Mood Swings: Calibrated Reward–Cost Control in a Frozen PPO Policy

Dario Fumarola Jin Tan Ruan

Amazon Web Services
New York, New York

Abstract

Deploying a preference-conditioned policy requires a way to relate its inputs to task outcomes. We study this problem with a frozen PPO actor controlled by a relative cost weight and a sampling temperature. Validation rollouts map requested mean costs to settings, while a reference-risk alarm governs entry into and release from caution. In Pac-Mind++ and MiniHack-HazardRooms, the actor retains 98.0% and 97.2% of an eight-actor library's mean reward–cost coverage with one-eighth of its policy-training interactions. From common alarm states, raising the cost weight reduces 100-action failure from 12.6% to 5.0% and from 18.4% to 7.9%; entropy matching reproduces approximately half of these reductions. With the actor held fixed, risk-based switching improves on hazard-blind schedules and visible-hazard alarms, and releasing caution recovers completion and reward. Calibrated critic thresholds perform similarly to separately tuned raw thresholds. A controlled change in hazard behavior degrades cost selection and alarm reliability, both of which improve after recalibration with the network frozen. CCPO is a competitive single-policy alternative throughout these comparisons.

1 Introduction

A navigation policy that takes a short route through an empty corridor may need a detour when a hazard appears. Preference conditioning offers a way to make that change after training, although choosing the input remains a practical problem: a cost weight of 0.7 has no direct meaning in raw cost units. Low cost can also be misleading when the policy has stopped making progress.

We study this deployment problem with Mood-Conditioned PPO (MC-PPO), which exposes a cost weight through feature conditioning and a temperature through final-logit scaling. Once the network is frozen, we use validation rollouts to select settings for mean-cost requests and fit an alarm that controls entry into and release from caution.

The study separates the behaviors available in a conditioned actor from the rules used to select them. We first evaluate cost selection, paired interventions, and repeated switching, then hold the actor fixed while comparing supervisors and recalibrating under changed hazards. Applying the same procedure to CCPO and interpolated specialist libraries lets us examine how these deployment choices interact with the underlying policy.

2 Related Work

Conditioning a policy or value function on a goal, return, or preference is well established. Universal value function approximators share value estimates across goals (Schaul et al., 2015), and successor features support changes in reward specification (Barreto et al., 2017). Within multi-objective reinforcement learning (Hayes et al., 2022), Pareto Conditioned Networks select behavior through desired returns (Reymond et al., 2022). Our implementation uses a scalarization weight as its policy input and trains with on-policy PPO (Schulman et al., 2017).

Constraint-Conditioned Policy Optimization (CCPO) is a particularly relevant comparison because it learns policies for different cost thresholds without retraining (Yao et al., 2023). CPO optimizes against a fixed constraint (Achiam et al., 2017), whereas CCPO exposes the threshold as a deployment control. We use a categorical adaptation and align its threshold input with our preference input through achieved reward and cost, giving both methods validation-based setting selection and calibrated alarms.

FiLM applies condition-dependent affine transformations to hidden features (Perez et al., 2018). In our actor, it conditions features on the cost weight, while temperature acts at the final logits. This arrangement preserves action rankings and gives a fixed-input entropy ordering. Specialists whose logits are mixed before temperature scaling have the same properties, so we compare them with MC-PPO on training allocation, behavioral response, and calibration.

Post-hoc calibration maps prediction scores to outcome frequencies (Niculescu-Mizil and Caruana, 2005); here we use it to estimate the failure risk of a specified reference policy from a cost-critic score. Distribution shift can degrade such estimates (Ovadia et al., 2019). Since switching also changes which states the actor visits, our calibration checks cover cautious and post-release states as well as seeking trajectories.

3 Policy and Calibration

3.1 Controls and training objective

The environment has a finite action set, task reward r_t, non-negative hazard cost c_t, and an episode horizon H. The actor observes x_t, comprising four egocentric frames and task-specific scalar features, and receives the control input $u = (w, \tau)$, with $w \in [0.1, 0.8]$ and $\tau \in [0, 1]$. Since only

the ratio of the positive reward and cost gains matters, we parameterize it as

$$w = \frac{\beta_C}{\beta_R + \beta_C}. \tag{1}$$

Let $\tilde{r}_t = r_t/S_R$ and $\tilde{c}_t = c_t/S_C$, using task scales fixed before evaluation. For a constant setting, the training objective is

$$J(\theta; u) = \mathbb{E}_{\pi_\theta(\cdot|x,u)} \sum_{t=0}^{H-1} \gamma^t \left[(1-w)\tilde{r}_t - w\tilde{c}_t\right]. \tag{2}$$

We train with discount $\gamma = 0.99$ and evaluate total episode reward, total episode cost, discounted cost, and terminal failure separately. Recording both cost measures keeps late hazards visible even when discounting gives them little weight in the training objective.

3.2 Actor and optimization

An observation encoder feeds two FiLM-conditioned actor layers whose feature scales and shifts depend on w. The resulting logits $z_\theta(x, w)$ define the action distribution through

$$T(\tau) = 0.7(1.8/0.7)^\tau, \tag{3}$$

$$\pi_\theta(a \mid x, w, \tau) = \mathrm{softmax}(z_\theta(x, w)/T(\tau))_a\,. \tag{4}$$

The output gate is the sole entry point for temperature. At a fixed observation and weight, increasing T preserves every action ranking and gives nondecreasing entropy, as shown in Appendix A.

The reward and cost critics share the observation encoder and use a separate critic trunk conditioned on (w, τ), since both controls can affect future returns. We apply generalized advantage estimation (Schulman et al., 2016) to the scaled reward and cost streams separately, then combine them for the actor:

$$\widehat{A}_t^w = (1-w)\widehat{A}_t^R - w\widehat{A}_t^C. \tag{5}$$

We normalize this combined advantage once per rollout and use it in PPO's clipped surrogate. The two value losses remain separate.

We allocate one worker to each point of an 8×8 weight–temperature grid, with 128 transitions per worker in every update. This keeps the training allocation balanced across settings despite differences in episode length. A fixed entropy bonus of 0.01, independent of the temperature input, and a small adjacent-weight smoothness penalty regularize the update. Appendix B gives the losses, network sizes, and optimization settings.

3.3 Choosing a setting after training

For each candidate setting u, validation rollouts estimate mean episode reward $\widehat{R}_{\mathrm{val}}(u)$ and an upper estimate $U_C(u)$ of mean episode cost. A request b is translated into

$$\widehat{u}(b) \in \arg\max_{u:\, U_C(u) \le b} \widehat{R}_{\mathrm{val}}(u). \tag{6}$$

We mark a request unsupported when no setting passes the screen and evaluate each selected setting on held-out layouts. Because the screen uses measured episode cost, it accommodates nonmonotone relationships between cost and weight.

For online changes, we evaluate the cost critic at a fixed reward-seeking reference setting on every decision. Isotonic regression fitted to validation trajectories maps its score to an estimate of failure probability within 100 actions under continued seeking. Holding this reference input fixed throughout the episode keeps the score comparable as the acting policy changes its control setting.

The default supervisor latches at the first alarm above 0.10 and ramps w from 0.2 to 0.7 over ten actions, keeping $\tau = 2/7$. A reversible variant adds release and re-entry using the same frozen actor, critics, and calibration map. We evaluate these repeated control changes after training at constant settings.

4 Evaluation

4.1 Tasks, splits, and baselines

Pac-Mind++ is a partially observed 20×20 maze with pellets, power pellets, and four moving ghosts. Its five-action policy has 3,000 actions to clear the maze; collision with an active ghost terminates the episode at unit cost, and proximity adds a smaller nonterminal cost. MiniHack-HazardRooms is a custom task built with MiniHack and the NetHack Learning Environment (Samvelyan et al., 2021; Küttler et al., 2020), containing rooms, doors, keys, lava, spikes, monsters, and an amulet goal. Its wrapper exposes nine actions with a deadline of 1,000 decisions. Fatal transitions cost one and nonfatal spike entries cost 0.2. Appendix C describes the rewards, observations, and map generators.

Training, validation, and test use 128/32/64 layouts in Pac-Mind++ and 64/16/32 in HazardRooms. The per-run training budget B is 50 million and 100 million decisions, respectively, with 12 independently trained replicates per method evaluated at their final checkpoints. All methods share the task interfaces, layout splits, reward scales $S_R = 200$ and 100, and cost scale $S_C = 1$.

Our main baselines are parameter-matched Concat-PPO, categorical CCPO, and eight weight-specialist actors. Concat-PPO replaces FiLM with concatenation at both actor layers while keeping the rest of the training procedure fixed. CCPO retains threshold-conditioned value factorization and variational policy updates; we evaluate action expectations by exact categorical sums. Its discounted-cost threshold is mapped to episode outcomes through validation, using the same procedure as for the weight-based methods.

Each specialist trains at one weight and all eight temperatures. We obtain continuous weight control by combining neighboring specialists, either through action-probability mixing or through logit mixing before a common temperature gate; the latter also preserves the fixed-input temperature property. The matched-budget library receives $B/8$ interactions per actor, and the full-budget library receives B per actor. A complete library counts as one training replicate. Here B counts policy-training interactions, with calibration rollouts, evaluation, and optimizer

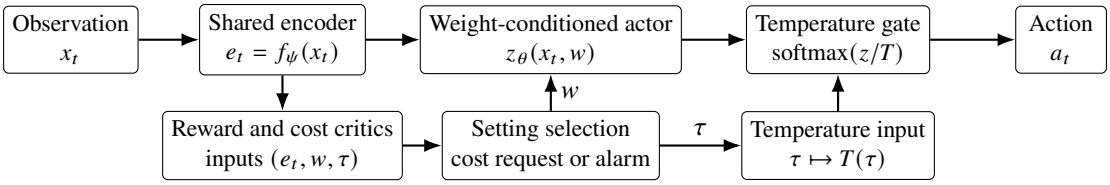


Figure 1: Runtime control around a frozen actor. Validation maps requested mean costs to settings, and the online supervisor uses a calibrated critic score to change the weight. Temperature acts at the output gate.

work accounted for separately in Appendix D.6. Additional fixed-policy and 64-actor references appear in Appendix E.1.

4.2 Static calibration and behavior

We evaluate 64 grid settings and 49 cell centers, using 256 test episodes per setting and training seed, balanced across layouts. The centers are held out from training trajectories and regularization, and CCPO receives the same number of evaluated thresholds. Reward–cost hypervolume uses a fixed raw reference of zero reward and cost one, with reward divided by S_R. We compute it for each trained seed before averaging.

For the selection rule in (6), we request mean episode costs $b \in \{0.10, 0.15, 0.20, 0.35, 0.50\}$ and record reward, completion, and achieved cost at each selected setting. We also use 64 fresh episodes per test layout to estimate the fraction of layout means exceeding the request, allowing aggregate performance to be read alongside layout-level variation.

Fixed-input diagnostics use a common buffer of 20,000 observation stacks per seed, drawn equally from neutral, seeking, cautious, and random rollouts. We compare action rankings at $w = 0.2$ and $w = 0.7$. A junction subset, defined from visible geometry before examining policy outputs, requires three passable cardinal neighbors and a visible active hazard.

4.3 Alarm continuations and full episodes

We save 256 alarm states per seed from seeking MC-PPO rollouts, balanced across test layouts, with alarms enabled after 25 elapsed actions. Each snapshot includes the simulator state, observation stack, elapsed time, and random-number state. Four continuation seeds per snapshot are shared across interventions, yielding 12,288 continuations per domain and branch. We measure the first 100 actions from the original alarm, then follow every branch to completion, failure, or its original deadline.

To test the contribution of confidence changes, an entropy-matched branch keeps the seeking logits and adjusts their temperature. At each visited state, it computes the cautious ramp's entropy and uses bisection over $[0.05, 20]$ to match it as closely as possible. This preserves the seeking action rankings while allowing a wider temperature range than the public interface.

On validation snapshots, we select competitor response settings that minimize 100-action failure while retaining at least 85% of seeking reward and preserving eventual completion. At the common test snapshots, each competitor supplies a hold branch and a response branch; both replace the generating actor, so their paired difference isolates the control change. MC-PPO uses the fixed pair $(0.2, 2/7)$ and $(0.7, 2/7)$, and all ramps last ten actions.

The reset-based study uses 1,024 paired episodes per seed for each complete policy–supervisor system, including episodes without alarms. Each method begins at its own seeking setting and uses its own calibrated alarm. Alongside hold and latched branches, we test a reversible branch that releases caution after the reference alarm probability stays below 0.04 for five decisions. Modes last at least 20 decisions, transitions use ten-action ramps, and re-entry is allowed throughout the episode. We fix these shared thresholds and durations before testing.

4.4 Supervisor controls and hazard shift

We compare supervisors while holding the MC-PPO actor, its two operating points, and the ramp and dwell rules fixed. Controls include hazard-blind schedules, visible-hazard distance, raw critic thresholds tuned per seed, and an offline simulator oracle. The oracle uses a copy of the simulator and future random streams to check whether continued seeking would fail within 100 actions. All branches share reset seeds; Appendix D.3 specifies threshold selection and the timing control.

A separate audit samples states visited during caution and the first 20 decisions after release, then relabels them with independent continued-seeking rollouts to preserve the alarm's reference outcome. We also increase Pac-Mind++ ghost pursuit from 75% to 90% on the same layouts. We first test the existing calibration and then refit it on shifted validation rollouts, keeping actor and critic weights frozen throughout.

4.5 Uncertainty

Tables average 12 trained replicates. Unless noted otherwise, brackets are 95% percentile intervals from 5,000 bootstrap resamples of training seeds and test layouts as crossed clusters, with identical resampling indices for paired comparisons. Continuations within a seed–layout cell stay together, and validation-selected settings remain fixed during test resampling. The trained seed is the independent model replicate (Agarwal et al., 2021).

Table 1: Complete episodes from reset: hold → latched response. Each method uses its own seeking policy and calibrated alarm. The final column gives the paired change in failure, in percentage points, with a 95% interval. Completion and failure are percentages; the remaining episodes time out.

Method	Budget	Completion	Failure	Episode reward	Failure change
Pac-Mind++					
MC-PPO	B	60.5 → 62.1	8.1 → 4.7	173.0 → 164.0	−3.4 [−4.5, −2.3]
Concat-PPO	B	55.2 → 56.4	10.4 → 6.6	163.0 → 155.0	−3.8 [−5.2, −2.4]
CCPO	B	61.4 → 63.5	7.9 → 4.6	165.0 → 160.0	−3.3 [−4.5, −2.1]
Eight actors, logit mixture	B	56.8 → 58.4	9.5 → 5.9	166.0 → 158.0	−3.6 [−4.8, −2.4]
Eight actors, logit mixture	$8B$	63.4 → 65.5	7.2 → 4.0	177.0 → 170.0	−3.2 [−4.3, −2.1]
MiniHack-HazardRooms					
MC-PPO	B	71.9 → 77.3	20.6 → 13.8	78.0 → 82.0	−6.8 [−8.8, −4.8]
Concat-PPO	B	65.4 → 69.8	24.9 → 18.7	70.8 → 73.9	−6.2 [−8.5, −4.0]
CCPO	B	73.1 → 79.0	19.1 → 12.7	77.3 → 82.8	−6.4 [−8.4, −4.4]
Eight actors, logit mixture	B	68.0 → 72.6	23.0 → 16.7	73.6 → 77.0	−6.3 [−8.5, −4.1]
Eight actors, logit mixture	$8B$	75.4 → 81.1	17.3 → 11.5	80.9 → 86.1	−5.8 [−7.7, −3.9]

5 Results

5.1 Latched caution reduces failure from reset

We begin with complete episodes, including the alarm and any time spent seeking before it fires. Enabling the latched MC-PPO response reduces failure from 8.1% to 4.7% in Pac-Mind++ and from 20.6% to 13.8% in HazardRooms (Table 1). Completion in HazardRooms rises by 5.4 percentage points, with a 95% interval of $[3.1, 7.7]$, while the Pac-Mind++ change is a smaller 1.6 $[-0.2, 3.4]$ points. The reward effects differ across tasks: pellet-collection reward falls, whereas more HazardRooms episodes reach the amulet and reward rises.

CCPO achieves similar intervention gains, and the full-budget specialist library has the highest mean completion and lowest mean failure among the tested systems. That library uses eight actors and budget $8B$. At the matched total budget B, MC-PPO's shared actor performs better than the library trained on data divided among eight specialists.

In ordinary MC-PPO seeking episodes, the alarm fires in 24.8% of Pac-Mind++ episodes and 37.6% of HazardRooms episodes. This limited coverage helps explain the smaller reset-based reductions relative to continuations from selected hazardous states. Failures with no prior alarm account for 1.3% and 4.9% of all episodes; among failures, 71.6% and 65.0% have an alarm at least ten actions in advance. Appendix D.2 gives the reliability and denominator measurements.

5.2 Higher cost weights more than halve short-term failure

We next replay common alarm states so that interventions begin at the same place and time (Table 2). Raising the MC-PPO cost weight reduces 100-action failure from 12.6% to 5.0% in Pac-Mind++ and from 18.4% to 7.9% in HazardRooms, while reward over the same window falls from 31.7 to 28.6 and from 20.8 to 18.2. CCPO and the full-budget mixtures give comparable responses. The paired MC-PPO minus CCPO failure differences are -0.2 $[-1.1, 0.7]$ and 0.4 $[-0.9, 1.7]$ percentage points.

Following the MC-PPO branches to their original deadlines, we find that eventual failure falls from 27.4% to 13.6% in Pac-Mind++ and from 41.8% to 23.6% in HazardRooms. Completion also improves, from 56.8% to 63.4% and from 48.5% to 62.9%, respectively.

The high-weight branch in Appendix E shows the effect of stronger caution. Raising w to 0.8 lowers short-window failure further, but completion drops to 49.2% and 48.6%, with timeout rates of 40.6% and 32.6%. Inactive decisions increase to 27.4% and 34.0%.

A delay weakens the response. Counting failures from the original alarm, postponing the ten-action ramp by 40 actions raises 100-action failure to 11.7% in Pac-Mind++ and 17.3% in HazardRooms, close to continued seeking.

5.3 Releasing caution recovers completion and reward

We then let the supervisor return to seeking after a sustained low alarm score, with further entries into caution allowed. Relative to latching, this increases MC-PPO completion in Pac-Mind++ from 62.1% to 64.6% and reward from 164.0 to 171.2. Failure rises from 4.7% to 5.0%, compared with 8.1% for continued seeking. In HazardRooms, completion rises from 77.3% to 80.0% and failure from 13.8% to 14.4%.

Table 3 gives the paired changes relative to the latched supervisor. CCPO and the specialist libraries show a similar recovery in completion and reward alongside a small increase in failure. Repeated entries into caution occur in 8.9% of all MC-PPO Pac-Mind++ episodes and 13.7% of HazardRooms episodes.

5.4 Risk-based timing improves on matched schedules

Holding the actor fixed, we compare risk-based switching with hazard-blind schedules matched on validation caution time

Table 2: Responses from common alarm states. All entries are percentages with 95% intervals. Hold and response branches apply the named method to the same saved MC-PPO states. Completion and eventual failure use the original task deadline; every ramp lasts ten actions.

		100-action failure (%)		Response to task deadline (%)	
Method	Budget	Hold	Response	Completion	Failure
Pac-Mind++					
MC-PPO	B	12.6 [11.4, 13.9]	5.0 [4.3, 5.8]	63.4 [60.0, 66.6]	13.6 [11.9, 15.4]
Concat-PPO	B	13.8 [12.4, 15.3]	7.3 [6.3, 8.4]	56.2 [52.8, 59.6]	18.9 [16.8, 21.1]
CCPO, categorical	B	12.4 [11.1, 13.7]	5.2 [4.4, 6.1]	63.8 [60.4, 67.1]	13.4 [11.7, 15.3]
Eight actors, logit mixture	B	13.2 [11.9, 14.6]	5.9 [5.0, 6.9]	59.3 [55.9, 62.7]	16.2 [14.3, 18.3]
Eight actors, logit mixture	$8B$	12.2 [11.0, 13.5]	4.5 [3.8, 5.3]	65.5 [62.2, 68.7]	12.7 [11.0, 14.5]
MiniHack-HazardRooms					
MC-PPO	B	18.4 [16.7, 20.1]	7.9 [6.7, 9.2]	62.9 [59.4, 66.3]	23.6 [21.2, 26.1]
Concat-PPO	B	21.0 [19.1, 23.0]	11.8 [10.3, 13.4]	53.8 [50.1, 57.4]	31.6 [28.8, 34.5]
CCPO, categorical	B	18.2 [16.6, 20.0]	7.5 [6.3, 8.8]	64.0 [60.5, 67.4]	22.7 [20.4, 25.2]
Eight actors, logit mixture	B	19.9 [18.1, 21.8]	9.3 [8.0, 10.8]	59.0 [55.4, 62.5]	26.9 [24.4, 29.6]
Eight actors, logit mixture	$8B$	17.8 [16.2, 19.6]	6.8 [5.7, 8.1]	66.1 [62.7, 69.4]	21.4 [19.1, 23.8]

Table 3: Reversible supervision minus the latched response, on paired episodes from reset. Completion and failure changes are percentage points. Brackets are 95% intervals. The final column is the percentage of all episodes with at least two entries into caution.

Method	Budget	Completion change	Failure change	Reward change	Re-entry (%)
Pac-Mind++					
MC-PPO	B	2.5 [0.8, 4.2]	0.3 [−0.2, 0.8]	7.2 [4.4, 10.0]	8.9
CCPO	B	2.4 [0.7, 4.2]	0.3 [−0.2, 0.8]	6.8 [4.0, 9.6]	8.5
Eight actors, logit mixture	B	2.3 [0.4, 4.2]	0.5 [−0.1, 1.1]	7.0 [3.9, 10.0]	10.1
Eight actors, logit mixture	$8B$	2.3 [0.7, 4.0]	0.3 [−0.2, 0.8]	6.5 [3.9, 9.1]	8.0
MiniHack-HazardRooms					
MC-PPO	B	2.7 [0.9, 4.5]	0.6 [−0.2, 1.4]	2.1 [0.4, 3.8]	13.7
CCPO	B	2.2 [0.5, 4.0]	0.6 [−0.2, 1.4]	1.8 [0.2, 3.4]	12.9
Eight actors, logit mixture	B	2.4 [0.4, 4.4]	0.9 [0.0, 1.8]	1.8 [−0.1, 3.7]	15.1
Eight actors, logit mixture	$8B$	1.9 [0.3, 3.6]	0.6 [−0.1, 1.3]	1.7 [0.2, 3.2]	12.3

(Table 4). The schedules spend nearly the same fraction of test decisions in caution, yet failure remains higher by 2.2 [1.3, 3.1] percentage points in Pac-Mind++ and 4.3 [2.6, 6.0] in HazardRooms. Visible-hazard alarms narrow that gap. The simulator oracle, which has access to hidden state and future randomness, achieves the lowest failure rates.

Calibrated thresholds perform similarly to raw critic cutoffs tuned separately for each seed. Because the calibration map is monotone, its trigger and release boundaries can be implemented directly in score units (Appendix D.3). We use the probability scale to specify shared risk thresholds and to check reference-failure predictions.

Those predictions are systematically low in the visited-state audit. During caution, mean predicted and observed reference failure are 10.7% and 12.3% in Pac-Mind++, and 14.9% and 17.4% in HazardRooms. The gap persists after release and grows under changed hazards (Appendix D.4). Appendix Table 8 reports the completion and failure changes obtained by moving the trigger and release thresholds around the default.

5.5 Aggregate cost targets hold under the original hazards

Every trained seed has a qualifying setting at each of the five requested budgets (Figure 2a and Table 5), and test mean cost remains below the request in every aggregate comparison. At $b = 0.20$, MC-PPO reaches costs of 0.168 [0.158, 0.178] in Pac-Mind++ and 0.167 [0.157, 0.177] in HazardRooms. The gap between requested and achieved cost reflects both the finite candidate set and screening with an upper cost estimate.

Tighter requests also reduce completion: lowering b from 0.50 to 0.10 takes Pac-Mind++ from 60.2% to 39.6% and HazardRooms from 70.2% to 47.6%. We report these outcomes alongside cost because reduced progress contributes to the low-cost operating points. Layout-level variation remains substantial. At $b = 0.20$, estimated layout mean cost exceeds the target in 18.8% and 19.8% of seed–layout cells despite the lower aggregate means.

CCPO is competitive at the same requested budget. At $b = 0.20$, it attains reward 141.3 at cost 0.182 in Pac-Mind++ and reward 67.0 at cost 0.181 in HazardRooms, compared with MC-PPO's 142.4 at 0.168 and 64.8 at 0.167. The full specialist comparison at this request appears in Appendix E.1.

Table 4: Supervisor comparison with the same frozen MC-PPO actor and operating points. All switching rules are reversible and share ramps and dwell times. Failure includes a 95% interval; other entries are means. Caution time is the mean within-episode fraction of decisions in cautious mode, including its entry ramp. Hazard-blind schedules are matched using validation data.

Supervisor	Completion (%)	Failure (%)	Episode reward	Caution time (%)
Pac-Mind++				
No switching	60.5	8.1 [7.0, 9.3]	173.0	0.0
Hazard-blind schedule	61.5	7.2 [6.2, 8.3]	168.1	12.2
Visible-hazard alarm	63.1	6.2 [5.3, 7.2]	169.8	13.1
Critic thresholds, tuned per seed	64.4	5.1 [4.3, 6.0]	170.6	12.8
Calibrated critic	64.6	5.0 [4.2, 5.8]	171.2	12.4
Simulator oracle	66.1	3.6 [2.9, 4.4]	173.4	9.1
MiniHack-HazardRooms				
No switching	71.9	20.6 [18.7, 22.6]	78.0	0.0
Hazard-blind schedule	73.3	18.7 [16.9, 20.6]	78.8	21.0
Visible-hazard alarm	77.8	16.5 [14.8, 18.3]	82.2	22.6
Critic thresholds, tuned per seed	79.8	14.6 [12.8, 16.5]	83.9	21.7
Calibrated critic	80.0	14.4 [12.6, 16.3]	84.1	21.1
Simulator oracle	82.7	11.1 [9.6, 12.8]	86.9	18.7

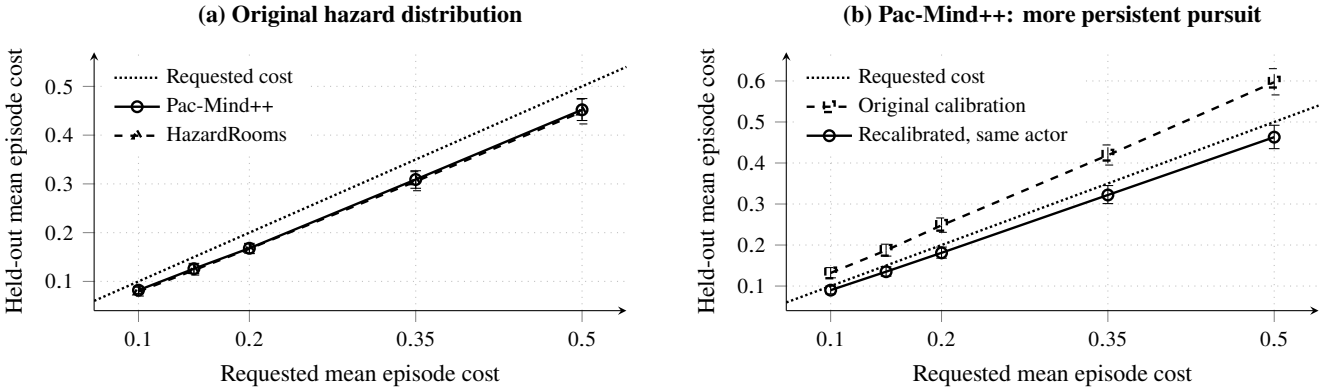


Figure 2: Requested and realized episode cost for validation-selected fixed settings. Error bars are 95% intervals, and the dotted line marks agreement with the request. Left: original task distributions. Right: increasing ghost pursuit from 75% to 90% raises costs above the original requests; recalibration on shifted validation rollouts recovers settings with aggregate costs below them. The network remains frozen. Table 5 gives the full original-distribution results.

Increasing ghost pursuit to 90% breaks the original cost selection. At $b = 0.20$, Pac-Mind++ cost rises from 0.168 to 0.248 [0.231, 0.266], and 53.1% of seed–layout cells exceed the request. Recalibrating the same actor on shifted validation layouts lowers cost to 0.181 [0.168, 0.195] as reward falls from 134.2 to 128.3 (Figure 2b). Refitting the alarm separately lowers shifted reset-based failure from 9.1% to 7.2%, a paired change of -1.9 [-2.9, -0.9] points; Appendix D.5 gives the full comparison.

5.6 Action choice and confidence

Changing w from 0.2 to 0.7 changes the highest-ranked action on 21.8 [19.1, 24.7]% of shared Pac-Mind++ observations and 25.6 [22.8, 28.5]% of HazardRooms observations. The rates rise to 39.7 [35.8, 43.7]% and 44.2 [39.8, 48.7]% on the junction subset, where at least three directions are passable and a hazard is visible. Figure 3 illustrates one such route change on an unchanged observation stack.

We initially expected confidence changes to explain much of the short-window response. Entropy-matched seeking does reduce 100-action failure to 8.4% and 13.1%, reproducing about 55% and 50% of the reductions observed after raising the weight. The cautious policy still has lower failure by 3.4 [2.3, 4.5] and 5.2 [3.8, 6.7] percentage points, with mean entropy-matching errors of just 0.0003 and 0.0006 nats.

Matching entropy leaves room for different probability assignments to both top-ranked and lower-ranked actions, and the branches go on to visit different states. Action-choice and confidence effects therefore remain coupled in this diagnostic. Detailed continuation outcomes appear in Appendix E.

5.7 Coverage and shared training

The full-budget library has a small advantage over MC-PPO across most of the cost range in Figure 4; Appendix Table 13 gives hypervolumes and intervals. CCPO and MC-PPO are particularly close in HazardRooms, where their frontiers cross. At

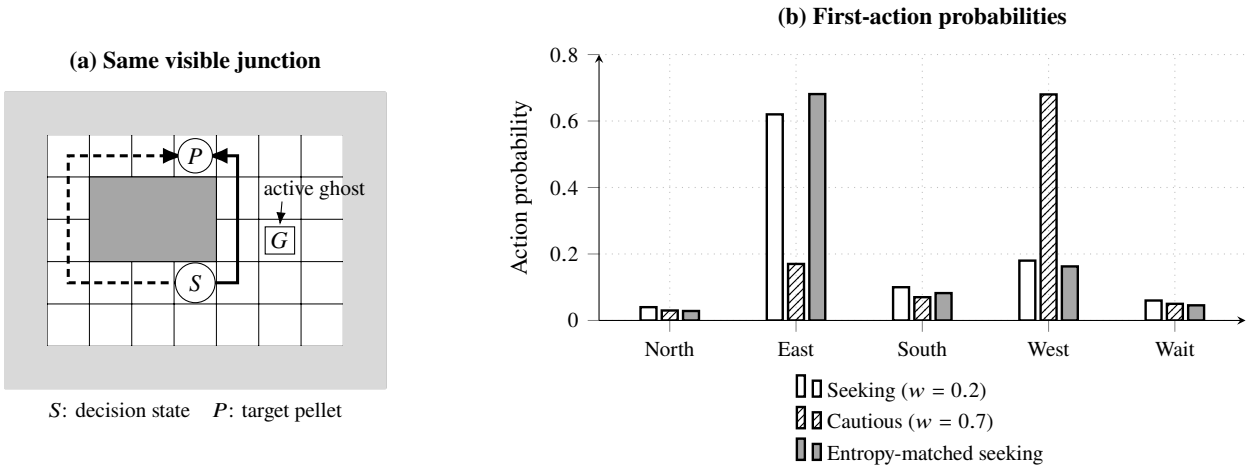


Figure 3: A Pac-Mind++ decision state. The map shows two candidate routes to a pellet. Seeking favors the short eastern route, while the cautious setting favors the western detour. The four-frame input is unchanged. At $T(2/7) = 0.917$, seeking entropy is 1.133 nats and cautious entropy is 1.005 nats. Entropy matching lowers the seeking temperature to 0.792 and preserves its preference for east.

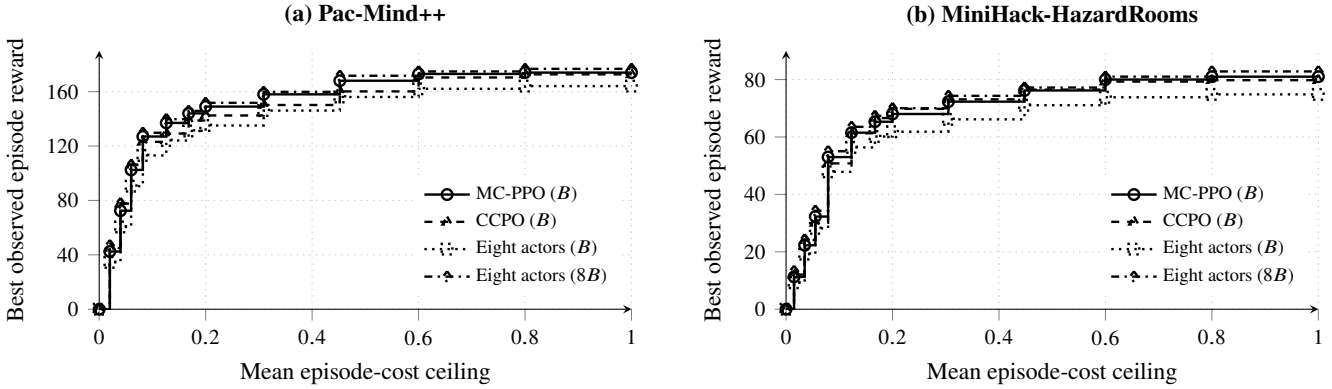


Figure 4: Reward–cost frontiers of the frozen policies. At each cost ceiling, the curve averages the seedwise best observed reward among qualifying settings, using zero where none qualifies. Steps occur as higher cost ceilings admit additional settings. Both libraries use logit mixing. These envelopes use test results; the selections evaluated in Figure 2 use validation rollouts.

total budget B, MC-PPO outperforms the eight-actor mixtures. With full training, the action mixture reaches hypervolumes of 0.784 and 0.719, compared with MC-PPO's 0.768 and 0.699: the shared actor retains 98.0% and 97.2% of the library's mean coverage using one-eighth of its policy-training interactions. Logit mixing gives similar coverage while preserving the temperature ordering.

The paired hypervolume difference between MC-PPO and CCPO is 0.021 $[0.001, 0.041]$ in Pac-Mind++. In HazardRooms it is 0.001 $[-0.016, 0.018]$.

At unseen cell centers, MC-PPO's normalized response errors are 0.018 and 0.029, compared with 0.044 and 0.061 for Concat-PPO. Smooth interpolation coexists with occasional cost reversals: increasing w raises total episode cost by more than 0.02 in 4.2% of Pac-Mind++ adjacent-weight comparisons and 6.3% of HazardRooms comparisons, while the discounted-cost rates are 1.5% and 2.7%. Among total-cost reversals, discounted cost is unchanged or lower in 68% and 62% of cases.

The ablations in Appendix E.2 help account for these results. Parameter-matched concatenation lowers coverage, and removing the smoothness penalty worsens interpolation and cost ordering with little change in coverage. Adding a bonus for disagreement between distant settings raises action divergence from 0.175 to 0.291; in Pac-Mind++, hypervolume falls from 0.768 to 0.751 and total-cost reversals rise from 4.2% to 8.9%.

6 Discussion

Reversible caution recovers completion and reward in both tasks, at the expense of a small increase in failure relative to latching. The preferred operating point will therefore depend on how a task values failure, timeout, and progress. Our default thresholds encode one such choice, and the sensitivity results in Table 8 show how completion and failure vary around it.

The calibration audit exposes a difficulty with validation drawn only from seeking trajectories. Reference-risk predictions run low after the supervisor has changed the visited-state

distribution, and the discrepancy grows under more persistent pursuit. Refitting on shifted validation rollouts improves both cost selection and alarm reliability, although the remaining failure and layout-exceedance rates indicate that the new calibration also needs checking.

The training comparison favors MC-PPO over eight specialists sharing the same interaction budget, with fully trained libraries doing somewhat better and CCPO remaining competitive. A deployment budget must also include calibration: static selection alone uses 28,928 episodes per trained replicate, in addition to the alarm-fitting and checking rollouts. These collections can be reused across requests, but a changed relationship between settings and outcomes calls for new validation data. Appendix D.6 accounts for these costs separately from policy training.

Our experiments cover discrete, partially observed navigation with explicit hazard costs and one controlled change in ghost behavior. Within that scope, aggregate cost selection leaves considerable variation across layouts, and the visited-state audit reveals residual underprediction. Broader claims would require additional tasks and hazard shifts, together with calibration checks on the states reached by the deployed supervisor.

7 Conclusion

MC-PPO combines a conditioned actor with cost settings selected on validation rollouts and a reversible supervisor. In the two navigation tasks, it approaches the coverage of a more expensive specialist library and supports interventions that reduce failure while preserving task progress. CCPO gives comparable results with a different conditioning scheme. The remaining calibration errors, especially after switching or a hazard shift, motivate collecting validation data that better reflects the supervisor's visited states.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, volume 70 of PMLR, pages 22–31, 2017.

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

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36, article 26, 2022.

Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The NetHack Learning Environment. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *International Conference on Machine Learning*, pages 625–632, 2005.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. Pareto conditioned networks. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 1110–1118, 2022.

Mikayel Samvelyan, Robert Kirk, Vitaly Kurin, Jack Parker-Holder, Minqi Jiang, Eric Hambro, Fabio Petroni, Heinrich Küttler, Edward Grefenstette, and Tim Rocktäschel. MiniHack the planet: A sandbox for open-ended reinforcement learning research. In *NeurIPS Datasets and Benchmarks*, 2021.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International Conference on Machine Learning*, volume 37 of PMLR, pages 1312–1320, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Yihang Yao, Zuxin Liu, Zhepeng Cen, Jiacheng Zhu, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constraint-conditioned policy optimization for versatile safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Temperature, Scalarization, and Mixing

For fixed observation x and weight w, write $z = z_\theta(x, w)$ and $p_T(a) = \exp(z_a/T)/Z(T)$. Then

$$\mathcal{H}(p_T) = \log Z(T) - \frac{\mathbb{E}_{p_T}[z]}{T}, \qquad \frac{\mathrm{d}\log Z}{\mathrm{d}T} = -\frac{\mathbb{E}_{p_T}[z]}{T^2}, \qquad \frac{\mathrm{d}\mathbb{E}_{p_T}[z]}{\mathrm{d}T} = -\frac{\mathrm{Var}_{p_T}(z)}{T^2}. \tag{7}$$

Differentiating entropy gives

$$\frac{\mathrm{d}\mathcal{H}(p_T)}{\mathrm{d}T} = \frac{\mathrm{Var}_{p_T}(z)}{T^3} \geq 0. \tag{8}$$

Positive scaling preserves action rankings, and the entropy ordering also holds in τ because $T(\tau)$ is increasing. Both statements concern fixed inputs; a temperature change can alter the states visited during a rollout.

For a constant setting and values conditioned on the full observable history, linearity gives

$$Q_w = (1-w)Q_R - wQ_C, \qquad V_w = (1-w)V_R - wV_C, \qquad A_w = (1-w)A_R - wA_C. \tag{9}$$

With the same discount, trace parameter, termination masks, and bootstrap convention in both streams, the combination also commutes with generalized advantage estimation. We therefore combine advantages before normalization.

For neighboring specialists, let $\alpha = (w_{j+1} - w)/(w_{j+1} - w_j)$. The two mixture rules are

$$p_T^{\text{action}} = \alpha\,\mathrm{softmax}(z_j/T) + (1-\alpha)\,\mathrm{softmax}(z_{j+1}/T), \tag{10}$$

$$p_T^{\text{logit}} = \mathrm{softmax}\big((\alpha z_j + (1-\alpha)z_{j+1})/T\big). \tag{11}$$

Both mixtures are continuous in weight and recover the specialist at a grid vertex. Logit mixing satisfies (8). Action mixing can reverse the entropy ordering when the specialists disagree and become uncertain at different rates: with two actions, equal mixture weights, $z_j = (10, 0)$, and $z_{j+1} = (0, 1)$, entropy decreases from 0.67434 nats at $T = 0.7$ to 0.62660 nats at $T = 1.8$. We center each specialist's logits by subtracting its action mean before mixing and retain their original scales.

A.1 Discounted and total cost can have opposite orderings

Consider two policies with exactly known returns. Policy A receives scaled reward one and cost 0.1 immediately; policy B receives scaled reward 0.99 immediately and cost 0.2 after 100 actions, with no other outcomes. At $\gamma = 0.99$, B's discounted cost is $0.2\gamma^{100} \simeq 0.07321$. The scalarized values at $w = 0.2$ are 0.7800 for A and 0.7774 for B, favoring A. At $w = 0.7$, they become 0.2300 and 0.2458, favoring B, whose discounted cost is lower and total cost is twice as high.

B Training and Comparator Implementation

B.1 Networks and PPO update

Pac-Mind++ stacks four $15 \times 15 \times 8$ frames along the channel dimension. Three 3×3 convolutional layers use 32, 64, and 64 output channels, strides 1, 2, and 2, padding one, and ReLU activations; a linear layer maps the flattened output to 256 features. The actor then applies two 256-unit FiLM layers with layer normalization. A two-hidden-layer MLP of width 32 maps w to 1,024 FiLM scale and shift parameters, with the final projection initialized to identity modulation.

The critic concatenates (w, τ) to the encoder features, applies two 256-unit ReLU layers, and outputs separate reward and cost values. HazardRooms uses the same encoder with health, key inventory, and remaining time appended to its features. Actor heads have five outputs in Pac-Mind++ and nine in HazardRooms. No action mask depends on mood or hidden hazards. Concat-PPO uses width 298 in its actor layers, giving an actor parameter count within one percent of the FiLM actor; the encoder and critics are unchanged.

The training grid is $\{0.1, 0.2, \ldots, 0.8\} \times \{0, 1/7, \ldots, 1\}$. One worker is assigned to each pair. Let $y_t^R = \tilde{r}_t$, $y_t^C = \tilde{c}_t$, and let b_t be zero at task termination, including the task deadline, and one otherwise. Component advantages use

$$\delta_t^i = y_t^i + \gamma b_t V_i(x_{t+1}, u) - V_i(x_t, u), \tag{12}$$

$$\widehat{A}_t^i = \delta_t^i + \gamma\lambda b_t \widehat{A}_{t+1}^i, \qquad i \in \{R, C\}. \tag{13}$$

A rollout cut bootstraps from the next value and ends the trace. The component advantages and targets are fixed during optimization. The value target is $V_i^{\text{old}} + \widehat{A}^i$. Only the combined actor advantage in (5) is normalized, using one rollout-wide mean and standard deviation.

For $\rho_t = \pi_\theta(a_t \mid x_t, u)/\pi_{\text{old}}(a_t \mid x_t, u)$ and normalized advantage $\bar{A}_t^w$, the maximized loss is

$$L_{\text{PPO}} = \mathbb{E}_t \min\left[\rho_t \bar{A}_t^w, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\bar{A}_t^w\right], \tag{14}$$

$$L = L_{\text{PPO}} + 0.01\mathbb{E}_t[\mathcal{H}(\pi_\theta)] - 0.5(L_{V_R} + L_{V_C}) - 10^{-3}R_s, \tag{15}$$

$$R_s = \mathbb{E}_{x,j,k} \frac{D_{\text{JS}}\left(\pi_{w_j,\tau_k}(\cdot \mid x), \pi_{w_{j+1},\tau_k}(\cdot \mid x)\right)}{(w_{j+1} - w_j)^2}. \tag{16}$$

Value losses are mean squared errors. Each regularization batch contains 256 current-rollout observations, each assigned a uniformly sampled adjacent weight pair and training temperature. Gradients pass through both distributions. The default regularizer acts along the weight coordinate, with distant-setting separation evaluated as an ablation.

Each rollout has 8,192 transitions. Four optimization epochs use minibatches of 1,024 transitions, with 16 from each setting and sampling without replacement within an epoch. PPO uses $\epsilon = 0.2$, GAE uses $\lambda = 0.95$, and Adam uses learning rate 2.5×10^{-4} and numerical epsilon 10^{-5}. The gradient norm is clipped at 0.5. Training budgets are rounded up to full rollouts. A weight specialist keeps the same settings but divides its 64 workers equally across the eight temperatures.

B.2 CCPO and libraries

The categorical CCPO adaptation retains the factorized reward and cost critics and conditioned variational update of Yao et al. (2023). State–action features and threshold features each have dimension 64. Action expectations in critic targets and policy updates are exact finite sums. The nonparametric E-step target is

$$q_d(a \mid x) \propto \pi_{\text{old}}(a \mid x, d) \exp\left(\frac{Q_R(x,a,d) - \lambda_d Q_C(x,a,d)}{\eta_d}\right). \tag{17}$$

The dual variables enforce the sampled cost condition and an E-step KL radius of 0.1. The categorical M-step fits this target with a mean old-to-new policy KL limit of 0.01. These are the finite-action implementation choices used in this study.

CCPO uses replay capacity 10^6, batch size 256, Adam learning rate 3×10^{-4}, and target averaging coefficient 0.005. After 10,000 warm-up transitions, each collected transition is followed by one critic/policy update cycle. Eight behavior thresholds are log-spaced over $[0.01, 0.5]$ in discounted-cost units; static evaluation uses 113 thresholds. Threshold inputs are normalized in log space. Reward and cost use the same scales as PPO.

The eight-actor libraries use (10) or (11), evaluating at most two members per decision. The 64-actor reference stores one member per weight–temperature pair and selects the nearest setting or bilinearly mixes the four corner action distributions. Vertex outcomes are reused across these rules. Reported actor counts measure storage; runtime work is described by the number of member evaluations per decision.

Neutral PPO fixes $(w, \tau) = (0.5, 2/7)$. The loss-only control has an unconditioned actor: weight enters only the training advantage and temperature enters only an entropy coefficient $0.01(0.5 + \tau)$. Temperature-only PPO fixes $w = 0.5$ and trains across the eight temperatures. Lagrangian PPO uses discounted-cost targets of 0.10 and 0.07 in the two tasks. Its nonnegative multiplier is updated after each rollout using completed-episode cost, with projected step size 0.01; updates with no completed episodes are skipped.

C Environment Specifications

C.1 Pac-Mind++

A connected 20×20 wall layout is generated by randomized depth-first carving, followed by the removal of interior wall segments until the mean shortest-path distance between free cells is below 35. The outer boundary remains closed. Pellets occupy 35% of initially free cells. Four power pellets are placed in different quadrants. The player and ghosts start on distinct free cells; each ghost is at least six graph-distance steps from the player. Placement is resampled within the selected layout at reset.

Each step has five available actions: the four cardinal moves and wait. A wall move consumes a step without moving the player. After the player's move, each active ghost chooses a shortest-path move toward the player with probability 0.75, a random legal move with probability 0.15, and its previous direction with probability 0.10 when legal. Ties are broken uniformly. Collisions are checked both after the player move and after ghost moves. A power pellet makes all ghosts vulnerable for 40 steps. Vulnerable ghosts use random legal moves; a captured ghost respawns at its assigned start cell after ten steps and remains vulnerable if the common timer is still active.

The eight observation planes encode walls, pellets, power pellets, active ghosts, vulnerable ghosts, the player, the normalized vulnerability timer, and normalized remaining episode time. Outside-board cells are encoded as walls, and all objects within the

crop are visible without an additional line-of-sight mask. The four-frame stack conveys recent motion within that crop. Terminal collision incurs unit cost and suppresses the proximity charge on that step. Rewards are +1 per pellet, +10 per power pellet, +25 per captured vulnerable ghost, +100 for clearing the maze, and −0.01 per action. On nonfatal steps, proximity cost is $0.05 \max(0, 3 - d_t)/3$, where d_t is Manhattan distance to the nearest active ghost. Collision adds no reward penalty.

C.2 MiniHack-HazardRooms

The custom level generator uses a 20×20 map containing three to five rooms linked by one-cell passages. It places one to three lava pools, one to three spike passages, up to two monsters, up to two locked doors, keys when required, and one amulet goal. A graph check rejects levels with no traversable route to the goal after necessary doors are unlocked. A route avoiding static lava and spikes exists by construction, although moving monsters can still make that route hazardous. Training and evaluation use separate layout seeds and fresh object placements.

The action set contains four cardinal moves, four directional door interactions, and wait. A door interaction opens an unlocked door or uses an available key on a locked door. Pickups are automatic. These command sequences are defined by the wrapper as one decision; rewards and costs include all game events caused by that decision. All methods use this same decision count for training budgets and episode horizons. Illegal or inapplicable actions consume a decision without progress. The action mask is independent of hidden hazards and the control input.

The semantic planes encode walls, lava, spikes, monsters, keys or designated pickups, closed doors, the goal, and the player. Current health fraction, key inventory, and remaining episode fraction are supplied separately. Spike cost is charged once per entry, with a fatal transition receiving unit cost and suppressing any additional spike charge. At most four designated pickups yield +5 each, once per pickup. Reaching the amulet gives +100, and each decision gives −0.01.

D Calibration and Evaluation Details

D.1 Aggregation and static selection

For each seed, outcomes first average repeated episodes within a layout and then average layouts with equal weight. Hypervolume is computed from that seed's points and only then averaged across seeds:

$$\mathrm{HV}_s = \mathrm{area}\left(\bigcup_u [0, (\widehat{R}_{su}/S_R)_+] \times [0, (1 - \widehat{C}_{su})_+] \right), \tag{18}$$

where $(q)_+ = \max(q, 0)$. Positive reward values remain uncapped, and normalization is fixed independently of test outcomes. Figure 4 averages the seedwise best-reward envelopes at the displayed cost ceilings, using zero where no evaluated setting has positive reward within the ceiling. For sorted nonnegative points (x_i, y_i), the area is $\sum_i (x_i - x_{i-1}) \max_{j \geq i} y_j$, with $x_0 = 0$.

Each bootstrap sample independently draws 12 seed indices and the domain's full number of layout indices with replacement. The layout draw is shared across seeds, settings, and paired methods. Episodes and continuations in a sampled seed–layout cell stay together. Validation choices are held fixed during test resampling.

Static selection uses 256 validation episodes per candidate, balanced across layouts. We take $U_C(u)$ as the 95th percentile of 5,000 layout-cluster bootstrap means, then choose the qualifying setting with highest reward, breaking ties by lower cost. The bootstrap screen is pointwise over a finite candidate set; it provides no simultaneous confidence guarantee after selection. Every MC-PPO seed has a qualifying setting for all five requests in Table 5. Selected settings stay fixed throughout each test episode, and alarm supervisors are evaluated separately.

Layout exceedance uses 64 fresh episodes per selected seed–layout–budget cell. With their mean denoted $\widehat{C}^{\mathrm{check}}_{s\ell}(b)$, the reported proportion is

$$\widehat{p}_{\mathrm{layout}}(b) = \frac{1}{12L} \sum_{s=1}^{12} \sum_{\ell=1}^{L} \mathbf{1}\{\widehat{C}^{\mathrm{check}}_{s\ell}(b) > b\}. \tag{19}$$

This proportion measures exceedance of estimated layout-conditional means and includes finite-rollout variation. Each indicator summarizes 64 episodes in one seed–layout–budget cell.

D.2 Response selection and alarm calibration

Validation layouts are split in half. The first half selects each competitor's seeking setting and fits its alarm. Seeking maximizes mean episode reward among candidates whose upper cost estimate is at most 0.7. Weight-based competitors have 15 candidate weights from 0.1 to 0.8 in increments of 0.05 at $\tau = 2/7$; CCPO has 15 log-spaced thresholds. MC-PPO keeps its specified seeking setting.

Table 5: MC-PPO at five requested mean episode-cost budgets. Validation selects one fixed setting per seed and request. Cost includes a 95% interval; reward and completion are means. The last column gives the share of seed–layout cells whose mean cost, estimated from 64 additional episodes, exceeds b.

Requested cost b	Test mean cost	Episode reward	Completion (%)	Layouts above b (%)
Pac-Mind++				
0.10	0.082 [0.075, 0.090]	125.8	39.6	20.3
0.15	0.126 [0.116, 0.137]	136.7	46.9	19.5
0.20	0.168 [0.158, 0.178]	142.4	51.8	18.8
0.35	0.309 [0.291, 0.327]	157.1	58.8	21.5
0.50	0.452 [0.430, 0.475]	166.0	60.2	23.4
MiniHack-HazardRooms				
0.10	0.079 [0.070, 0.088]	52.7	47.6	21.9
0.15	0.123 [0.113, 0.134]	60.9	55.8	20.8
0.20	0.167 [0.157, 0.177]	64.8	60.1	19.8
0.35	0.305 [0.286, 0.325]	71.8	66.4	23.4
0.50	0.448 [0.423, 0.474]	75.9	70.2	25.8

The second half supplies 256 common MC-PPO alarm snapshots for response selection. We test each candidate with four paired continuations per snapshot, requiring at least 85% of its own hold-branch reward and eventual completion at least as high as the hold branch. Among qualifying candidates, we select the one with least 100-action failure, breaking ties by completion and then reward. A method keeps its seeking setting when none qualifies. Weight ramps are linear in w, CCPO ramps are linear in the normalized log-threshold input, and temperatures remain fixed.

Alarm maps use 512 fresh seeking episodes on the fit layouts and another 512 on the check layouts. Training labels mark failure within 100 subsequent decisions under continued seeking; states are sampled every ten decisions. Remaining follow-up near a task deadline is padded with nonfailure absorbing steps. PPO's score is V_C at the seeking setting. CCPO uses the seeking-policy expectation of Q_C. Libraries use a weighted component value score, calibrated on trajectories of the actual mixture. Isotonic fitting minimizes squared error with the pool-adjacent-violators algorithm. We use the resulting nondecreasing step function with midpoint boundaries between fitted blocks and clip predictions beyond the observed score range to the nearest endpoint. Calibration is fitted separately for each trained replicate.

For reversible supervision, the minimum 20-decision dwell starts when a mode is entered, including its ramp. Threshold counters are inactive during ramps. Release requires five consecutive scores below 0.04 after the dwell has elapsed. Re-entry requires a score above 0.10 after the seeking dwell has elapsed. A release begins from the cautious setting and returns to the original seeking setting over ten actions. Episodes ending during a dwell or ramp keep their observed outcome. The same reset, action, and environment seed pairs are reused across hold, latched, and reversible branches.

Alarm incidence and failures with no prior alarm use all reset episodes as their denominator. Timely recall conditions on eventual failure and requires an alarm at least ten decisions before death. Re-entry in Table 3 uses all episodes and requires two distinct entries into caution separated by a release. The reliability table below concerns continued seeking, which is the alarm's reference behavior.

Table 6: Alarm reliability on validation-check layouts. Each bin reports the mean predicted and observed probability of failure within 100 decisions under continued seeking.

	Pac-Mind++		HazardRooms	
Probability bin	Mean prediction	Failure frequency	Mean prediction	Failure frequency
[0, 0.05)	0.025	0.029	0.027	0.033
[0.05, 0.10)	0.073	0.078	0.076	0.083
[0.10, 0.20)	0.139	0.148	0.151	0.166
[0.20, 1]	0.286	0.310	0.327	0.354

D.3 Supervisor controls and threshold sensitivity

We reuse 1,024 reset episodes per seed with settings $(0.2, 2/7)$ and $(0.7, 2/7)$ and the ramp, dwell, and release rules from Appendix D.2, changing the switching signal for each comparison.

The hazard-blind schedule uses constant entry and release probabilities fitted from validation mode-exit counts and eligible exposure, with episode endings treated as censored observations. The release clock starts the five-decision counter, whose

Table 7: Alarm frequency and missed failures in ordinary MC-PPO seeking episodes, without alarm-quota sampling. Estimates include 95% confidence intervals; the denominator is given for each measure.

Measure	Denominator	Pac-Mind++	HazardRooms
At least one alarm (%)	All episodes	24.8 [22.9, 26.8]	37.6 [35.4, 39.9]
Failure with no prior alarm (%)	All episodes	1.3 [0.9, 1.8]	4.9 [3.8, 6.0]
Alarm at least ten actions before failure (%)	Failed episodes	71.6 [65.9, 77.1]	65.0 [59.8, 70.0]
Median first-alarm time (actions)	Episodes with alarms	147 [134, 161]	91 [82, 101]

duration is excluded from the clock-fitting exposure. This schedule depends on elapsed time alone; we fit it on validation data and report the resulting test caution fraction.

The visible-hazard alarm enters when the nearest visible hazard is within Manhattan distance d_{in} and releases after five decisions beyond d_{out}. Active ghosts count in Pac-Mind++; monsters, lava, and spikes count in HazardRooms. No visible hazard gives infinite distance. The 20 candidate pairs use $d_{\text{in}} \in \{1, 2, 3, 4, 5\}$ and $d_{\text{out}} - d_{\text{in}} \in \{1, 2, 3, 4\}$.

Raw-critic entry candidates are the 80th, 85th, 90th, 95th, and 97.5th percentiles of seeking-reference validation scores. Release candidates are the 20th, 40th, 60th, and 70th percentiles. Each raw-critic or distance pair receives 512 check-layout episodes. Selection maximizes completion subject to validation failure at most 6% in Pac-Mind++ or 16% in HazardRooms; reward breaks ties. If no pair qualifies, minimum failure is used. Selection is per seed, before testing.

We also implement the calibrated trigger and release sets as raw-score intervals using the same step-function boundaries. This direct conversion reproduces identical trajectories on paired seeds. The separately tuned raw-score control selects its cutoffs from deployment outcomes on validation layouts.

The oracle clones the live simulator and its random streams, then runs seeking for up to 100 actions or the original deadline. Its binary signal records whether that continuation fails, and the usual $0.10/0.04$ thresholds and dwell rules govern switching. The live rollout is unaffected by look-ahead. Access to hidden state and future randomness makes the oracle an offline reference.

Table 8: Sensitivity to calibrated trigger and release thresholds. All other settings remain fixed. Completion is a percentage; failure is a percentage with a 95% interval. The first three rows vary the trigger and the last two vary release around the default $(0.10, 0.04)$.

		Pac-Mind++		HazardRooms	
Trigger	Release	Completion	Failure	Completion	Failure
0.08	0.04	64.3	4.6 [3.9, 5.4]	80.5	13.2 [11.7, 14.8]
0.10	0.04	64.6	5.0 [4.2, 5.8]	80.0	14.4 [12.6, 16.3]
0.12	0.04	64.2	5.6 [4.8, 6.5]	79.1	15.6 [13.8, 17.5]
0.10	0.02	63.4	4.8 [4.0, 5.6]	78.4	14.0 [12.3, 15.8]
0.10	0.06	64.8	5.5 [4.7, 6.4]	79.9	15.3 [13.5, 17.2]

D.4 Calibration on states visited after switching

Each seed supplies 256 states sampled uniformly from cautious-mode decisions and 256 from the first 20 decisions after release begins, including ramps. Four independent seeking rollouts per state, without further supervision, label failure within 100 decisions or the original deadline. Test-layout visitation frequencies are retained.

Table 9 reports prediction, reference failure, and their difference. Four-bin error is the frequency-weighted absolute gap across the probability bins in Table 6. Resampling retains seed and layout clusters. The shifted audit scores both maps on the same cautious states, collected under the original supervisor, and the same continued-seeking outcomes.

D.5 Changed hazard dynamics

Pursuit probability rises from 0.75 to 0.90; random moves and continuation of the previous direction each become 0.05. Ghost count, speed, vulnerability, observations, costs, and layouts stay fixed. Recalibration repeats static selection and the seeking-reference alarm fit on shifted validation rollouts, without changing network weights.

We test all five budgets in the static experiment. In the reset experiment, we change the alarm map while retaining the original operating points and reversible supervisor. Table 10 reports both evaluations on untouched test layouts. The episode-cost request applies to static selection; runtime switching uses the alarm thresholds specified above.

Table 9: Reference-risk calibration on visited states. Predictions and failure frequencies are percentages. Gaps and four-bin errors are percentage points; gap intervals are 95% intervals. Every outcome is evaluated under continued seeking from the saved state. The final two rows use identical shifted states and outcome labels.

State pool and alarm map	Prediction	Reference failure	Observed minus predicted	Four-bin error
Pac-Mind++, original hazards				
Cautious mode	10.7	12.3	1.6 [0.9, 2.4]	2.0
After release	3.9	5.1	1.2 [0.5, 1.9]	1.6
MiniHack-HazardRooms				
Cautious mode	14.9	17.4	2.5 [1.5, 3.5]	3.0
After release	4.8	6.7	1.9 [0.9, 3.0]	2.3
Pac-Mind++, shifted hazards				
Cautious mode, original map	16.2	21.1	4.9 [3.7, 6.1]	5.3
Cautious mode, refitted map	19.5	21.1	1.6 [0.8, 2.5]	2.1

Table 10: Pac-Mind++ under changed ghost behavior. Static columns use request $b = 0.20$; the two rightmost columns use reversible supervision from reset. Cost and failure include 95% intervals. Layout exceedance is measured with 64 fresh episodes per seed–layout cell, as in Table 5.

	Fixed setting selected for $b = 0.20$			Reversible supervision	
Hazards and calibration	Cost	Reward	Layouts above b (%)	Completion (%)	Failure (%)
Original	0.168 [0.158, 0.178]	142.4	18.8	64.6	5.0 [4.2, 5.8]
Shifted, original maps	0.248 [0.231, 0.266]	134.2	53.1	56.5	9.1 [7.9, 10.5]
Shifted, refitted maps	0.181 [0.168, 0.195]	128.3	22.7	57.4	7.2 [6.1, 8.4]

D.6 Training, calibration, and deployment costs

Static calibration requires $113 \times 256 = 28{,}928$ episodes per trained replicate; alarm fitting and checking add 1,024 seeking episodes. The same data support all reported requests and calibrated thresholds. Recalibration repeats these collections under changed hazards.

Competitor response selection adds $15 \times 256 \times 4 = 15{,}360$ continuations to the original deadline. MC-PPO uses its specified response without that search. The tuned raw-critic and distance controls each require 20×512 validation episodes. These costs are outside the policy-training budget B; realized decision counts depend on episode length.

Deployment evaluates one MC-PPO actor and its seeking-reference cost critic, followed by a scalar calibration lookup. An interpolated library evaluates at most two actors and their value scores. We report policy-training interactions and rollout counts; wall-clock time, energy, and optimizer cost are outside this accounting. Oracle look-ahead and reference-risk continuations are used only for evaluation.

D.7 Fixed-input and continuation measurements

Diagnostic buffers contain 5,000 live observations from each of neutral PPO, seeking MC-PPO, cautious MC-PPO, and uniform-random rollouts. Ranking tests use a fixed action-order tie rule. Restricting the comparison to states with a top-two probability gap of at least 0.02 at both weights gives change rates of 20.9% and 24.8%, close to the unrestricted results.

The entropy-matched branch follows $w_j = 0.2 + 0.5 \min((j+1)/10, 1)$ as its target, but acts from seeking logits at a fitted temperature $T^*(x, j)$. It recomputes the target entropy on its own current observation at every decision. Bisection is in log temperature and stops at residual 10^{-6} or 60 iterations. An unattainable target uses the closer endpoint; endpoints are selected on 0.4% and 0.8% of decisions. The diagnostic requires two actor evaluations plus the scalar search.

Snapshots include elapsed time, inventory, and the observation stack. Action and environment randomness use separate streams, with four shared continuation seed pairs per snapshot. Every continuation retains the original task deadline. Completion means clearing Pac-Mind++ or obtaining the amulet; surviving to the deadline without completion is a timeout. Inactive decisions are waits or ineffective commands with no change in position, inventory, pickups, or doors. Their proportion is computed within each continuation and then averaged. Completion times are medians among completed continuations.

For held-out center u, response error is the mean absolute difference between its reward and cost and the average of the four corner outcomes, divided by S_R and S_C and averaged over the two channels. Independent repeat evaluations have discrepancies of 0.012 and 0.017 for MC-PPO. Local Jensen–Shannon divergence uses weight displacements of 0.02; distant divergence compares 0.2 with 0.7.

Cost reversals are increases exceeding 0.02 between adjacent weights, tested at each of eight temperatures. Among 672 seed–weight–temperature comparisons, MC-PPO has 10 discounted and 28 total-cost reversals in Pac-Mind++, and 18 discounted and 42 total-cost reversals in HazardRooms. Discounted cost is nonincreasing in 19 of the 28 and 26 of the 42 total-cost reversals. Entropy ordering is tested on 51 temperatures at all 15 weights, using tolerance 10^{-7}. The eight-actor action mixture has entropy reversal rates of 2.3% and 2.9%; MC-PPO and logit mixing have none.

E Additional Continuation and Static Results

Tables 11 and 12 give the within-actor interventions and their full-deadline follow-up. The high-weight branch reduces failure further while lowering completion; its 100-action reward also falls below the 85% retention requirement in both tasks.

Table 11: Paired 100-action continuations from the same MC-PPO seeking-policy alarm states. Targets are reached in ten actions. The entropy-matched branch keeps $w = 0.2$ and adjusts temperature separately at each state. Failure includes deaths during the ramp; reward is measured in the same fixed window, with zero reward after termination. Entries are means with 95% confidence intervals.

		Pac-Mind++		HazardRooms	
Intervention	Target (w, τ)	Failure (%)	Reward	Failure (%)	Reward
Continue seeking	$(0.2, 2/7)$	12.6 [11.4, 13.9]	31.7 [30.2, 33.1]	18.4 [16.7, 20.1]	20.8 [19.5, 22.1]
Lower temperature only	$(0.2, 0)$	10.8 [9.7, 12.0]	31.0 [29.6, 32.4]	16.6 [15.0, 18.2]	20.4 [19.1, 21.7]
Entropy-matched seeking	$w = 0.2,\ T = T^*(x, j)$	8.4 [7.4, 9.5]	30.0 [28.6, 31.4]	13.1 [11.7, 14.6]	19.6 [18.4, 20.9]
Raise cost weight	$(0.7, 2/7)$	5.0 [4.3, 5.8]	28.6 [27.3, 29.9]	7.9 [6.7, 9.2]	18.2 [17.0, 19.4]
Cost weight + lower temp.	$(0.7, 0)$	4.7 [4.0, 5.5]	28.1 [26.8, 29.4]	7.4 [6.3, 8.7]	17.8 [16.6, 19.0]
Cost weight + higher temp.	$(0.7, 4/7)$	6.3 [5.4, 7.3]	28.3 [27.0, 29.6]	11.2 [9.8, 12.7]	17.2 [16.0, 18.4]
High cost weight	$(0.8, 2/7)$	3.9 [3.2, 4.7]	24.8 [23.5, 26.1]	6.2 [5.1, 7.5]	15.1 [14.0, 16.3]

Table 12: Follow-up from the same alarm states to the original task deadline. Completion, failure, and timeout partition all continuations. Completion time is the median number of actions after the alarm among successful continuations. Inactive actions are waits or ineffective commands without a change in position, inventory, pickups, or doors. All entries include 95% confidence intervals.

Response	Completion (%)	Failure (%)	Timeout (%)	Completion time	Inactive (%)
Pac-Mind++					
Continue seeking	56.8 [53.7, 60.0]	27.4 [25.2, 29.7]	15.8 [13.8, 17.9]	628 [596, 662]	4.7 [4.1, 5.4]
Entropy-matched seeking	57.5 [54.3, 60.7]	21.3 [19.3, 23.4]	21.2 [18.9, 23.7]	701 [663, 741]	7.2 [6.3, 8.2]
Cost weight to 0.7	63.4 [60.0, 66.6]	13.6 [11.9, 15.4]	23.0 [20.4, 25.7]	801 [758, 845]	9.8 [8.6, 11.1]
Cost weight to 0.8	49.2 [45.8, 52.6]	10.2 [8.7, 11.9]	40.6 [37.4, 43.8]	1108 [1044, 1178]	27.4 [24.8, 30.1]
MiniHack-HazardRooms					
Continue seeking	48.5 [45.2, 51.9]	41.8 [38.9, 44.7]	9.7 [8.1, 11.4]	216 [204, 229]	6.1 [5.2, 7.1]
Entropy-matched seeking	53.0 [49.6, 56.4]	33.7 [31.0, 36.5]	13.3 [11.5, 15.3]	243 [228, 259]	8.7 [7.5, 10.0]
Cost weight to 0.7	62.9 [59.4, 66.3]	23.6 [21.2, 26.1]	13.5 [11.5, 15.6]	281 [264, 299]	11.4 [10.0, 12.9]
Cost weight to 0.8	48.6 [45.0, 52.1]	18.8 [16.6, 21.1]	32.6 [29.5, 35.8]	397 [373, 423]	34.0 [30.9, 37.1]

E.1 Additional baselines

Table 13 includes the fixed-policy references and larger libraries. Fixed actors supply an operating-point reference, while controllable methods supply multiple settings. Table 14 compares validation-selected settings at $b = 0.20$. The Lagrangian baseline retains its training-time discounted-cost target; the table records its resulting episode cost.

E.2 Architectural ablations

The single-critic variant predicts scalarized return with the same hidden layers. The joint-temperature variant adds τ to the FiLM input while retaining the final gate, and the distant-separation variant adds $0.005\mathbb{E}_x[\min(D_{\mathrm{JS}}(\pi_{0.2,\tau}, \pi_{0.7,\tau}), 0.3)]$ to the maximized objective. We hold all other training choices fixed. Allowing temperature into FiLM raises mean coverage slightly and creates entropy reversals in 14.7% of comparisons.

Table 13: Static reward–cost coverage on held-out layouts. Entries are mean hypervolume with 95% confidence intervals. B is the per-domain training budget in Section 4. The eight-actor mixtures share members and differ in their deployment rule. Fixed-policy rows measure coverage at a single operating point.

Method	Total steps	Actors	Pac-Mind++ HV	HazardRooms HV
Neutral PPO	B	1	0.471 [0.455, 0.487]	0.416 [0.402, 0.428]
Loss-only control	B	1	0.450 [0.434, 0.466]	0.390 [0.375, 0.404]
Temperature-only PPO	B	1	0.616 [0.597, 0.637]	0.531 [0.508, 0.552]
Lagrangian PPO	B	1	0.584 [0.559, 0.606]	0.509 [0.488, 0.529]
Concat-PPO	B	1	0.697 [0.674, 0.720]	0.616 [0.592, 0.640]
CCPO, categorical	B	1	0.747 [0.723, 0.771]	0.698 [0.676, 0.722]
MC-PPO	B	1	0.768 [0.749, 0.788]	0.699 [0.676, 0.723]
Eight actors, action mixture	B	8	0.706 [0.686, 0.727]	0.637 [0.617, 0.657]
Eight actors, logit mixture	B	8	0.711 [0.689, 0.733]	0.643 [0.621, 0.665]
Eight actors, action mixture	$8B$	8	0.784 [0.759, 0.808]	0.719 [0.693, 0.745]
Eight actors, logit mixture	$8B$	8	0.781 [0.756, 0.805]	0.715 [0.689, 0.741]
64 actors, nearest setting	B	64	0.631 [0.604, 0.656]	0.527 [0.508, 0.545]
64 actors, nearest setting	$64B$	64	0.809 [0.790, 0.828]	0.736 [0.711, 0.762]
64 actors, action mixture	$64B$	64	0.809 [0.791, 0.829]	0.736 [0.711, 0.763]

Table 14: Test outcomes at a requested mean episode-cost budget of 0.2, with settings selected on validation data. Entries are means with 95% confidence intervals. Achieved costs vary across methods at the shared request.

	Pac-Mind++		HazardRooms	
Method	Reward	Cost	Reward	Cost
Lagrangian PPO	139.0 [132.7, 145.1]	0.160 [0.152, 0.168]	61.0 [58.2, 63.8]	0.165 [0.156, 0.175]
Concat-PPO	129.5 [124.4, 134.3]	0.163 [0.155, 0.172]	56.4 [54.0, 58.9]	0.166 [0.157, 0.176]
CCPO, categorical	141.3 [136.5, 146.2]	0.182 [0.176, 0.187]	67.0 [64.7, 69.4]	0.181 [0.173, 0.189]
MC-PPO	142.4 [138.7, 146.4]	0.168 [0.158, 0.178]	64.8 [62.5, 67.2]	0.167 [0.157, 0.177]
Eight actors, action mixture, $8B$	144.5 [139.3, 149.6]	0.168 [0.159, 0.176]	67.7 [64.7, 70.6]	0.170 [0.160, 0.180]
64 actors, nearest, $64B$	148.0 [143.5, 152.8]	0.137 [0.122, 0.153]	68.8 [66.2, 71.5]	0.156 [0.140, 0.170]
64 actors, mixture, $64B$	152.2 [147.5, 157.1]	0.167 [0.151, 0.181]	69.1 [66.2, 72.1]	0.167 [0.159, 0.176]

Table 15: Pac-Mind++ ablations at the same training budget. Entries are means with 95% confidence intervals. Cost reversals are increases above 0.02 between adjacent weights, measured separately on discounted and undiscounted episode cost. Entropy reversals use fixed observations and a 10^{-7} tolerance.

			Cost reversals (%)		Entropy
Variant	Hypervolume	Response error	Discounted	Total	reversals (%)
Parameter-matched concatenation	0.697 [0.674, 0.720]	0.044 [0.036, 0.053]	6.8 [4.9, 8.9]	12.8 [9.7, 16.1]	0
Single scalar critic	0.742 [0.719, 0.764]	0.030 [0.025, 0.037]	3.9 [2.6, 5.2]	7.6 [5.4, 10.0]	0
No smoothness penalty	0.763 [0.742, 0.784]	0.027 [0.022, 0.033]	3.3 [2.1, 4.8]	7.0 [5.2, 9.0]	0
Temperature also enters FiLM	0.776 [0.756, 0.795]	0.028 [0.023, 0.034]	3.0 [1.8, 4.3]	6.3 [4.5, 8.1]	14.7 [12.2, 17.3]
Add distant-separation bonus	0.751 [0.727, 0.775]	0.026 [0.021, 0.032]	4.6 [3.1, 6.3]	8.9 [6.7, 11.4]	0
Full MC-PPO	0.768 [0.749, 0.788]	0.018 [0.015, 0.022]	1.5 [0.9, 2.3]	4.2 [2.9, 5.4]	0