amir-massoud Farahmand. Value gradient weighted model-based reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=4-D6CZkRXxI.

Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. COPlanner: Plan to roll out conservatively but to explore optimistically for model-based RL. In *International Conference on Learning Representations*, pp. 17089–17110, 2024. URL https://openreview.net/forum?id=jnFcKjtUPN.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K. Terry. Minari. Zenodo, 2024. URL https://doi.org/10.5281/zenodo.13767625. Software, version 0.5.0.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James 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_files/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: Conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28954–28967. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f29a179746902e331572c483c45e5086-Abstract.html.

Xianyuan Zhan, Xiangyu Zhu, and Haoran Xu. Model-based offline planning with trajectory pruning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 3716–3722. International Joint Conferences on Artificial Intelligence Organization, 2022. doi: 10.24963/ijcai.2022/516.

Weixuan Zhang, Marco Tognon, Lionel Ott, Roland Siegwart, and Juan Nieto. Active model learning using informative trajectories for improved closed-loop control on real robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4467–4473. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9560869.

# A Mechanism Analysis and Implementation Scope

These algebraic properties describe the Part I scoring and training changes; their closed-loop benefit is evaluated empirically.

## A.1 Relative score error and fixed-candidate selection

Fix an initial simulator snapshot and finite candidate set $\mathcal{C}$. Let $\mathbf{a}^* \in \arg\max_{\mathbf{a}\in\mathcal{C}} G(\mathbf{a})$, $\widehat{\mathbf{a}} \in \arg\max_{\mathbf{a}\in\mathcal{C}} \widehat{G}(\mathbf{a})$ with fixed tie-breaking, and $e(\mathbf{a}) = \widehat{G}(\mathbf{a}) - G(\mathbf{a})$. Then

$$\begin{aligned} L_{\text{sel}} &= e(\widehat{\mathbf{a}}) - e(\mathbf{a}^*) - \left[\widehat{G}(\widehat{\mathbf{a}}) - \widehat{G}(\mathbf{a}^*)\right] \\ &\le e(\widehat{\mathbf{a}}) - e(\mathbf{a}^*) \le \max_{\mathbf{a}\in\mathcal{C}} e(\mathbf{a}) - \min_{\mathbf{a}\in\mathcal{C}} e(\mathbf{a}) \le 2\max_{\mathbf{a}\in\mathcal{C}} |e(\mathbf{a})|. \end{aligned} \tag{10}$$

The first inequality uses score maximization; an approximate maximizer adds the gap $\delta = \max_{\mathbf{a}\in\mathcal{C}} \widehat{G}(\mathbf{a}) - \widehat{G}(\widehat{\mathbf{a}}) \ge 0$ to each upper bound. Common score errors cancel, whereas relative errors can reverse ordering. Common state displacements need not induce common score errors under nonlinear rewards. The comparator is the best candidate in $\mathcal{C}$; changing candidate sets and visited states prevents this identity from bounding a closed-loop episode.

## A.2 Exact support scores and reward transformation

The support library is a seed-zero subsample of at most 30,000 training transitions. Let $\mathcal{N}_k(s)$ be its $k = 32$ nearest states after standardization by original training state standard deviations $\sigma_s$, with coordinates below $10^{-6}$ assigned scale one. State support is

$$u_s(s) = \frac{1}{k} \sum_{n\in\mathcal{N}_k(s)} \|(s - s_n) \oslash \sigma_s\|_2 , \tag{11}$$

where $\oslash$ denotes coordinate-wise division. Action support is $u_a(s,a) = -\log \widehat{q}(a \mid s)$, with discrete-action estimate

$$\widehat{q}(a \mid s) = \frac{\sum_{n\in\mathcal{N}_k(s)} \mathbf{1}\{a_n = a\} + 0.5}{k + 0.5|\mathcal{A}|}. \tag{12}$$

For $d_a$-dimensional continuous actions, the local Gaussian density is

$$\begin{aligned} \widehat{q}(a \mid s) &= \frac{1}{k} \sum_{n\in\mathcal{N}_k(s)} \prod_{j=1}^{d_a} \frac{1}{\sqrt{2\pi}h_j} \exp\left[-\frac{(a_j - a_{n,j})^2}{2h_j^2}\right], \\ h_j &= \max\left(\widehat{\sigma}_{a,j} N^{-1/(d_a+4)}, 0.05\right). \end{aligned} \tag{13}$$

Here $N$ and $\widehat{\sigma}_a$ refer to the sampled library. State distance detects remote states even when actions are locally common; neither support score is ensemble disagreement or calibrated prediction correctness.

With calibration means $\mu_a, \mu_s$, the recorded combination ($u$ in the main text) is

$$u_{\text{joint}}(s,a) = \frac{1}{2}\left[\frac{u_a(s,a)}{\mu_a} + \frac{u_s(s)}{\max(\mu_s, 0.05)}\right]. \tag{14}$$

Only the state denominator is floored; $u_{\text{joint}}$ is unclipped and requires $\mu_a \neq 0$. Discrete $u_a$ is nonnegative, but continuous densities can exceed one, making $u_a$ negative and dependent on action units. If $\mu_a < 0$, increasing $u_a$ decreases its normalized contribution. These conditions do not establish a calibration failure in a reported run. On the calibration pool,

$$\overline{u_{\text{joint}}} = \tfrac{1}{2}\left[1 + \frac{\mu_s}{\max(\mu_s, 0.05)}\right]. \tag{15}$$

For $\mu_a \neq 0$, this mean equals one only when the state floor is inactive.

For predicted reward $r$, support score $u$, and $\alpha \ge 0$, define $T_\alpha(r,u) = r - \alpha u \max(r, 0)$. Thus

$$T_\alpha(r,u) = \begin{cases} r, & r \le 0, \\ (1 - \alpha u) r, & r > 0. \end{cases} \tag{16}$$

Positive rewards are discounted for $0 \leq \alpha u \leq 1$, become negative for $\alpha u > 1$, and increase for $u < 0$, $\alpha > 0$. The factor is unclipped; discounting does not establish a lower bound on true reward.

For $c > 0$, $T_\alpha(cr, u) = cT_\alpha(r, u)$, preserving candidate order under reward rescaling when support, trajectories, and termination masks remain fixed. An additive penalty $r - \lambda u$ requires rescaling $\lambda$ too; the recorded baselines already calibrate it to reward scale, so homogeneity alone establishes no advantage. Nonpositive rewards receive no additional penalty. Inserting a zero-reward step preserves the sum if other rewards, support, and termination remain fixed; changing a path generally changes these quantities.

For a fixed candidate, let $b_h \in \{0, 1\}$ denote model-predicted activity:

$$P(\mathbf{a}) = \alpha \sum_h b_h u_{\text{joint}}(\widehat{s}_h, a_h) \max(\widehat{r}_h, 0). \tag{17}$$

Then $\widehat{G}_g = \widehat{G} - P$ and $e_g = e - P$, relative to the same true return. An error difference changes by $-[P(\mathbf{a}_i) - P(\mathbf{a}_j)]$: larger corrections can counter larger overestimates, but support is a proxy and selection can change. Substituting $e_g$ into Equation 10 describes the gated choice without guaranteeing improvement. Gating changes scores while retaining predicted dynamics, raw reward, and termination functions.

### A.3 Pairwise supervision and relative transition error

For actions at the same restored state with true outcomes $s'_i, s'_j$, define member errors $\epsilon_i = f_\theta(s, a_i) - s'_i$ and $\epsilon_j = f_\theta(s, a_j) - s'_j$. With $D = \text{diag}(\sigma_1, \ldots, \sigma_d)$, each pair contributes

$$\left\| D^{-1} \left[ (f_\theta(s, a_i) - f_\theta(s, a_j)) - (s'_i - s'_j) \right] \right\|_1 = \left\| D^{-1}(\epsilon_i - \epsilon_j) \right\|_1 . \tag{18}$$

Scales are training next-state standard deviations, not residual scales; coordinates below $10^{-3}$ receive scale one. Residual predictions are converted from normalized mean increments to state units before applying $D^{-1}$; the shared starting state and mean increment cancel in differences.

A common shift $b(s)$ leaves the loss unchanged; ordinary prediction training anchors absolute location. Zero loss equates errors within connected components of the measured action-pair graph, without ensuring absolute accuracy or correctness on unmeasured actions. Sampling at most ten pairs per group does not ensure connectivity. Targets are true simulator branches, not model preferences; equal actions yield zero loss only when their true outcomes agree.

For a one-step comparison with shared linear continuation $V(x) = v^\top x + c$ and no unequal direct-reward prediction error, Hölder's inequality gives

$$\begin{aligned} &\left| \left[ V(f_\theta(s, a_i)) - V(f_\theta(s, a_j)) \right] - \left[ V(s'_i) - V(s'_j) \right] \right| \\ &\quad = \left| v^\top (\epsilon_i - \epsilon_j) \right| \leq \|Dv\|_\infty \left\| D^{-1}(\epsilon_i - \epsilon_j) \right\|_1 . \end{aligned} \tag{19}$$

A true score gap exceeding this bound in magnitude cannot reverse sign through this transition error. The bound depends on $v$ and the measured pair; it does not cover nonlinear continuation, unequal reward errors, or general multistep returns. Unlike value-weighted losses, our objective does not use learned values or value gradients (Farahmand et al., 2017; Voelcker et al., 2022).

## B Part I: Implementation and Additional Results

### B.1 Training, calibration, and evaluation

The support scores in Appendix A use the original training library, which remains fixed after branch retraining. Calibration means use the first 5,000 available state–action inputs from each study's held-out IID test pool, without next-state or reward labels; this is not a separate calibration split. Gate strengths were chosen in the original within-task exploration: 1.0 for DoorKey and 0.3 for CP. All Minari tasks, tiers, and model seeds use 0.3.

Each member completes ordinary Adam minibatches before additional pairwise minibatches each epoch, with weight one, at most ten pairs per same-state group, and the same optimizer state. The

Table 6: **Recorded component-training settings.** Branch budgets count stored labels, excluding navigation costs. These historical settings remain to be checked against the updated component runs in Table 4.

| Setting | DoorKey | CartPole |
|---|---|---|
| Ensemble members | 5 | 5 |
| Hidden layers × width | $3 \times 256$ | $3 \times 256$ |
| Adam learning rate / batch size | $10^{-3}$ / 256 | $10^{-3}$ / 256 |
| Epochs | 20 | 30 |
| Original-transition cap | 200,000 | 500,000 |
| Branch-label budget | 20,000 | 50,000 |
| Evaluation episodes per model | 1 | 20 |

ordinary objective combines dynamics NLL (continuous) or categorical cross-entropy (discrete), reward MSE, and termination BCE. Pairwise updates affect the shared trunk and can therefore change reward and termination predictions. Discrete supervision uses expected numeric state values. Planner-path variants share branch labels; retraining reinitializes the model and fits normalization. In the recorded trainer, the seed is set after model construction and does not fully fix initialization. Five training runs share the original task–mixture dataset; episodes are averaged within each model before aggregation.

For normalized regret, DoorKey uses a BFS-optimal reference; a run is counted as solved when its normalized regret is below 0.5. CP uses the larger mean return of true-simulator CEM and a SAC expert on the same twenty resets. Minari uses each environment's expert-dataset mean: 16,242.92 for HC, 3,857.80 for Hopper, and 6,847.79 for Walker2d. The latter references are dataset statistics, not newly evaluated expert controllers. MEG compares unpenalized predictions with simulator returns for the same selected plan.

For a same-state branch group $g$, let $a_{g0}$ be its recorded reference action and $D = \mathrm{diag}(\sigma)$. Counterfactual error is

$$\mathrm{CFE} = \frac{1}{M} \sum_{g} \sum_{a \neq a_{g0}} \left\| D^{-1} \left[ (\widehat{s}'_{ga} - \widehat{s}'_{g0}) - (s'_{ga} - s'_{g0}) \right] \right\|_2 , \tag{20}$$

where $M$ counts non-reference branches. The reference action is the behavior action in behavior groups and the selected action in planner groups. Here $\sigma$ contains original-training next-state standard deviations, with values below $10^{-3}$ replaced by one. Planner CFE depends on visited states and uses $L_2$, whereas pairwise training uses $L_1$.

### B.2 Fixed-model planning comparisons

The two recorded CP comparisons supporting Table 1 have gate-minus-base mean regret $-0.599$ with 95% interval $[-0.776, -0.377]$, and gate-minus-additive-combination regret $-0.419$ with interval $[-0.515, -0.310]$. These percentile intervals use 10,000 bootstrap resamples of five paired model differences, with bootstrap seed zero. The comparison changes both penalty form and support information: the additive support penalty uses action support, whereas the gate also uses state distance.

### B.3 Public-data protocol and model-level results

Minari ensembles train for fifteen epochs on at most 900,000 original transitions. CEM uses horizon 40, 500 candidates, 50 elites, and five iterations. Each task–tier shares one dataset across three model seeds, with twenty episodes per model–planner condition. The 54 result records contain 1,080 returns; means and sample SD aggregate model means. IID errors use original-training next-state standard deviations. Table 7 retains all pairs underlying the main-text group summaries and Figure 3.

Table 7: **All 27 public-data model pairs.** Returns average twenty episodes per model and planner. $\Delta J = J_{\text{gate}} - J_{\text{base}}$; positive differences favor the gate. MEG is signed; bold marks the higher return in each pair. Seeds share their task–tier dataset.

**HalfCheetah**

| Tier | Seed | Base return | Gate return | Δ return | Base MEG | Gate MEG |
|---|---|---|---|---|---|---|
| simple | 0 | -537.74 | **-279.96** | +257.78 | 83.75 | 3.31 |
| simple | 1 | -422.34 | **-191.52** | +230.83 | 194.98 | 73.21 |
| simple | 2 | **-408.50** | -452.67 | -44.17 | 123.79 | 4.02 |
| medium | 0 | -387.67 | **-278.63** | +109.05 | 155.79 | 2.52 |
| medium | 1 | **-135.43** | -296.80 | -161.37 | 162.27 | 2.88 |
| medium | 2 | -404.76 | **-267.10** | +137.66 | 134.49 | 3.34 |
| expert | 0 | -297.32 | **-211.95** | +85.37 | 257.97 | -4.90 |
| expert | 1 | **-159.59** | -174.35 | -14.77 | 223.00 | -1.37 |
| expert | 2 | -214.32 | **-155.61** | +58.71 | 249.96 | 68.90 |

**Hopper**

| Tier | Seed | Base return | Gate return | Δ return | Base MEG | Gate MEG |
|---|---|---|---|---|---|---|
| simple | 0 | 10.47 | **54.73** | +44.26 | 34.98 | 21.45 |
| simple | 1 | 8.58 | **90.04** | +81.46 | 45.98 | 16.55 |
| simple | 2 | 11.09 | **83.36** | +72.28 | 55.43 | 38.27 |
| medium | 0 | **15.59** | 5.25 | -10.35 | 53.24 | -1.15 |
| medium | 1 | **98.28** | 5.64 | -92.64 | 72.43 | 2.78 |
| medium | 2 | **90.06** | 4.98 | -85.08 | 79.97 | -0.79 |
| expert | 0 | 3.49 | **144.92** | +141.43 | 66.49 | 32.06 |
| expert | 1 | 5.91 | **154.08** | +148.17 | 91.27 | 39.91 |
| expert | 2 | 4.26 | **11.90** | +7.64 | 93.04 | 40.86 |

**Walker2d**

| Tier | Seed | Base return | Gate return | Δ return | Base MEG | Gate MEG |
|---|---|---|---|---|---|---|
| simple | 0 | -4.57 | **0.16** | +4.74 | 197.14 | -0.77 |
| simple | 1 | -4.11 | **-1.52** | +2.60 | 166.33 | -7.61 |
| simple | 2 | -3.86 | **-1.73** | +2.13 | 156.84 | -1.71 |
| medium | 0 | -8.03 | **-0.72** | +7.31 | 91.47 | 75.35 |
| medium | 1 | **53.22** | -2.91 | -56.13 | 55.98 | 33.78 |
| medium | 2 | -7.02 | **1.27** | +8.29 | 50.61 | 59.52 |
| expert | 0 | -6.94 | **-6.25** | +0.70 | 148.93 | 69.44 |
| expert | 1 | -5.09 | **-2.63** | +2.46 | 165.71 | 121.98 |
| expert | 2 | -7.68 | **-6.46** | +1.22 | 146.76 | 111.30 |

### B.4 COMMON-ERROR DIAGNOSTIC

Table 5 diagnoses frozen-model predictions rather than pairwise training. Each environment uses twenty models (five seeds at each of four mixtures) and sixteen shared source episodes from behavior and planner policies at $p_s \in \{0.05, 0.5\}$. Four anchors per episode and four fixed continuation suffixes yield $20 \times 16 \times 4 \times 4 = 5{,}120$ dependent cases. Each compares eight first actions at horizon 40, using true first-step rewards, a known time limit, and no state projection. The action-averaged first-state prediction error is removed before rescoring the same candidates with the frozen learned continuation. A fixed external scale is pooled from the four training mixtures. Common energy divides summed common-error energy by summed total error energy; choice frequencies are descriptive paired-case summaries.

## C TEMPORAL PLACEMENT OF EXACT STATE CORRECTIONS

To examine where accurate states affect candidate selection, we substitute true states at selected depths of H40 rollouts while freezing candidate actions, model parameters, and the complete learned reward function. The primary arms share recorded true termination and complete simulator snapshots validated by canonical replay. The budget counts corrected depths, independently of simulator calls or training updates. Each group contains eight fixed candidates. For $k$ corrections, uniform uses depths $\lfloor 40j/(k+1) \rfloor$, $j = 1, \ldots, k$; early and late use the first and last $k$ internal depths, respectively. Random schedules sample $k$ distinct depths from $\{1, \ldots, 39\}$. Figure 5 compares early, uniform, late, and random placement at 2, 4, 8, and 16 corrections, averaging three target-source means. Random placement first averages twenty prescribed schedules within each candidate group. All 384 valid groups are retained, including 115 whose free-rollout selection is already optimal. Candidate banks, budgets, and schedules are repeated measurements rather than additional independent sources.

In HC, uniform has the lowest group-mean loss at every budget under both mixtures. Four uniform corrections give losses of 3.813 and 2.193, below sixteen early corrections at 4.884 and 4.110. CP also has a four-versus-sixteen benefit, but uniform leads only five of eight mixture–budget cells; at sixteen corrections, late and random placement are better under both mixtures. Increasing the correction budget therefore need not improve selection. These exact-state comparisons motivate collecting along planned paths; learned repair benefits are evaluated separately below.

## D ACQUISITION PROTOCOL AND RESULTS

### D.1 ACQUISITION AND TRAINING PROTOCOL

From a 300-step frozen-model navigation episode, all collectors use six anchors at steps 0, 40, 80, 120, 160, and 200. At each anchor, the frozen model proposes an H40 action path. Six path collectors select five depths for additional H8 branches: uniform uses depths 3, 11, 19, 27, and 35; stratified disagreement takes the largest score in each of five depth bins; global disagreement takes the five largest scores without bins; random takes five distinct depths; early and late use depths 0–4 and 35–39. Disagreement is computed from normalized differences between frozen ensemble members' predicted means. Global disagreement resolves ties deterministically, and random placement uses a fixed strategy seed. Paths, scores, and locations are fixed before querying true path labels.

Each selected state starts two H8 branches: one with a constant random action and one with independently sampled actions over time. Path collectors share branch-generation rules and paired seeds, assigning branch pairs in temporal order. They retain all forty true path transitions, including late-path labels in the early condition. Local instead starts fifteen H8 branches at the anchor, sharing its first ten action sequences with the path collectors. Navigation is charged to each condition even when reused computationally, giving

$$B = 300 + 6(40 + 10 \times 8) = 300 + 6(15 \times 8) = 1{,}020 \tag{21}$$

true transitions per collector. Every paid transition enters a flat repair pool with equal sampling multiplicity; overlapping sequence windows are not extracted. Equal transition counts do not imply equal collection or training time.

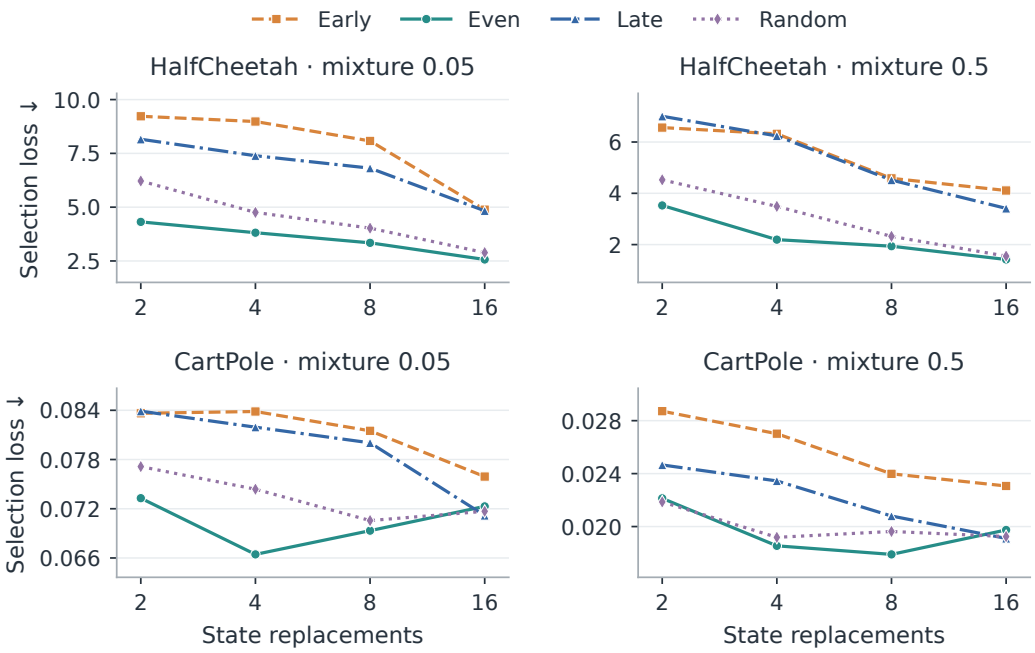



Figure 5: **The placement of exact corrections matters.** Selection loss (↓) at four H40 correction budgets, separately by environment and mixture. Points average three target-source means, with twenty random schedules averaged within a candidate group. Environment scales differ. The diagnostic measures candidate selection after exact state substitution.

Dynamics repair uses U, the single-step increment-normalized squared-error loss, for 500 updates per ensemble member with batch size 128, Adam learning rate $10^{-4}$, and gradient-norm clipping at ten. Old and new pools each supply half a batch. The complete learned reward function and all normalization statistics remain frozen, and deployment uses the fixed planner with learned predictions. Frozen retains the original model; old-only matches the training updates using existing data without new interaction. Common-label training removes the 480 branch labels while retaining 300 navigation and 240 full-path labels from the already paid package. This control tests branch labels; its 540 retained transitions do not establish a 540-interaction acquisition method.

Each model–condition return first averages three paired 1,000-step resets, then enters source-level aggregation. Return SD is across independent source means, with three sources per mixture in development and five per task–mixture cell in confirmation. IID NMSE uses 50,000 independent held-out transitions per model and fixed base-model state variances. Repeated resets are not additional independent sources.

### D.2 Development comparisons

The HC development study uses three independently collected datasets and trained models per mixture. Table 8 includes all seven collectors, both training controls, and the frozen reference. Uniform exceeds old-only in all six models but exceeds common-label training in only three; its mixture-mean differences from common-label training are $+98.28$ and $-151.82$ at $p = 0.05$ and $0.5$. Navigation and traversal labels already support useful repair, while additional branch labels show no consistent advantage. Uniform has the highest low-mixture mean and late collection the highest high-mixture mean. Stratified disagreement failed its prespecified expansion criterion, and the four-condition confirmation below was a separately specified study on new sources.

Table 8: **HC acquisition development with matched training controls.** Return is mean ± sample SD across three source means, each averaging three paired resets. Each collector costs 1,020 true transitions; old-only adds none, and common-label training uses part of an already paid package. $N$: navigation; $P$: full paths; $B_\ell/B_p$: local/path branches. Bold marks the highest displayed mean.

| Condition | New labels used | $p = 0.05$: return ↑ | $p = 0.5$: return ↑ |
|---|---|---|---|
| Frozen | None | $-403.57 \pm 192.21$ | $86.41 \pm 799.27$ |
| Old data only | None | $-385.38 \pm 64.94$ | $-238.16 \pm 434.25$ |
| Common labels only | $N + P$ | $23.66 \pm 203.62$ | $1237.81 \pm 1821.48$ |
| Broad local | $N + B_\ell$ | $-201.28 \pm 98.10$ | $317.37 \pm 298.22$ |
| Uniform midpoints | $N + P + B_p$ | $\mathbf{121.94} \pm 434.09$ | $1085.99 \pm 1039.67$ |
| Stratified disagreement | $N + P + B_p$ | $-253.40 \pm 226.05$ | $1069.34 \pm 1480.09$ |
| Global disagreement | $N + P + B_p$ | $-233.79 \pm 54.83$ | $203.07 \pm 306.00$ |
| Random | $N + P + B_p$ | $-114.42 \pm 158.24$ | $788.73 \pm 634.72$ |
| Early branches | $N + P + B_p$ | $-199.68 \pm 244.12$ | $1043.67 \pm 718.80$ |
| Late branches | $N + P + B_p$ | $70.26 \pm 151.71$ | $\mathbf{1404.77} \pm 2174.37$ |

### D.3 Confirmation on independent sources

Confirmation independently generates five new datasets and trained models for each HC/CP and $p \in \{0.05, 0.5\}$ cell. Frozen, old-only, local, and uniform each receive three paired resets, totaling 240 full deployments under the protocol above. Table 9 reports all four conditions. Table 10 retains the eight primary contrasts, uniform-minus-old-only and uniform-minus-local, together with supporting local-minus-old-only comparisons in every cell. Each unadjusted 95% paired $t$ interval uses five source differences and four degrees of freedom; no multiplicity correction is applied. Source indices match conditions within a cell and do not create twenty replications of a common effect across tasks and mixtures.

Both acquisition strategies improve HC mean return over frozen and old-only at both mixtures. Local exceeds old-only in all ten models, supporting gains beyond additional training updates. Uniform has the highest mixture means, but both uniform-minus-local intervals cross zero; its high-mixture interval against old-only also crosses zero. The evidence therefore supports HC acquisition gains with uncertain incremental benefits from distributing branches. In CP, local leads at $p = 0.05$ and old-only at $p = 0.5$; all reported acquisition contrasts in Table 10 cross zero. Old-only has lower IID NMSE than both collectors in all twenty model conditions, showing that prediction on the original distribution does not rank their deployment returns.

Table 9: Independent-source confirmation: complete returns (↑) and IID NMSE (↓), mean ± source SD, $n = 5$ per cell. Each return source mean averages three paired 1,000-step resets.

| Env. | $p$ | Condition | Return ↑ | IID NMSE ↓ |
|---|---|---|---|---|
| HC | 0.05 | Frozen | $-406.57 \pm 69.36$ | $0.021333 \pm 0.000288$ |
| HC | 0.05 | Old data only | $-483.20 \pm 92.57$ | $0.015021 \pm 0.000212$ |
| HC | 0.05 | Broad local | $-208.12 \pm 121.81$ | $0.017733 \pm 0.000663$ |
| HC | 0.05 | Uniform midpoints | $-59.01 \pm 180.46$ | $0.017539 \pm 0.000517$ |
| HC | 0.5 | Frozen | $601.85 \pm 808.31$ | $0.024067 \pm 0.001181$ |
| HC | 0.5 | Old data only | $603.62 \pm 1351.44$ | $0.015443 \pm 0.000418$ |
| HC | 0.5 | Broad local | $1856.33 \pm 932.10$ | $0.019167 \pm 0.001067$ |
| HC | 0.5 | Uniform midpoints | $2403.63 \pm 964.78$ | $0.018792 \pm 0.000989$ |
| CP | 0.05 | Frozen | $121.34 \pm 79.77$ | $0.000372 \pm 0.000014$ |
| CP | 0.05 | Old data only | $130.95 \pm 68.96$ | $0.000141 \pm 0.000004$ |
| CP | 0.05 | Broad local | $137.26 \pm 70.41$ | $0.000223 \pm 0.000032$ |
| CP | 0.05 | Uniform midpoints | $116.24 \pm 68.74$ | $0.000245 \pm 0.000101$ |
| CP | 0.5 | Frozen | $221.62 \pm 141.32$ | $0.000520 \pm 0.000104$ |
| CP | 0.5 | Old data only | $377.08 \pm 155.31$ | $0.000220 \pm 0.000027$ |
| CP | 0.5 | Broad local | $297.80 \pm 43.90$ | $0.000307 \pm 0.000023$ |
| CP | 0.5 | Uniform midpoints | $256.77 \pm 124.67$ | $0.000374 \pm 0.000078$ |

Table 10: Primary and supporting paired acquisition contrasts. Uniform and Local denote Uniform midpoints and Broad local. Positive differences favor the first condition. Intervals are unadjusted 95% paired $t$ intervals; positive counts are out of five matched sources.

| Env. | $p$ | Contrast | $\Delta$ return | 95% paired $t$ interval | Positive |
|---|---|---|---|---|---|
| HC | 0.05 | Uniform − Old-only | +424.19 | [153.02, 695.35] | 5/5 |
| HC | 0.05 | Uniform − Local | +149.11 | [-3.23, 301.45] | 4/5 |
| HC | 0.05 | Local − Old-only | +275.08 | [43.48, 506.68] | 5/5 |
| HC | 0.5 | Uniform − Old-only | +1800.00 | [-58.04, 3658.05] | 4/5 |
| HC | 0.5 | Uniform − Local | +547.30 | [-558.58, 1653.18] | 3/5 |
| HC | 0.5 | Local − Old-only | +1252.71 | [363.79, 2141.62] | 5/5 |
| CP | 0.05 | Uniform − Old-only | -14.71 | [-61.99, 32.57] | 1/5 |
| CP | 0.05 | Uniform − Local | -21.02 | [-47.64, 5.60] | 1/5 |
| CP | 0.05 | Local − Old-only | +6.31 | [-65.36, 77.99] | 3/5 |
| CP | 0.5 | Uniform − Old-only | -120.31 | [-295.91, 55.30] | 1/5 |
| CP | 0.5 | Uniform − Local | -41.03 | [-180.05, 97.99] | 2/5 |
| CP | 0.5 | Local − Old-only | -79.27 | [-251.57, 93.03] | 2/5 |