

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

Dario Fumarola Jin Tan Ruan

Amazon Web Services
New York, New York

Abstract

A preference-conditioned policy can change its behavior after training, but its cost weight does not specify the cost that will follow. We study calibrated runtime deployment of a frozen PPO actor with two inputs: a relative cost weight and a sampling temperature. Validation rollouts map requested mean costs to settings, and a reference-risk alarm controls when to enter and leave a cautious mode. In Pac-Mind++ and MiniHack-HazardRooms, one 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, increasing 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 the observed reduction. With the actor held fixed, risk-based switching improves on hazard-blind schedules and visible-hazard alarms; calibrated thresholds perform similarly to separately tuned critic thresholds. Releasing caution recovers completion and reward. A changed-hazard test shows how cost selection and alarm reliability degrade, and what recalibration recovers without retraining. CCPO remains a competitive single-policy alternative. The contribution is a deployment procedure and evaluation that distinguish available behaviors from the rules used to select them.

1 Introduction

A navigation policy that takes a short route in an empty corridor may need to take a detour when a hazard appears. A preference-conditioned policy can make that change without retraining. The practical question is how to choose its input. A cost weight of 0.7 has no meaning in raw cost units, and a policy that incurs little cost may simply have stopped making progress.

Mood-Conditioned PPO (MC-PPO) exposes a cost weight through feature conditioning and a temperature through final-logit scaling. With the network frozen, validation chooses settings for mean-cost requests, and a calibrated alarm controls entry into and release from caution.

Our contribution is a deployment procedure for conditioned policies. We evaluate cost selection, paired interventions, and repeated switching, then hold the actor fixed to compare supervisors and test recalibration under changed hazards. CCPO and interpolated specialist libraries receive the same treatment. The comparison distinguishes useful behaviors from the rules used to select them.

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); successor features support changes in reward specification (Barreto et al., 2017). Multi-objective reinforcement learning studies policies for competing outcomes (Hayes et al., 2022). Pareto Conditioned Networks select behavior through desired returns (Reymond et al., 2022). Our actor instead receives a scalarization weight and uses on-policy PPO training (Schulman et al., 2017).

A directly relevant comparison is Constraint-Conditioned Policy Optimization (CCPO), which learns policies for different

cost thresholds without retraining (Yao et al., 2023). Unlike a fixed-constraint method such as CPO (Achiam et al., 2017), CCPO already exposes a deployment control. We compare against a categorical adaptation, using achieved reward and cost to align its threshold input with our preference input. Both are evaluated through validation-based setting selection and calibrated alarms.

FiLM applies condition-dependent affine transformations to hidden features (Perez et al., 2018). We use it only for the cost weight. Keeping temperature outside the learned logit computation gives a fixed-input entropy ordering and preserves action rankings. These properties also hold for specialists whose logits are mixed before temperature scaling. The empirical comparison therefore concerns training allocation, behavioral response, and calibration rather than possession of a temperature control.

Post-hoc calibration maps prediction scores to outcome frequencies (Niculescu-Mizil and Caruana, 2005). We apply it to the failure risk of a specified reference policy, rather than treating a cost value as a probability. Calibration can deteriorate under distribution shift (Ovadia et al., 2019). Here the supervisor itself changes which states are visited, so we check the reference-risk estimate during caution and after release as well as on 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. Its effective input is $u = (w, \tau)$, with

$w \in [0.1, 0.8]$ and $\tau \in [0, 1]$. A public pair of positive reward and cost gains is normalized as

$$w = \frac{\beta_C}{\beta_R + \beta_C}. \quad (1)$$

Only the ratio matters. The interface has two independent controls, not three.

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

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

The discount is $\gamma = 0.99$. Evaluation separately records total episode reward, total episode cost, discounted cost, and terminal failure. In particular, a late cost can be small under the training discount and still matter at the task deadline.

3.2 Actor and optimization

An observation encoder feeds two FiLM-conditioned actor layers. The weight w determines their feature scales and shifts. The resulting logits $z_\theta(x, w)$ are converted to a distribution by

$$T(\tau) = 0.7(1.8/0.7)^\tau, \quad (3)$$

$$\pi_\theta(a | x, w, \tau) = \text{softmax}(z_\theta(x, w)/T(\tau))_a. \quad (4)$$

Temperature enters neither the encoder nor the FiLM layers. At a fixed observation and weight, increasing T preserves every action ranking and cannot reduce entropy. The elementary identity behind this property is given in Appendix A.

The reward and cost critics share the observation encoder but use a separate trunk. Both receive (w, τ) , since either input can change future returns. Generalized advantage estimation (Schulman et al., 2016) is applied separately to the scaled reward and cost streams. The actor uses their combination

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

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

Training allocates one worker to each point of an 8×8 grid over weight and temperature. Each worker contributes 128 transitions per update, so long episodes do not change the allocation across settings. A fixed entropy bonus of 0.01 and a small adjacent-weight smoothness penalty regularize the update. The entropy bonus is independent of the temperature input. Appendix B specifies 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 $\hat{R}_{\text{val}}(u)$ and an upper estimate $U_C(u)$ of mean episode cost. A request b is translated into

$$\hat{u}(b) \in \arg \max_{u: U_C(u) \leq b} \hat{R}_{\text{val}}(u). \quad (6)$$

If no setting passes the screen, the request is unsupported. The selected setting is then evaluated on held-out layouts. This rule uses measured episode cost, rather than assuming that cost is a monotone function of the weight.

Online changes use a different signal. At every decision, the cost critic is evaluated at a fixed reward-seeking reference setting. Isotonic regression, fitted on validation trajectories, maps that score to an estimate of failure probability within 100 actions under continued seeking. Evaluating the same reference setting throughout an episode prevents a control change from suppressing the score merely by changing the critic’s input.

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 second supervisor can release caution and trigger again. Both use the frozen actor, critics, and calibration map. The latter tests repeated control changes directly; training itself uses 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. The policy has five actions. Collision with an active ghost terminates the episode and incurs unit cost; proximity to an active ghost adds a smaller nonterminal cost. The deadline is 3,000 actions. MiniHack-HazardRooms is a custom task built with MiniHack and the NetHack Learning Environment (Samvelyan et al., 2021; Küttler et al., 2020). It has rooms, doors, keys, lava, spikes, monsters, and an amulet goal. Fatal transitions cost one, nonfatal spike entries cost 0.2, and the deadline is 1,000 decisions. Its wrapper exposes nine actions. Appendix C gives the rewards, observations, and map generators.

The training, validation, and test sets contain 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. Each method has 12 independently trained replicates and is evaluated at its final checkpoint. Reward scales are $S_R = 200$ and 100; cost uses $S_C = 1$ in both tasks. All methods use the same task interfaces and layout splits.

The main baselines are parameter-matched Concat-PPO, categorical CCPO, and eight weight-specialist actors. Concat-PPO replaces FiLM with concatenation at both actor layers and otherwise keeps the same training procedure. CCPO retains threshold-conditioned value factorization and variational policy updates, with action expectations evaluated by exact categorical sums. Its input is a discounted-cost threshold; validation maps it to episode outcomes just as for the weight-based methods.

Each specialist is trained at one weight and all eight temperatures. Neighboring specialists can be combined by mixing action probabilities or by mixing logits before a common temperature gate. Both give continuous weight control, and the latter preserves the fixed-input temperature property. The matched-budget library receives $B/8$ interactions per actor;

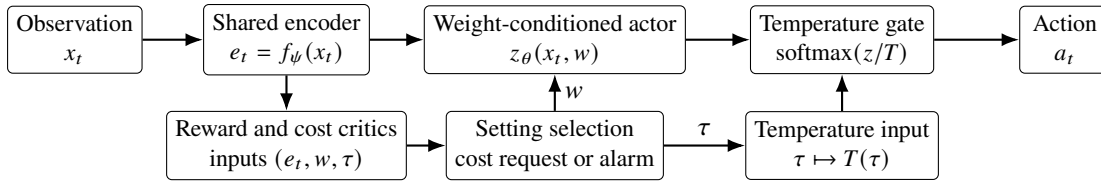


Figure 1: Runtime control around a frozen actor. Validation maps requested mean costs to settings; the online supervisor uses a calibrated critic score to change the weight. Temperature acts only at the output gate. The critics read the control inputs but do not produce actor logits.

the full-budget library receives B per actor. A complete library is one training replicate. The budget B counts policy-training interactions. Calibration rollouts, evaluation, and optimizer work are 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

Static evaluation uses 64 grid settings and 49 cell centers, with 256 test episodes per setting and training seed, balanced across layouts. The centers are absent from training trajectories and regularization. CCPO receives the same number of evaluated thresholds. Reward–cost hypervolume uses the fixed raw reference of zero reward and cost one, with reward divided by S_R ; it is computed per trained seed before averaging.

We evaluate the selection rule in (6) at requested mean episode costs $b \in \{0.10, 0.15, 0.20, 0.35, 0.50\}$. At each selected setting we report reward, completion, and achieved cost. A separate set of 64 episodes per test layout estimates how often a layout’s mean cost exceeds the request. This distinguishes a target for the test-layout mixture from one that holds on every layout.

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

Seeking MC-PPO rollouts supply 256 saved alarm states per seed, balanced across test layouts. Alarms are enabled after 25 elapsed actions. A snapshot contains the simulator state, observation stack, elapsed time, and random-number state. Four continuation seeds per snapshot are shared across interventions, giving 12,288 continuations per domain and branch. The first outcome window starts at the original alarm and lasts 100 actions. We then follow every branch to completion, failure, or its original deadline.

An entropy-matched seeking branch tests how much of the observed response can be reproduced by changing confidence alone. At each state that it visits, it computes the cautious ramp’s entropy but samples from the seeking logits. Bisection chooses a temperature in $[0.05, 20]$ that matches that entropy as closely as possible. This diagnostic retains the seeking

action rankings and has a wider temperature range than the public interface.

Competitor response settings are selected on validation snapshots to minimize 100-action failure while retaining at least 85% of the method’s seeking reward and not lowering eventual completion. Each competitor has both a hold branch and a response branch at the common snapshots. Both replace the generating actor, so their difference isolates the control change. MC-PPO uses the fixed pair $(0.2, 2/7)$ and $(0.7, 2/7)$. All ramps last ten actions.

The reset-based study evaluates 1,024 paired episodes per seed for each complete policy–supervisor system. It includes episodes without alarms. Each method starts from its own seeking setting and uses its own calibrated alarm. Alongside the hold and latched branches, a reversible branch releases caution when the reference alarm probability stays below 0.04 for five decisions. A mode lasts at least 20 decisions, and every transition uses a ten-action ramp. The supervisor can re-enter caution throughout the episode. These thresholds and durations are shared across methods and fixed before testing.

4.4 Supervisor controls and hazard shift

We hold the MC-PPO actor, its two operating points, and the ramp and dwell rules fixed while changing the supervisor. Controls use hazard-blind schedules, visible-hazard distance, raw critic thresholds tuned per seed, and a simulator oracle. The oracle checks whether continued seeking would fail within 100 actions using a copy of the simulator and future random streams. It is an offline reference. All branches use the same reset seeds. Appendix D.3 specifies threshold selection and the timing control.

A separate audit samples states actually visited during caution and the first 20 decisions after release. Each is relabeled using independent continued-seeking rollouts, preserving the alarm’s reference outcome. Finally, we change Pac-Mind++ ghost behavior from 75% to 90% pursuit, without changing the layouts. We test the existing calibration, then refit it using only shifted validation rollouts. No actor or critic weights change.

4.5 Uncertainty

Tables report averages across 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. Paired comparisons use identical resampling indices. Continuations from one seed–layout cell stay together,

and validation-selected settings remain fixed during test resampling. This treats the trained seed, rather than each rollout, as an independent model replicate (Agarwal et al., 2021).

5 Results

5.1 A useful response from reset

The first comparison is the complete deployed system, including the alarm. 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). HazardRooms completion rises by 5.4 percentage points, with a 95% interval of [3.1, 7.7]. Pac-Mind++ completion changes by a smaller 1.6 [−0.2, 3.4] points. Reward falls in the pellet-collection task but rises in HazardRooms, where more episodes reach the amulet.

CCPO achieves similar intervention gains. The full-budget specialist library has the highest mean completion and lowest mean failure of the tested systems. At the same total training budget, however, sharing one actor is more effective than dividing the data among eight specialists. MC-PPO uses one actor and budget B ; the strongest library uses eight actors and budget $8B$.

Alarm coverage explains why the reset-based reductions are smaller than those from selected hazardous states. The alarm fires in 24.8% of ordinary MC-PPO seeking episodes in Pac-Mind++ and 37.6% in HazardRooms. Failures without a 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 reports the reliability and denominator measurements.

5.2 What changes at the alarm

Common-state continuations remove differences in when and where an alarm occurs (Table 2). Increasing 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. Reward in 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.

The benefit persists beyond the first 100 actions. In the MC-PPO branches, eventual failure falls from 27.4% to 13.6% and from 41.8% to 23.6%. Completion rises from 56.8% to 63.4% and from 48.5% to 62.9%. These branches keep the original deadline, so the response does more than postpone a failure past the short measurement window.

Excessive caution gives a different outcome. Raising the weight 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%. The lower cost is partly purchased by not finishing. The full follow-up table is in Appendix E.

Timing matters as well. Delaying 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. Deaths during the delay remain in the original outcome window.

5.3 Returning to reward seeking

Latching in the cautious mode is simple, but it can remain expensive after the hazard has passed. The reversible supervisor returns to the seeking setting after a sustained low alarm score and can enter caution again. On MC-PPO, this increases Pac-Mind++ completion from 62.1% to 64.6% and reward from 164.0 to 171.2. Failure rises from 4.7% to 5.0%, still below the 8.1% seeking baseline. In HazardRooms, completion rises from 77.3% to 80.0%, while failure moves from 13.8% to 14.4%.

Table 3 reports paired changes relative to the latched supervisor. The same pattern appears for CCPO and the specialist libraries: releasing caution recovers completion and reward at 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. Thus the test includes actual re-entry, not only a single change followed by release.

These results separate two choices that a static frontier combines. The setting determines what the actor does while caution is active. The release rule determines how long it pays that cost. Keeping a useful cautious policy active indefinitely is not the best deployment strategy in either task.

5.4 What does the supervisor add?

Table 4 changes only the supervisor. Hazard-blind schedules spend nearly the same fraction of decisions in caution as the calibrated supervisor, but leave failure higher by 2.2 [1.3, 3.1] percentage points in Pac-Mind++ and 4.3 [2.6, 6.0] in HazardRooms. The benefit therefore depends on when caution is used, not just how often. Visible-hazard alarms recover part of it. The simulator oracle improves further with access to hidden state and future randomness.

Separately tuned raw critic thresholds perform similarly to calibrated thresholds. Indeed, a monotone calibration map assigns a probability scale to the critic score but adds no ranking information: its trigger and release boundaries can be implemented directly as raw-score cutoffs. Calibration provides a common risk scale for choosing and checking those boundaries, rather than a new hazard detector.

The visited-state audit finds systematic underprediction. In cautious states, 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). Threshold sweeps show the resulting trade-off between earlier intervention and time spent in caution; the default is one operating point on that trade-off.

5.5 How well does the cost request transfer?

The validation selector finds a setting for every trained seed at all five requested budgets (Figure 2a and Table 5). Test mean cost remains below the request in each aggregate com-

Table 1: Complete episodes from reset: hold $\rightarrow$ 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
<i>Pac-Mind++</i>					
MC-PPO	B	60.5 $\rightarrow$ 62.1	8.1 $\rightarrow$ 4.7	173.0 $\rightarrow$ 164.0	-3.4 [-4.5, -2.3]
Concat-PPO	B	55.2 $\rightarrow$ 56.4	10.4 $\rightarrow$ 6.6	163.0 $\rightarrow$ 155.0	-3.8 [-5.2, -2.4]
CCPO	B	61.4 $\rightarrow$ 63.5	7.9 $\rightarrow$ 4.6	165.0 $\rightarrow$ 160.0	-3.3 [-4.5, -2.1]
Eight actors, logit mixture	B	56.8 $\rightarrow$ 58.4	9.5 $\rightarrow$ 5.9	166.0 $\rightarrow$ 158.0	-3.6 [-4.8, -2.4]
Eight actors, logit mixture	$8B$	63.4 $\rightarrow$ 65.5	7.2 $\rightarrow$ 4.0	177.0 $\rightarrow$ 170.0	-3.2 [-4.3, -2.1]
<i>MiniHack-HazardRooms</i>					
MC-PPO	B	71.9 $\rightarrow$ 77.3	20.6 $\rightarrow$ 13.8	78.0 $\rightarrow$ 82.0	-6.8 [-8.8, -4.8]
Concat-PPO	B	65.4 $\rightarrow$ 69.8	24.9 $\rightarrow$ 18.7	70.8 $\rightarrow$ 73.9	-6.2 [-8.5, -4.0]
CCPO	B	73.1 $\rightarrow$ 79.0	19.1 $\rightarrow$ 12.7	77.3 $\rightarrow$ 82.8	-6.4 [-8.4, -4.4]
Eight actors, logit mixture	B	68.0 $\rightarrow$ 72.6	23.0 $\rightarrow$ 16.7	73.6 $\rightarrow$ 77.0	-6.3 [-8.5, -4.1]
Eight actors, logit mixture	$8B$	75.4 $\rightarrow$ 81.1	17.3 $\rightarrow$ 11.5	80.9 $\rightarrow$ 86.1	-5.8 [-7.7, -3.9]

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.

Method	Budget	100-action failure (%)		Response to task deadline (%)	
		Hold	Response	Completion	Failure
<i>Pac-Mind++</i>					
MC-PPO	<i>B</i>	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	<i>B</i>	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	<i>B</i>	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	<i>B</i>	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	<i>8B</i>	12.2 [11.0, 13.5]	4.5 [3.8, 5.3]	65.5 [62.2, 68.7]	12.7 [11.0, 14.5]
<i>MiniHack-HazardRooms</i>					
MC-PPO	<i>B</i>	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	<i>B</i>	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	<i>B</i>	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	<i>B</i>	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	<i>8B</i>	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 (%)
<i>Pac-Mind++</i>					
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
<i>MiniHack-HazardRooms</i>					
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

parison. At $b = 0.20$, for example, MC-PPO reaches costs of 0.168 [0.158, 0.178] and 0.167 [0.157, 0.177]. The achieved costs do not match the request exactly: the selector chooses from a finite set and screens with an upper estimate rather than the sample mean.

The task-level price of a tight request is substantial. Reducing b from 0.50 to 0.10 lowers Pac-Mind++ completion from 60.2% to 39.6% and HazardRooms completion from 70.2% to 47.6%. Mean cost alone would conceal this loss of progress. Individual layouts also vary: at $b = 0.20$, estimated layout mean cost exceeds the target in 18.8% and 19.8% of seed-

Table 4: Supervisor comparison with the same frozen MC-PPO actor and operating points. All switching rules are reversible and use the same 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 on validation data, not on test outcomes.

Supervisor	Completion (%)	Failure (%)	Episode reward	Caution time (%)
<i>Pac-Mind++</i>				
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
<i>MiniHack-HazardRooms</i>				
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

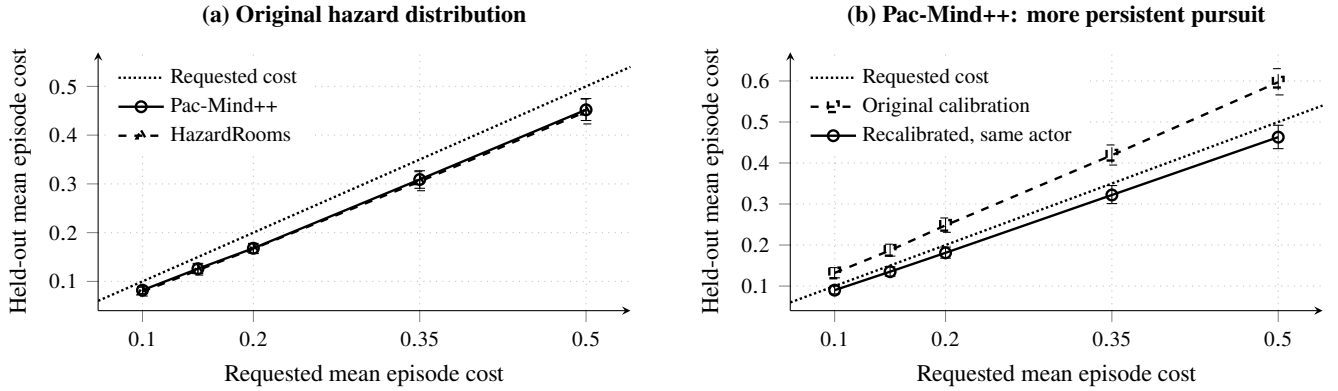


Figure 2: Requested and realized episode cost for validation-selected fixed settings. Error bars are 95% intervals. The dotted line marks exact agreement with the request. Left: original task distributions. Right: increasing ghost pursuit from 75% to 90% invalidates the original cost selection; shifted validation rollouts recover settings whose aggregate costs lie below the requests. Neither test outcomes nor network updates enter recalibration. Table 5 retains the full original-distribution results.

layout cells, even though the aggregate means pass it. The calibration is useful for selecting average operating points; layout-specific control requires more information.

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.

More persistent ghost pursuit changes this result. With the original calibration, Pac-Mind++ cost at $b = 0.20$ 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 reduces cost to 0.181 [0.168, 0.195], at a reward reduction from 134.2 to 128.3 (Figure 2b). Refitting the alarm separately lowers shifted reset-based failure from 9.1% to 7.2%, with paired change $-1.9 [-2.9, -0.9]$ points. Recalibration recovers useful operating points, but does not restore the original deployment outcomes (Appendix D.5).

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. On the junction subset, the rates are 39.7 [35.8, 43.7]% and 44.2 [39.8, 48.7]%. Figure 3 shows a state where the weight changes the preferred route. Temperature scaling alone cannot make that change.

Entropy-matched seeking reduces 100-action failure to 8.4% and 13.1%, reproducing about 55% and 50% of the observed reductions from raising the weight. The cautious policy remains better by 3.4 [2.3, 4.5] and 5.2 [3.8, 6.7] percentage points. Matching is close: the mean entropy errors are 0.0003 and 0.0006 nats. Equal entropy leaves the allocation of probability across actions different, and the branches subsequently visit different states.

The percentages summarize this diagnostic comparison, rather than partitioning the benefit into independent causal

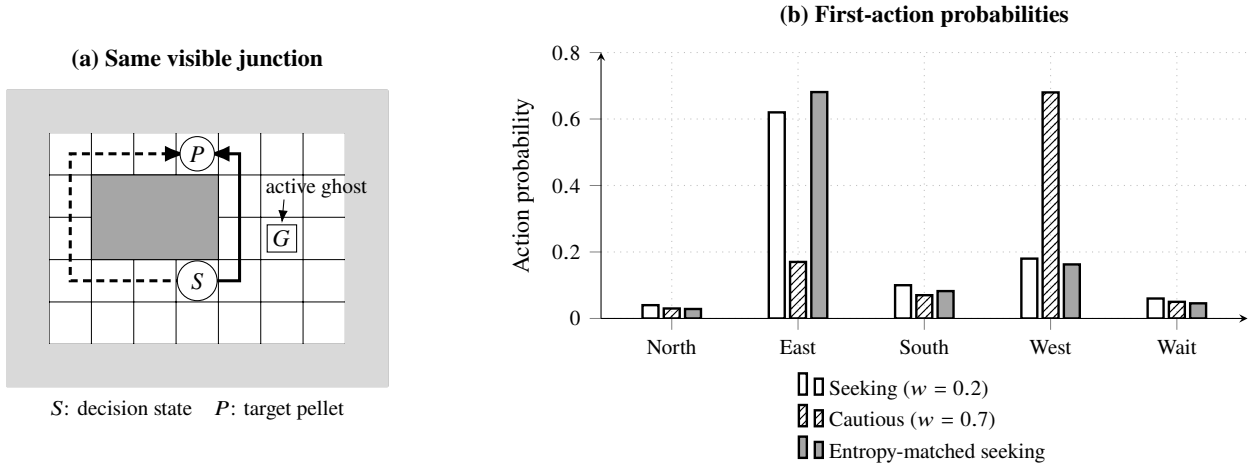


Figure 3: A Pac-Mind++ decision state. The map shows two candidate routes to a pellet. Seeking favors the short eastern route, while the cautious setting favors the western detour. The four-frame input is unchanged. At $T(2/7) = 0.917$, seeking entropy is 1.133 nats and cautious entropy is 1.005 nats.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

effects. Cost conditioning produces a response that entropy matching alone does not recover, through changes among both top-ranked and lower-ranked actions. Detailed continuation outcomes are in Appendix E.

5.7 Coverage and shared training

Figure 4 shows the reward–cost frontiers; Appendix Table 13 gives hypervolumes and intervals. The full-budget library has a small advantage across most of the tested cost range. CCPO and MC-PPO are especially close in HazardRooms, with neither uniformly ahead. At total training budget B , MC-PPO outperforms the eight-actor mixtures. The fully trained action mixture reaches hypervolumes of 0.784 and 0.719, compared with MC-PPO’s 0.768 and 0.699. The shared actor therefore retains 98.0% and 97.2% of that library’s mean coverage with one-eighth of its policy-training interactions. Mixing logits instead of action probabilities gives similar coverage and preserves the temperature ordering.

CCPO is close to MC-PPO: the paired hypervolume differences are 0.021 [0.001, 0.041] in Pac-Mind++ and 0.001 [−0.016, 0.018] in HazardRooms. The clearest advantage of shared training is over splitting a fixed interaction budget among specialists, rather than over another well-trained conditioned policy.

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. Cost ordering is less regular than interpolation. 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. The corresponding discounted-cost rates are 1.5% and 2.7%. Among the total-cost reversals, discounted cost does not increase in 68% and 62% of cases. Discounting and the deployment metric can therefore disagree about the direction of a change.

Ablations identify which choices improve the interface (Appendix E.2). Replacing FiLM with parameter-matched

concatenation lowers coverage. Removing the smoothness penalty changes coverage little but makes interpolation and cost ordering worse. A bonus that rewards disagreement between distant settings raises action divergence from 0.175 to 0.291, yet lowers coverage and roughly doubles total-cost reversals. Larger visible action changes are not necessarily better control.

6 Discussion

Available behaviors and the rules for using them are distinct parts of a deployed policy. The fixed-actor comparison makes that distinction concrete: random schedules can spend as long in caution as a risk-based supervisor yet prevent fewer failures. Releasing caution also matters, because a useful detour can become unproductive waiting when maintained too long.

Calibration puts episode-cost requests and alarm thresholds on measurable scales. It does not improve the critic’s ordering of states, and per-seed threshold tuning gives similar deployment results. Its practical role is to support a common risk specification, expose prediction errors, and make recalibration possible without training another actor. The changed-hazard experiment demonstrates both the need for that check and its cost in reward.

The choice of actor remains a resource decision. MC-PPO outperforms eight specialists when their total policy-training interactions are equal. Fully trained libraries do somewhat better, while CCPO is a competitive single-policy alternative. The 98.0% and 97.2% coverage ratios concern training interactions, not end-to-end computation: static calibration alone uses 28,928 episodes per trained replicate. Calibration rollouts can be reused across requests, but must be refreshed when the measured relationship changes.

The evidence covers discrete, partially observed navigation with explicit hazard costs and one controlled change in ghost behavior. Aggregate cost calibration does not make the same request hold on every layout. The visited-state audit also

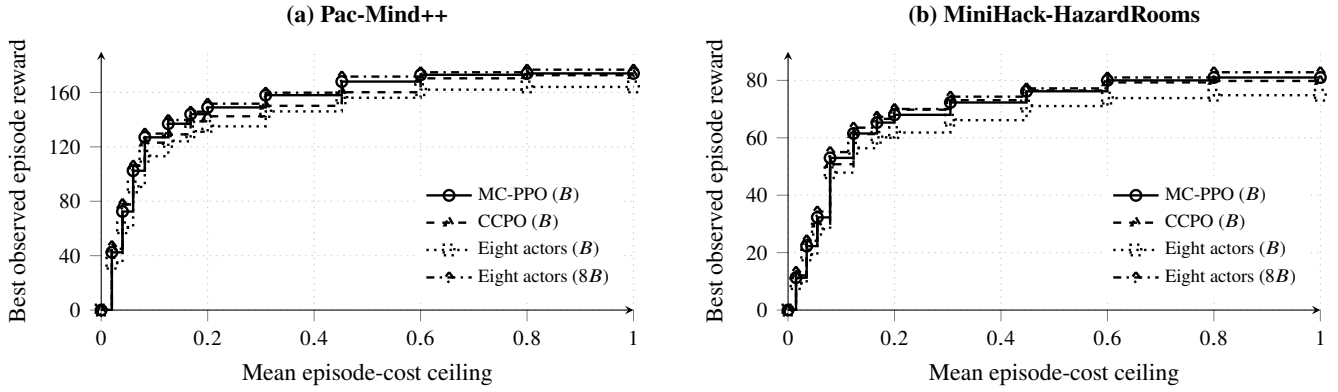


Figure 4: Reward–cost frontiers of the frozen policies. At each cost ceiling, the curve reports the mean across seeds of the best observed reward among qualifying settings, or zero if none qualifies. A step occurs when a higher cost ceiling admits another setting. Both libraries use logit mixing. These test-set envelopes describe available behavior; the setting selector in Figure 2 uses validation data only.

finds underprediction after switching. These measurements define where the tested supervisors work and where more conservative settings or new calibration data are needed.

7 Conclusion

A frozen conditioned actor can provide useful reward–cost choices, but deployment performance also depends on when those choices are made. With the actor fixed, risk-based timing reduces failure more than time-matched caution, and release restores progress. Calibration makes the choices interpretable in task units and supports adjustment to changed hazards without retraining. The resulting method is a deployment layer for conditioned policies, evaluated through cost selection, intervention timing, and task completion rather than static coverage alone.

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, Mingqi 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

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

Differentiating entropy gives

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

Positive scaling also preserves all action rankings. Since $T(\tau)$ increases with τ , the same entropy ordering holds in that coordinate. These are fixed-input properties; the states visited by the policy can change with temperature.

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

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

The same combination commutes with generalized advantage estimation when both streams use the same discount, trace parameter, termination masks, and bootstrap convention. This is the reason to combine the advantages before normalization.

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

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

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

Both are continuous in the weight and recover the specialist at a grid vertex. Logit mixing satisfies (8); action mixing need not. For example, with two actions, equal mixture weights, $z_j = (10, 0)$, and $z_{j+1} = (0, 1)$, the action mixture’s entropy decreases from 0.67434 nats at $T = 0.7$ to 0.62660 nats at $T = 1.8$. The two specialists disagree and become uncertain at different rates. In implementation, logits are centered by subtracting their action mean before mixing; their scales are otherwise unchanged.

A.1 Discounted and total cost can have opposite orderings

Consider two policies. 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 $0.2\gamma^{100} \approx 0.07321$. At $w = 0.2$, the scalarized values are 0.7800 for A and 0.7774 for B, so A is preferred. At $w = 0.7$, they are 0.2300 and 0.2458, so B is preferred. Increasing the weight lowers discounted cost while doubling total cost. This can occur without approximation error.

B Training and Comparator Implementation

B.1 Networks and PPO update

Pac-Mind++ uses four $15 \times 15 \times 8$ frames stacked along the channel dimension. The encoder has three 3×3 convolutional layers with 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 has 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. The final projection starts at 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 $\{0.1, 0.2, \dots, 0.8\} \times \{0, 1/7, \dots, 1\}$. 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

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

$$\hat{A}_t^i = \delta_t^i + \gamma \lambda b_t \hat{A}_{t+1}^i, \quad i \in \{R, C\}. \quad (13)$$

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 $V_t^{\text{old}} + \hat{A}_t^i$. Only the combined actor advantage in (5) is normalized, using one rollout-wide mean and standard deviation.

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

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

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

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

Value losses are mean squared errors. Each regularization batch contains 256 current-rollout observations, with a uniformly sampled adjacent weight pair and training temperature assigned to each. Both distributions receive gradients. There is no penalty along the temperature coordinate and no default distant-separation bonus.

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

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

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) and evaluate at most two members per decision. The 64-actor reference trains one member per weight–temperature pair and uses either nearest-setting selection or bilinear mixing of the four corner action distributions. Vertex outcomes are reused across these rules. Actor counts are storage counts; no hardware latency comparison is made.

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. Objects within the crop

are visible without an additional line-of-sight mask. The four-frame stack conveys recent motion but does not reveal ghosts outside the crop. Terminal collision incurs unit cost and replaces, rather than adds to, 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 itself 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 on entry rather than on every turn spent on the tile. A fatal transition has unit cost and no additional spike charge. At most four designated pickups yield the +5 reward, each once. Reaching the amulet gives +100, and each decision gives -0.01 . These wrappers define the custom HazardRooms task used here.

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:

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

where $(q)_+ = \max(q, 0)$. Reward is not upper-clipped. The normalization is fixed independently of test outcomes. Figure 4 averages the seedwise best-reward envelopes at the displayed cost ceilings; its curves are 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 validation layouts. $U_C(u)$ is the 95th percentile of 5,000 layout-cluster bootstrap means. The highest-reward qualifying setting is chosen, with lower cost breaking ties. This is a pointwise screen over a finite candidate set, not a simultaneous confidence guarantee after selection. All five requests in Table 5 have a qualifying setting for every MC-PPO seed. The static setting is held fixed throughout each test episode; the alarm supervisors are evaluated separately.

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

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

It describes exceedance of an estimated conditional mean. It includes finite-rollout variation and differs from the frequency with which an individual episode exceeds b .

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

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 is the share of seed-layout cells whose mean cost, estimated from 64 additional episodes, exceeds b . It is not an episode failure rate.

Requested cost b	Test mean cost	Episode reward	Completion (%)	Layouts above b (%)
<i>Pac-Mind++</i>				
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
<i>MiniHack-HazardRooms</i>				
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

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.

The second half supplies 256 common MC-PPO alarm snapshots for response selection. Each candidate response is tested with four paired continuations per snapshot and must retain at least 85% of its own hold-branch reward and not lower eventual completion. The qualifying candidate with least 100-action failure is selected; ties favor completion and then reward. If none qualifies, the method keeps its seeking setting. Weight ramps are linear in w ; CCPO ramps in its normalized log-threshold input. 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.

Probability bin	Pac-Mind++		HazardRooms	
	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

The comparison reuses 1,024 reset episodes per seed, settings (0.2, 2/7) and (0.7, 2/7), and the ramp, dwell, and release rules in Appendix D.2. Only the switching signal changes.

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]

The hazard-blind schedule uses constant entry and release probabilities fitted from validation mode-exit counts and eligible exposure. Episode endings are censored. The release clock starts the five-decision counter rather than bypassing it. Clock-fitting exposure excludes that counter. The schedule reads elapsed time but no observation; its test caution fraction is measured, not forced to match.

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 $d_{in} \in \{1, 2, 3, 4, 5\}$ and $d_{out} - d_{in} \in \{1, 2, 3, 4\}$.

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.

Implementing the calibrated trigger and release sets directly as raw-score intervals reproduces identical trajectories on paired seeds. This uses the same step-function boundaries and is distinct from tuning cutoffs by deployment outcomes.

The oracle clones the live simulator and its random streams, then runs seeking for up to 100 actions or the original deadline. Its signal is one if that continuation fails and zero otherwise. The usual 0.10/0.04 thresholds and dwell rules act on this signal. Look-ahead does not modify the live rollout. Hidden state and future randomness make this an offline reference rather than a deployable alarm.

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).

Trigger	Release	Pac-Mind++		HazardRooms	
		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.

The static experiment tests all five budgets. The reset experiment changes only the alarm map, retaining the original operating points and reversible supervisor. Table 10 reports both on untouched test layouts. The static request and runtime switching are separate deployment modes; the latter is not constrained to that episode-cost budget.

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
<i>Pac-Mind++</i> , 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
<i>MiniHack-HazardRooms</i>				
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
<i>Pac-Mind++</i> , 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.

Hazards and calibration	Fixed setting selected for $b = 0.20$			Reversible supervision	
	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. The comparison reports training interactions, not wall-clock time, energy, or optimizer cost. Oracle look-ahead and reference-risk continuations are evaluation-only computations.

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. Thus the effect is not confined to near ties.

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 shows why the default response is not selected by failure alone. Its 100-action reward also fails 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.

Intervention	Target (w, τ)	Pac-Mind++		HazardRooms	
		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 (%)
<i>Pac-Mind++</i>					
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]
<i>MiniHack-HazardRooms</i>					
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. The distant-separation variant adds $0.005\mathbb{E}_x[\min(D_{JS}(\pi_{0.2, \tau}, \pi_{0.7, \tau}), 0.3)]$ to the maximized objective. All other training choices are fixed. Allowing temperature into FiLM raises mean coverage slightly but creates entropy reversals in 14.7% of comparisons. The restricted actor chooses the simpler temperature semantics.

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 use the same members and differ only in the deployment rule. Fixed-policy rows are operating-point references, not estimates of a complete frontier.

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. Realized costs need not be identical at the same requested budget.

Method	Pac-Mind++		HazardRooms	
	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.

Variant	Hypervolume	Response error	Cost reversals (%)		Entropy reversals (%)
			Discounted	Total	
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