THINK UNTIL NO FUTURE THOUGHT CAN HELP: CERTIFIED REMAINING-GAIN STOPPING FOR ADAPTIVE-COMPUTE POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive-compute policies commonly stop on confidence or latent convergence rather than the decision consequence of further computation. We introduce RARE^2, which certifies the *remaining gain*: the largest improvement in a fixed reference utility that any uncomputed future prefix could provide over the current action. A normalized split-conformal score gives simultaneous coverage across depths under exchangeability, supporting adaptive stopping and post-calibration tolerance choice. We derive stopping-risk, expected-gain, oracle-depth, critic-error, shift and performance-difference guarantees, and distinguish population occupancy calibration from empirical recalibration. At matched violation rates on six graph tasks, RARE^2 requires 2.54 recurrent steps versus 5.29 for the best heuristic gate calibrated by Learn-Then-Test. Across eight planning tasks and forty training runs, RARE^2 removes 70.2% of recurrent steps at 95.14% simultaneous coverage. On all five MinAtar games, the policy is trained by Q-learning for 5M frames with five seeds, using its own frozen critic as reference. Held-out coverage averages 95.26%, with 49.7% fewer recurrent steps and 99.2% of full-compute return. Equivalence holds within $\pm 5\%$ ($p = 0.041$); a confidence gate at the same risk target requires $1.82\times$ the depth for indistinguishable return. CPU speedups range from $3.91\times$ to $4.91\times$, and a scalar remaining-gain head matches the structured lattice at lower cost. Empirical occupancy recalibration restores near-nominal held-out coverage on average after adaptive deployment shifts the state distribution.

1 INTRODUCTION

Most reinforcement-learning policies allocate the same inference depth to reflexive corrections and decisions requiring long-horizon planning. Weight-tied computation can solve tasks beyond the capabilities of shallow policies and generalize beyond training horizons without additional parameters (Ghugare et al., 2026). Robot policies allocate latent refinement to difficult states (Tur et al., 2026), and generative controllers learn state-dependent compute schedules (Li et al., 2026). These advances raise a decision question: when is it safe to stop computing?

Confidence, action agreement and latent convergence do not measure the consequence of stopping: two equally good actions may induce low confidence, while a prefix can be confidently wrong. Binary halting classifiers discard the magnitude of future improvements and require new labels and retraining when the tolerance changes. Calibrated early-exit methods instead certify agreement with the full-compute output (Schuster et al., 2021; 2022; Jazbec et al., 2024; Wu et al., 2025), which need not reflect decision utility.

We certify the remaining gain. A frozen recurrent policy produces actions $a_1(x), \dots, a_K(x)$, evaluated by a fixed reference utility $q(x, a)$, such as an exact action value, a frozen critic or a paired-rollout estimator. The gain at future prefix j over the current action at prefix k is

$$\Delta_{k,j}(x) = \big[q(x, a_j(x)) - q(x, a_k(x))\big]_+, \qquad 1 \le k < j \le K, \tag{1}$$

and $G_k(x) = \max_{j>k} \Delta_{k,j}(x)$ is the largest remaining improvement. A detached head uses existing prefix features to predict a location and scale; a split-conformal correction yields an upper bound valid simultaneously across depths. The policy stops at the earliest depth whose bound is at most a user-chosen ε. Simultaneous coverage supports adaptive depth selection and post-calibration tolerance

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

choice without further multiplicity correction. Bounding remaining gain also permits retaining a better current action when later computation would worsen it.

We prove finite-sample stopping-risk control and derive expected-gain, oracle-depth, critic-error, shift and performance-difference results. Our risk-matched evaluation calibrates seven adaptive-exit baselines by Learn-Then-Test across their full risk ranges, finding a reduction in computation by a factor of two against the best heuristic gate. We characterize a population occupancy-calibration iteration whose fixed point satisfies the return bound's coverage assumption and evaluate an empirical version on CartPole and MinAtar. Finally, a scalar head predicting G_k matches the structured $O(K^2)$ lattice of Equation (1), simplifying the certificate architecture.

2 Related Work

Adaptive Computation Time and PonderNet learn input-dependent halting distributions (Graves, 2016; Banino et al., 2021); BranchyNet and subsequent early-exit networks commonly gate intermediate classifiers by confidence (Teerapittayanon et al., 2016). Jointly learned gates can influence the trajectories they evaluate. Fixed-prior prefix supervision followed by a post-hoc readout can outperform entangled halting objectives in looped transformers (Popescu et al., 2026). RARE^2 adopts this separation but calibrates remaining decision value rather than halting probability.

Weight sharing supports repeated refinement in Universal Transformers (Dehghani et al., 2019), fixed-point computation in deep equilibrium models (Bai et al., 2019), and generalization from easy to harder instances through additional iterations (Schwarzschild et al., 2021; Bansal et al., 2022). Recurrent depth also provides test-time computation for language models (Geiping et al., 2025). Our certificate replaces iteration schedules and latent-convergence tests with a stopping criterion based on decision value.

Computation within policies has been studied through value-iteration networks, the Predictron and neural algorithm execution (Tamar et al., 2016; Silver et al., 2017; Veličković et al., 2020). Ghugare et al. (2026) show that recurrent depth improves online and offline RL, but fix depth within each deployment. RD-VLA uses latent convergence (Tur et al., 2026), ELASTIC learns generative-control compute schedules (Li et al., 2026), and complexity detectors select act, think or abstain modes (Izzo et al., 2026). We complement these approaches by bounding the remaining gain in action-score policies trained from scratch.

Split conformal prediction gives distribution-free marginal coverage under exchangeability (Vovk et al., 2005; Lei et al., 2018). Residual normalization (Lei et al., 2018) and conformalized quantile regression (Romano et al., 2019) adapt bounds to input difficulty. Risk-controlling prediction sets certify general losses (Bates et al., 2021), Learn Then Test uses multiple testing (Angelopoulos et al., 2021), and conformal risk control handles monotone losses (Angelopoulos et al., 2022). We use Learn-Then-Test for every scalar baseline, isolating stopping signals from calibration differences. Whereas Conformal Policy Control regulates changes from a safe reference policy (Prinster et al., 2026), we calibrate counterfactual gains along a fixed internal computation trajectory.

Certified early exiting is the closest related setting. Confident Adaptive Transformers (Schuster et al., 2021) and Confident Adaptive Language Modeling (Schuster et al., 2022) certify agreement with the full model; risk control extends to vision, generation and speculative decoding (Jazbec et al., 2024), and test-time reasoning (Wu et al., 2025). Our control setting differs in three respects. First, distinct actions can have equal utility, making output agreement an unnecessary computational expense; our multiple-optima diagnostic isolates this issue. Second, comparison with every future prefix can justify retaining a better current action when further computation causes overthinking, as reported in feedforward classifiers (Kaya et al., 2019) and observed here on LightsOut and MinAtar. Third, adaptive stopping changes the state distribution under which coverage is needed. Proposition 6 addresses this circular dependence through population occupancy calibration.

3 CERTIFIED REMAINING-GAIN STOPPING

3.1 EXECUTABLE RECURRENT PREFIXES

Let $X \in \mathcal{X}$ be a decision input and $\mathcal{A}$ a finite action set. A weight-tied recurrent policy applies a shared block F_θ at most K times,

$$h_0 = e_\theta(X), \quad h_k = F_\theta(h_{k-1}, e_\theta(X)), \quad z_k = g_\theta(h_k), \quad a_k = \arg\max_a z_k(a). \tag{2}$$

All prefixes are trained using a gate-independent loss: depth-weighted cross-entropy in the behavior-cloning tasks and a Bellman loss at every depth in the fitted-Q task (Ernst et al., 2005). Depth weights are fixed before training and increase linearly with k, which keeps shallow prefixes useful without letting an unstable gate decide which prefixes receive supervision.

A fixed reference utility $q : \mathcal{X} \times \mathcal{A} \to [0, B]$ defines the gains in Equation (1). The theory applies regardless of how q is obtained, but the interpretation of its guarantee depends on that choice: an exact $Q^\star$ yields an optimality statement, Q^{π_K} yields a statement relative to the full-depth policy, and an approximate critic yields a guarantee relative to that critic plus an estimation term. Within the eight-task planning suite, seven tasks use exact shortest-path or puzzle utility and one uses a frozen fitted-Q critic.

3.2 THE CERTIFICATE HEAD

At depth k, the certificate feature vector c_k concatenates detached state and action latents, entropy, the margin between the two highest action scores, maximum action probability, relative latent change, and normalized depth. A shared multilayer perceptron reads $\mathrm{sg}(c_k)$, with gradients stopped at the input, and outputs a location and a positive scale. We consider two instantiations. The structured form additionally takes a learned embedding d_{j-k} of the gap between the current and future depths and predicts every cell of the triangular lattice, $(\hat{\mu}_{k,j}, \hat{\sigma}_{k,j}) = r_\phi(\mathrm{sg}(c_k), d_{j-k})$. The scalar form predicts the remaining gain directly, $(\hat{\mu}_k, \hat{\sigma}_k) = r_\phi(\mathrm{sg}(c_k))$, using K outputs rather than $K(K-1)/2$. Section 5.3 shows that the scalar form suffices, so it is the variant we recommend; the structured form remains useful only if a downstream scheduler needs to know *which* future depth would help rather than whether any would.

We freeze θ, cache the prefix trajectories, and fit ϕ with a Huber location term plus a Laplace negative log-likelihood, sampling zero-gain and positive-gain targets equally. The scale normalizes residuals to improve the efficiency of the conformal step; it is not a calibrated uncertainty estimate, and validity comes from the conformal correction itself.

3.3 SIMULTANEOUS CONFORMALIZATION AND STOPPING

Let $\mathcal{I}$ be the finite index set the stop rule may inspect: $\{(k, j) : k < j\}$ for the structured form and $\{k : k < K\}$ for the scalar form. Write T_i for the target at index i and $(\hat{\mu}_i, \hat{\sigma}_i)$ for its prediction. The predictor is fixed before an exchangeable calibration sample $X_1, \ldots, X_n$ is observed. Define

$$S_m = \max_{i \in \mathcal{I}} \frac{T_i(X_m) - \hat{\mu}_i(X_m)}{\hat{\sigma}_i(X_m)}, \tag{3}$$

let $r = \lceil (n+1)(1-\alpha) \rceil$, and let $\hat{q}_\alpha$ be the rth order statistic of $S_1, \ldots, S_n$, with $+\infty$ if $r > n$. The upper bounds are $U_i(x) = \hat{\mu}_i(x) + \hat{q}_\alpha \hat{\sigma}_i(x)$. For a tolerance $\varepsilon \geq 0$ the exit is

$$\tau_\varepsilon(x) = \min\Big\{k : \max_{j>k} U_{k,j}(x) \leq \varepsilon\Big\} \quad \text{or} \quad \tau_\varepsilon(x) = \min\{k : U_k(x) \leq \varepsilon\}, \tag{4}$$

with $\tau_\varepsilon = K$ when the set is empty. The maximum in Equation (3) is what makes the coverage event simultaneous: calibrating each index separately and then inspecting many of them adaptively would require a further multiplicity correction.

The statistical construction is standard. A normalized maximum-residual score over a finite index set follows the locally weighted conformal literature (Lei et al., 2018; Romano et al., 2019), and calibrated early exit against a full-compute reference follows Schuster et al. (2021) and its successors (Schuster et al., 2022; Jazbec et al., 2024; Wu et al., 2025). Our change is the quantity being certified.

Certifying remaining decision value rather than agreement with the full-compute output avoids computation devoted solely to breaking ties, justifies stopping when further computation would worsen the decision, and raises the occupancy question of Section 5.4.

At deployment the policy computes one recurrent prefix, predicts its remaining gain, adds the shared conformal margin, and stops if the upper bound is at most ε; otherwise it executes the next shared block. Maximum depth is a guaranteed fallback, and changing ε changes only the stop rule, not the model or the calibration quantile.

4 GUARANTEES

Proofs are in Appendix A. Theorem 1 is stated over a general index set, so it covers both instantiations with one argument.

Theorem 1 (Simultaneous coverage)**.** *Assume the trained policy, reference utility, certificate head and preprocessing are fixed independently of the calibration sample, and that $X_1, \dots, X_n, X_{n+1}$ are exchangeable. Then $\Pr\big(T_i(X_{n+1}) \le U_i(X_{n+1})\ \forall i \in \mathcal{I}\big) \ge 1 - \alpha$.*

Corollary 2 (Adaptive stopping and post-hoc tolerances)**.** *Under Theorem 1, for every fixed $\varepsilon \ge 0$, $\Pr\big(\max_{j>\tau_\varepsilon} \Delta_{\tau_\varepsilon, j} > \varepsilon\big) \le \alpha$. Moreover the same coverage event implies the conclusion simultaneously for all $\varepsilon \ge 0$, so the tolerance may be chosen after calibration, including by a compute-budget rule, without a further multiplicity penalty.*

The guarantee controls the improvement available within the learned recurrent trajectory; it establishes global optimality only if q and the prefix family support that conclusion. It does not assert conditional coverage for every individual state.

Proposition 3 (Expected residual gain)**.** *If every gain is bounded by B, then $\mathbb{E}\big[\max_{j>\tau_\varepsilon} \Delta_{\tau_\varepsilon, j}\big] \le (1-\alpha)\varepsilon + \alpha B$.*

Proposition 4 (Oracle-depth sandwich)**.** *Let $\tau^\star_\varepsilon = \min\{k : G_k \le \varepsilon\}$. On a state for which $T_i \le U_i \le T_i + w$ for all i, we have $\tau^\star_\varepsilon \le \tau_\varepsilon \le \tau^\star_{\varepsilon - w}$, with $\tau^\star_t = K$ for $t < 0$. The certified stopping rule therefore never stops earlier than the oracle at the same tolerance, and its excess computation is controlled by the slack in its upper bound.*

Proposition 5 (Approximate critic and shift)**.** *If the calibrated utility $\hat{q}$ satisfies $\|\hat{q} - q\|_\infty \le \eta$, then every true gain is at most the corresponding gain under $\hat{q}$ plus 2η, so Corollary 2 holds with tolerance $\varepsilon + 2\eta$. If the calibration and deployment state laws have total variation distance at most ρ, the violation probability is at most $\alpha + \rho$.*

Suppose next that $q = Q^{\pi_K}$ and the full-depth policy is deterministic. For a fixed adaptive policy π_τ, assume that its simultaneous bounds cover with probability at least $1 - \alpha$ for a state drawn from its normalized discounted occupancy d^{π_τ}. The performance-difference lemma (Kakade & Langford, 2002) and the expected-gain argument in Proposition 3 then give

$$J(\pi_K) - J(\pi_\tau) \le \frac{(1-\alpha)\varepsilon + \alpha B}{1-\gamma}, \tag{5}$$

with an additional $2\eta/(1-\gamma)$ under critic error. If coverage holds jointly over calibration randomness and a state drawn conditionally from the selected policy's discounted occupancy, the corresponding return bound holds in expectation over calibration randomness.

Calibration on the full-compute policy's states does not ensure coverage under the adaptive policy's occupancy. Moreover, the adaptive policy depends on the calibration data, so matching the sampling policy alone does not establish the joint exchangeability required by Theorem 1. To separate this dependence from finite-sample estimation, let $\pi(q)$ denote the adaptive policy with quantile q and define the population operator $\mathcal{Q}(\pi) = \inf\{u : \Pr_{X \sim d^\pi}(S(X) \le u) \ge 1 - \alpha\}$, where S is the fixed score in Equation (3).

Proposition 6 (Population occupancy calibration)**.** *Fix the prefix actions, certificate head and tolerance, with deterministic tie-breaking, in a fixed MDP with a fixed initial-state law, $0 < \alpha < 1$ and $0 < \gamma < 1$. Suppose $0 \le T_i(x) \le B$, $|\hat{\mu}_i(x)| \le \mu_{\max}$ and $\hat{\sigma}_i(x) \ge \sigma_{\min} > 0$ on the reachable state set. For any finite q_0, the population iteration $q_{t+1} = \max\{q_t, \mathcal{Q}(\pi(q_t))\}$ is nondecreasing and bounded above by $\bar{q} = \max\{q_0, (B + \mu_{\max})/\sigma_{\min}\}$. It converges to a fixed point $q^\star$ satisfying*

$\mathcal{Q}(\pi(q^\star)) \le q^\star$, so its simultaneous bounds have coverage at least $1-\alpha$ under $d^{\pi(q^\star)}$. Under the reference-utility assumptions above, Equation (5) *therefore applies to this limiting policy.*

The running maximum prevents decreases in the quantile, eliminating the oscillation observed with the undamped update; Figure 3 in the appendix traces both iterations. The proposition concerns exact population quantiles. Our implementation estimates each quantile from fresh data, and an unchanged empirical update does not establish a population fixed point or a finite-sample coverage guarantee for the selected policy. We therefore evaluate the final policy on fresh held-out states and report its empirical coverage.

5 EXPERIMENTS

We evaluate coverage, computation at matched risk, the scalar-head reduction, empirical recalibration after a policy change, and measured latency. Supporting tables and additional protocol details appear in the appendix. A tolerance sweep also tests Corollary 2: one conformal quantile supports multiple operating points without retraining or recalibration (Table 8).

5.1 PROTOCOL

The planning suite contains five graph-navigation families—cycle with chords, open grid, random maze, tree and geometric graph—plus Cycle-FQI and LightsOut on 3×3 and 4×4 boards: eight tasks with five seeds each, for forty runs, at maximum depths from 6 to 11. Seven tasks use exact optimal-action supervision; Cycle-FQI uses fitted-Q backups with rewards based on exact shortest-path progress. Its frozen deepest critic supplies certificate targets; exact distances enter the rewards and final action-quality evaluation, not direct action-value or certificate targets. Training, calibration and test data are disjoint; each calibration or test graph is generated separately and contributes exactly one randomly sampled decision state, matching the exchangeability unit in Theorem 1. Graph policies use weight-tied message passing and LightsOut uses a weight-tied convolutional block. We also evaluate CartPole and all five MinAtar games (Young & Tian, 2019).

Baselines include maximum depth, an oracle with access to future gains, confidence, margin, entropy, latent convergence, action agreement, and a supervised binary halting head. We call confidence, margin, entropy, convergence and agreement rules *heuristic gates*, distinguishing them from the binary head and the scalar remaining-gain head. Each scalar baseline gate uses fixed-sequence Learn-Then-Test with a one-sided Clopper–Pearson bound at $\delta = 0.05$. We use $\varepsilon = 0.49$ throughout the planning suite: any positive gain violates this tolerance on the seven integer-valued utilities, whereas Cycle-FQI uses the real-valued units of its frozen critic. Primary metrics are mean recurrent depth, simultaneous coverage, future-improvement violation and exact optimal-action rate. CPU measurements use active-batch execution, removing completed states rather than estimating latency from a FLOP proxy.

5.2 COVERAGE AND THE RISK-MATCHED FRONTIER

Across eight planning tasks, simultaneous coverage averages 95.14% at a nominal 95%, while RARE^2 saves 70.2% of recurrent steps with 0.145% realized violation. Task-level coverage means range from 94.68% to 95.94%. Simultaneous coverage requires every index to be covered on the same state before depth selection; realized violation is lower because stopping additionally requires the whole future row to lie below ε. The compute-matched comparison in Table 4 uses different risk levels and therefore cannot establish the main comparison. At approximately matched depth, RARE^2 improves optimal-action rate by 4.10 points and reduces violation by 3.44 points relative to the confidence gate; both paired Wilcoxon tests give $p = 0.0078$. Their task-averaged depths differ by only -0.044 steps ($p = 0.916$), but the confidence gate is calibrated to 5% risk rather than RARE^2's observed 0.145%. We therefore trace compute–risk frontiers on six graph tasks with five seeds, recalibrating scalar baseline gates at risk targets from 5% to 0.3% and RARE^2 at α from 0.2 to 0.01.

At matched violation near 0.05%, RARE^2 requires 2.54 steps against 5.29 for the best heuristic gate, a factor of 2.09. Using the same frozen features and reference utility, the binary head requires 3.81 steps, a factor of 1.50, after reducing remaining gain to a binary label. Entropy, logit margin, latent convergence and three-step agreement require 5.62, 5.78, 7.14 and 9.01 steps, respectively. These

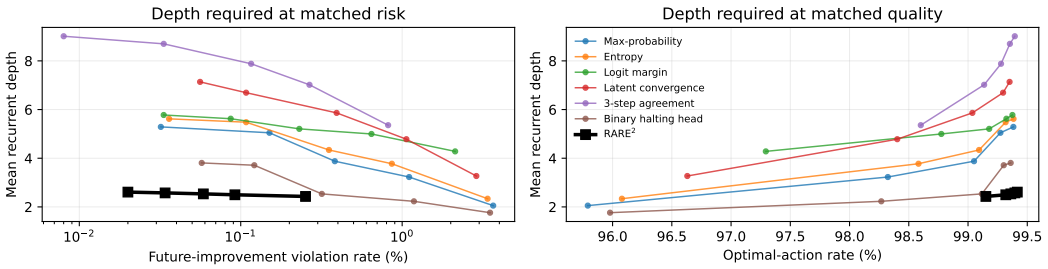


Figure 1: Compute–risk frontiers on six graph tasks and five seeds. Lower is better in both panels. RARE^2 lies below every scalar baseline gate at every risk level, and its own curve is nearly flat, so tightening the risk target adds little computation.

Table 1: Structured lattice versus its scalar reduction, at $\alpha = 0.05$ on all eight tasks (five seeds). The scalar head predicts only G_k, using K outputs instead of $K(K-1)/2$, with identical features, loss, optimizer, step count and conformal procedure. It requires less computation with indistinguishable action quality, so the lattice is not required for the stopping rule reported here.

	Depth ↓		Optimal (%) ↑		Violation (%) ↓	
Task	Lattice	Scalar	Lattice	Scalar	Lattice	Scalar
Cycle-20	1.63	1.62	99.83	99.85	0.08	0.08
Cycle-FQI	1.64	1.60	99.74	99.56	0.06	0.18
Geo-32	2.82	2.71	99.27	99.26	0.03	0.06
Grid-7	3.06	2.97	100.00	99.95	0.00	0.05
LightsOut-3	2.48	2.40	82.61	82.62	0.06	0.11
LightsOut-4	2.85	2.61	97.59	97.55	0.74	0.91
Maze-7	4.35	4.04	97.76	97.68	0.12	0.20
Tree-31	1.71	1.72	99.56	99.57	0.06	0.06
Task mean	2.57	**2.46**	97.05	97.01	0.14	0.21

Paired Wilcoxon over eight task means: depth $p = 0.023$ (favoring scalar); optimal-action rate $p = 0.211$ (not significant).

comparisons concern scalar baseline gates; the recommended scalar remaining-gain head is evaluated separately. Reducing α from 0.2 to 0.01 changes RARE^2 depth from 2.43 to 2.61 while violation falls by more than a factor of ten. Maze-7's 400 calibration units cannot certify baseline risk targets below 0.75%, forcing maximum-depth fallback; excluding this task still gives per-task depth ratios of $1.22\times$ to $2.86\times$ against the confidence gate. Table 6 in the appendix provides the detailed comparison.

5.3 THE SCALAR HEAD AND OVERTHINKING

Our stopping rule needs only $\max_{j>k} U_{k,j}$, so we compare the lattice with a head that predicts G_k directly under otherwise identical training and calibration: the detached features, Huber plus Laplace loss, optimizer, learning rate, update count and class balancing are unchanged. Only the index set changes from $K(K-1)/2$ to K elements. Across eight tasks, the scalar head reaches 97.01% optimal actions against 97.05% for the lattice (paired Wilcoxon $p = 0.211$, paired t $p = 0.135$), while using 2.46 rather than 2.57 steps (paired Wilcoxon $p = 0.023$). We therefore recommend the scalar form on these tasks. The lattice remains useful when a scheduler needs the identity of a helpful future depth or a functional other than the row maximum. Tables 4, 6 and 7 use this more expensive lattice variant; Table 1 reports the controlled scalar comparison.

On LightsOut-3 $\times$ 3, adaptive exit improves optimal-action rate from 77.66% at maximum depth to 82.61% using 2.48 of 6 steps; on LightsOut-4 $\times$ 4, the corresponding rates are 96.23% and 97.59% at 2.85 of 8 steps. Later prefixes can replace a correct action with an incorrect one. Comparing the incumbent with all later prefixes allows stopping to preserve that action even when the final readout is inferior. Because confidence gates do not observe the effect of continuing, they do not

Table 2: MinAtar at the standard 5M-frame budget, five seeds per game. The policy is trained by Q-learning from interaction alone and the reference utility is the frozen deepest Q head of that same policy, so no planning oracle enters the loop. The confidence gate is calibrated by Learn-Then-Test at the same 5% risk target on the same calibration states. Returns are means $\pm$ standard errors of the mean (SEM) over seeds; no per-game difference between RARE^2 and full compute is significant (paired t, smallest $p = 0.13$ on Seaquest).

	RARE^2				Confidence (LTT)		Full compute
Game	Depth	Saved (%)	Coverage (%)	Return	Depth	Return	Return
Asterix	2.97	40.7	95.03 ± 0.11	5.91 ± 1.37	4.56	6.24	5.59 ± 1.47
Breakout	2.30	53.9	94.91 ± 0.10	18.06 ± 2.34	4.08	20.43	18.59 ± 2.90
Freeway	2.00	60.0	95.34 ± 0.17	50.60 ± 1.54	4.84	50.17	50.23 ± 1.25
Seaquest	2.70	46.0	95.58 ± 0.23	3.83 ± 1.01	4.35	3.79	4.19 ± 1.02
Space Invaders	2.60	47.9	95.46 ± 0.09	21.37 ± 1.26	4.60	21.11	21.03 ± 1.29
Mean	**2.52**	**49.7**	**95.26**	**99.2% of full**	4.49	—	100%

directly represent this decision consequence. This overthinking effect is claimed only where observed; LightsOut-3 also has substantial seed variability, documented in Table 5.

5.4 RECALIBRATION UNDER THE ADAPTIVE POLICY

CartPole tests empirical recalibration motivated by Proposition 6. A six-step recurrent DQN uses deterministic 200-step counterfactual rollouts as its reference utility. States are collected after fresh resets at uniformly drawn finite stopping times, with restarts after early termination. This sampling law differs from discounted occupancy, so the experiment does not verify the return bound.

Calibrating on full-compute-policy states and deploying the adaptive policy gives 84.1% coverage against nominal 95%, 3.80% violation and return 239.5 ± 14.5, compared with 309.5 ± 1.3 for full compute. This is a separate run from the original pilot, whose coverage was measured on the calibration distribution (Appendix D). The pilot used 1800 calibration states and 64 episodes; the present run uses 1000 states and 160 episodes. Their reported coverage values, 94.89% and 84.1%, therefore describe different evaluation distributions, not contradictory estimates for the same experiment. One monotone empirical recalibration moves the quantile from 3.41 to 259.35; the next update is unchanged. Fresh-state coverage after this update is 99.7%, with zero violation. A separate held-out replay gives 99.8% with zero violation, return 329.3 ± 2.7 and mean depth 5.72 of 6. An unchanged empirical update is not a population fixed-point certificate or a finite-sample guarantee for the selected policy.

The 4.6% saving is specific to this reference policy; independently trained CartPole policies save more. At this survival-step utility scale, the predicted remaining gain itself constrains early stopping even when the conformal margin is reduced. Partial damping and bisection reduce the quantile but add fewer than two percentage points of recurrent-step savings and lower return. Their empirical coverage checks do not establish a finite-sample selection guarantee (Appendix F).

5.5 MINATAR WITHOUT A PLANNING ORACLE

All five MinAtar games use five seeds and the standard 5M-frame budget. A weight-tied recurrent Q network with $K = 5$ learns solely from environment interaction; its frozen deepest head supplies the reference utility, so every prefix action is evaluated by the same learned reference. No component uses an oracle. The certificate is relative to this critic; a return interpretation additionally requires the stated critic-error and sampling assumptions. Utility is normalized by each game's median action-value spread, making ε dimensionless (Appendix E). Without this normalization, an absolute tolerance can fall below the certificate's scale floor in states with similar action values, preventing stopping for an implementation reason rather than a statistical one.

After empirical recalibration, held-out coverage averages 95.26%, with game means from 94.91% to 95.58% and 1.26% realized violation. Thus coverage is close to nominal rather than at or above it on every game. Mean depth is 2.52 of 5, saving 49.7% of recurrent steps, with savings from 40.7%

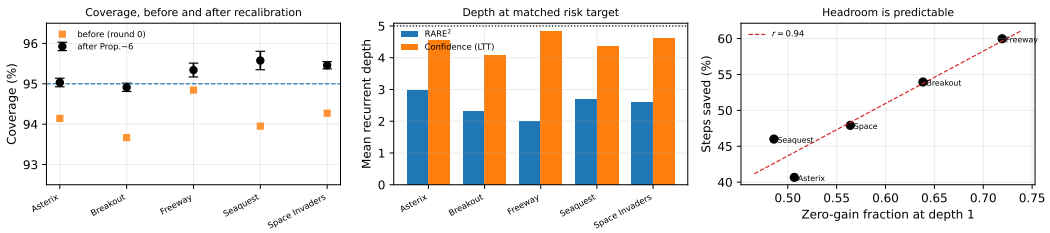


Figure 2: MinAtar at the standard budget, five games and five seeds. Left: held-out coverage before and after the occupancy iteration; round-zero calibration on the full-compute policy's states falls below nominal on nineteen of twenty-five runs. Middle: depth needed by RARE^2 and by a Learn-Then-Test confidence gate at the same 5% risk target. Right: realized savings against the zero-gain fraction measured before calibration.

on Asterix to 60.0% on Freeway. Full-compute returns range from 4.19 to 50.23, so we compare return through log-ratios rather than average per-seed ratios, which are dominated by the weakest reference policies. Across twenty-five runs, the mean log-ratio corresponds to 99.2% return retention, with a 90% confidence interval of $[95.5\%, 103.0\%]$; two one-sided tests establish equivalence within $\pm 5\%$ ($p = 0.041$) and $\pm 10\%$ ($p = 0.0003$). Seaquest retains 91.4%, below full compute in four of five seeds, although the difference is not significant at this sample size. No per-game return-retention difference is significant over five seeds; the smallest paired-t p value is 0.13 on Seaquest.

At the same 5% risk target on the same calibration states, the Learn-Then-Test confidence gate requires mean depth 4.49, a game-averaged factor of 1.82 relative to RARE^2. Per-game depth ratios range from $1.54\times$ on Asterix to $2.42\times$ on Freeway. The depth difference is significant on every game (largest paired-t $p = 0.011$), and the aggregate comparison over game means gives $p = 0.001$. Return differences are not significant on individual games (smallest $p = 0.15$) or over game means ($p = 0.48$), and returns are equivalent within $\pm 10\%$ by two one-sided tests ($p = 0.022$). Table 2 reports per-game results and Appendix E gives the remaining uncertainty and comparison statistics.

Calibration on full-compute-policy states initially gives 94.18% mean coverage, below nominal on nineteen of twenty-five runs. Twenty-one runs require an empirical update, after which mean held-out coverage reaches 95.26%. The largest quantile increase is from 6.56 to 17.19 for a Seaquest seed. These results support empirical policy-dependent recalibration; they do not verify the population iteration's guarantee. The zero-gain fraction at depth one, measured before calibration, correlates with savings at $r = 0.945$ across five games, providing a diagnostic of potential savings.

5.6 Depth and jointly trained gates

A depth sweep retrains at $K \in \{5, 8, 12, 16\}$, with optimizer-update counts proportional to K, so training effort is not held fixed. Cycle-20 savings rise from 68.2% to 89.8% under this schedule, with near-nominal coverage and a zero-gain fraction near 0.886. Non-monotonicity on Cycle-20 remains below 0.1% at every tested K, so these data do not support a general overthinking claim. These results do not isolate depth from training effort (Appendix G). ACT (Graves, 2016) and PonderNet (Banino et al., 2021) provide jointly trained comparators. ACT halts immediately and exceeds the 5% risk target by more than a factor of two; PonderNet requires separate training at each operating point. Neither provides an adjustable risk guarantee, whereas one RARE^2 calibration supports a tolerance range (Appendix H).

5.7 Latency and calibration robustness

Active-batch execution removes completed states and includes certificate overhead. Timings are medians of repeated runs after warm-up with one CPU intra-op thread; the active batch shrinks after each exit. Across six graph tasks, mean CPU speedups are $4.06\times$, $3.91\times$ and $4.91\times$ at batch sizes 1, 32 and 256 (Table 9); these measurements do not establish accelerator speedups. Per-task speedups range from $2.29\times$ to $7.47\times$. On Cycle-20, increasing calibration size from 100 to 2000 changes mean depth by less than 0.003 steps, with mean coverage between 95.45% and 96.63%. Graph-family shift reduces coverage to 91.90% on tree graphs and 90.27% on geometric graphs;

recalibrating only the quantile on 500 target-family states raises coverage to 94.83% and 95.13%, without retraining the policy or certificate head. The appendix gives the complete calibration-size and shift results. The method does not detect shift automatically.

6 Limitations and Conclusion

MinAtar uses small policies and 10×10 binary observations; these results do not establish performance on large-scale robot learning, full Atari, Procgen or OGBench. Seaquest's lower return-retention estimate remains unresolved with five seeds; the aggregate equivalence result should therefore not be interpreted as a resolved per-game guarantee. The experiments establish neither accelerator speedups nor the costs and savings of certification with a large shared encoder. Domains that require deeper decisions should offer smaller savings, which the zero-gain diagnostic helps anticipate. CartPole's 4.6% saving is reference-policy-specific; the tested damping alternatives add little and lower return. Direct calibration of trajectory loss remains untested.

The guarantee is marginal over an exchangeable decision-state distribution and relative to fixed utility q: it provides neither per-state conditional coverage nor protection against critic bias. The control experiments use finite stopping-time sampling rather than discounted occupancy and do not verify the return bound. Deployment requires shift monitoring and incurs certification overhead. Within these limits, calibrated remaining gain reduces recurrent depth relative to the tested scalar baseline gates at matched risk, using the consequence of stopping rather than a confidence heuristic.

Reproducibility Statement

The released package contains dependency-free environment generators, model and calibration code, all forty seed-level JSON files and checkpoints, the scalar-ablation, risk-sweep, tolerance-sweep, occupancy-iteration and MinAtar programs with their raw outputs, aggregation and plotting scripts, measured latency results, unit tests and a single-command smoke test. Dataset-generation seeds and optimizer seeds are separate, and calibration and test states are disjoint. Every number in the paper can be regenerated from stored JSON without retraining, and the risk-matched and ablation results can be regenerated from the stored checkpoints without retraining the policy.

AI-Use Statement

Generative AI systems were used for code scaffolding, language editing, test generation and preliminary literature discovery. The authors are responsible for verifying all mathematical arguments, citations, experimental claims, implementation behavior and final wording. Every bibliographic record was checked against its primary source. All reported numbers were produced using the released executable code and stored result files.

Broader Impact and Ethics

Adaptive computation can reduce latency and energy, but a misleading certificate may encourage premature action in safety-critical systems. The method therefore exposes its assumptions explicitly: calibration is distribution-specific, the utility is only as trustworthy as its source, and marginal risk control is not a per-state safety guarantee. The CartPole appendix documents a deployment in which the guarantee does not apply. Deployment in robotics, healthcare or autonomous systems would require domain-specific utility validation, shift monitoring, a fail-safe maximum-depth fallback, and evaluation of downstream harms beyond computational efficiency.

References

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv:2110.01052*, 2021.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, 2019.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. End-to-end algorithm synthesis with recurrent networks: Logical extrapolation without overthinking. In *Advances in Neural Information Processing Systems*, 2022.

Stephen Bates, Anastasios N. Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6), 2021.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

Raj Ghugare, Michał Bortkiewicz, Alicja Ziarko, and Benjamin Eysenbach. On computation and reinforcement learning. *arXiv preprint arXiv:2602.05999*, 2026.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Riccardo Andrea Izzo, Gianluca Bardaro, and Matteo Matteucci. Act, think or abstain: Complexity-aware adaptive inference for vision-language-action models. *arXiv preprint arXiv:2603.05147*, 2026.

Metod Jazbec, Alexander Timans, Tin Hadži Veljković, Kaspar Sakmann, Dan Zhang, Christian A. Naesseth, and Eric Nalisnick. Fast yet safe: Early-exiting with risk control. In *Advances in Neural Information Processing Systems*, volume 37, pp. 129825–129854, 2024.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, 2002.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International Conference on Machine Learning*, pp. 3301–3310, 2019.

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

Andrew Zou Li, Gokul Swamy, Yonatan Bisk, and Andrea Bajcsy. Elastic: Efficiently learning to adaptively scale test-time compute for generative control policies. *arXiv preprint arXiv:2606.31132*, 2026.

Andrei Cristian Popescu, Haitz Sáez de Ocáriz Borde, and Pietro Liò. Adaptive depth in looped transformers: Diagnosing learned halting gates and trajectory readouts. *arXiv preprint arXiv:2607.20519*, 2026.

Drew Prinster, Clara Fannjiang, Ji Won Park, Kyunghyun Cho, Anqi Liu, Suchi Saria, and Samuel Stanton. Conformal policy control. *arXiv preprint arXiv:2603.02196*, 2026.

Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, 2019.

Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4962–4979, 2021.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17456–17472, 2022.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In *Advances in Neural Information Processing Systems*, 2021.

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, and Thomas Degris. The predictron: End-to-end learning and planning. In *International Conference on Machine Learning*, 2017.

Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. In *Advances in Neural Information Processing Systems*, 2016.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet: Fast inference via early exiting from deep neural networks. In *International Conference on Pattern Recognition*, 2016.

Yalcin Tur, Jalal Naghiyev, Haoquan Fang, Wei-Chuan Tsai, Jiafei Duan, Dieter Fox, and Ranjay Krishna. Recurrent-depth vla: Implicit test-time compute scaling of vision-language-action models via latent iterative reasoning. *arXiv preprint arXiv:2602.07845*, 2026.

Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. Neural execution of graph algorithms. In *International Conference on Learning Representations*, 2020.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Menghua Wu, Cai Zhou, Stephen Bates, and Tommi Jaakkola. Thought calibration: Efficient and confident test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Kenny Young and Tian Tian. Minatar: An atari-inspired testbed for thorough and reproducible reinforcement learning experiments. *arXiv preprint arXiv:1903.03176*, 2019.

A PROOFS

Throughout, $\mathcal{I}$ is a finite index set, T_i is the target at index i, and $(\hat{\mu}_i, \hat{\sigma}_i)$ with $\hat{\sigma}_i > 0$ are the predicted location and scale. For the structured head, $\mathcal{I} = \{(k, j) : 1 \le k < j \le K\}$ with $T_{k,j} = \Delta_{k,j}$; for the scalar head, $\mathcal{I} = \{k : 1 \le k < K\}$ with $T_k = G_k = \max_{j>k} \Delta_{k,j}$. Stating the argument at this level of generality allows one theorem to cover both heads.

A.1 SIMULTANEOUS COVERAGE

Proof of Theorem 1. For a state x, define $S(x) = \max_{i \in \mathcal{I}} (T_i(x) - \hat{\mu}_i(x))/\hat{\sigma}_i(x)$. Because the policy, reference utility, certificate head and preprocessing are all fixed independently of the calibration sample, S is a fixed measurable function, so exchangeability of $X_1, \dots, X_{n+1}$ implies exchangeability of $S_1, \dots, S_{n+1}$. With randomized tie breaking, the rank of S_{n+1} among the $n+1$ scores is uniform, and the standard conservative treatment of ties gives the same lower bound on coverage. Let $r = \lceil (n+1)(1-\alpha) \rceil$ and let $S_{(r)}$ be the rth calibration order statistic, with $S_{(r)} = +\infty$ if $r > n$. The event $\{S_{n+1} \le S_{(r)}\}$ has probability at least $r/(n+1) \ge 1 - \alpha$. On that event, for every $i \in \mathcal{I}$,

$$\frac{T_i(X_{n+1}) - \hat{\mu}_i(X_{n+1})}{\hat{\sigma}_i(X_{n+1})} \le \hat{q}_\alpha,$$

which rearranges to $T_i(X_{n+1}) \le \hat{\mu}_i(X_{n+1}) + \hat{q}_\alpha \hat{\sigma}_i(X_{n+1}) = U_i(X_{n+1})$. A single exchangeable rank statement therefore yields simultaneous coverage of all indices. □

A.2 ADAPTIVE STOPPING AND POST-HOC TOLERANCES

Proof of Corollary 2. Let C be the simultaneous coverage event of Theorem 1. Work first with the structured index set. On C, if $\tau_\varepsilon = k < K$, the stop rule gives $U_{k,j} \le \varepsilon$ for every $j > k$ while coverage gives $\Delta_{k,j} \le U_{k,j}$, so $\max_{j>k} \Delta_{k,j} \le \varepsilon$. If $k = K$, the maximum over the empty set of future indices is zero. Hence the violation event is contained in C^c and has probability at most α. For the scalar index set, the same argument applies with $G_k \le U_k \le \varepsilon$ and $G_k = \max_{j>k} \Delta_{k,j}$ by definition.

The argument does not require a fixed ε: on the single event C, the implication holds for every threshold used in the stop rule, so

$$C \subseteq \bigcap_{\varepsilon \ge 0} \Big\{ \max_{j > \tau_\varepsilon} \Delta_{\tau_\varepsilon, j} \le \varepsilon \Big\}.$$

The tolerance may therefore be any measurable function of the calibration data, a deployment budget, or the observed upper bounds, provided it does not inspect the unobserved true gains of the test point. □

A.3 EXPECTED RESIDUAL GAIN

Proof of Proposition 3. Let $G = \max_{j > \tau_\varepsilon} \Delta_{\tau_\varepsilon, j}$ and $p = \Pr(G > \varepsilon)$, so $p \le \alpha$ by Corollary 2. If $0 \le \varepsilon \le B$, then $\mathbb{E}G \le \varepsilon(1-p) + Bp \le (1-\alpha)\varepsilon + \alpha B$, because $B - \varepsilon \ge 0$ and $p \le \alpha$. If $\varepsilon > B$, then $G \le B$ almost surely and $B \le (1-\alpha)\varepsilon + \alpha B$, so the bound remains valid. □

A.4 ORACLE-DEPTH SANDWICH

Proof of Proposition 4. For the state under consideration, $U_i \ge T_i$ for all i, so whenever the certified rule stops at k, we have $G_k \le \max_{j>k} U_{k,j} \le \varepsilon$, and the earliest oracle-feasible depth cannot be later: $\tau^\star_\varepsilon \le \tau_\varepsilon$. Conversely, if $G_k \le \varepsilon - w$, then $U_{k,j} \le \Delta_{k,j} + w \le \varepsilon$ for every $j > k$, so the certified rule is feasible by depth k. Applying this at $k = \tau^\star_{\varepsilon - w}$ gives $\tau_\varepsilon \le \tau^\star_{\varepsilon - w}$. □

A.5 CRITIC ERROR AND TOTAL-VARIATION SHIFT

Proof of Proposition 5. For any two actions a, a', uniform critic error gives

$$q(x, a') - q(x, a) \le \hat{q}(x, a') + \eta - \hat{q}(x, a) + \eta = \hat{q}(x, a') - \hat{q}(x, a) + 2\eta,$$

and taking positive parts preserves the inequality up to the same additive term, so every true gain is bounded by its $\hat{q}$ counterpart plus 2η. On the coverage event established by calibration, a certified gain under $\hat{q}$ of at most ε therefore implies a true gain of at most $\varepsilon + 2\eta$.

For the shift claim, let the calibration sample have law P^n, let the deployment point have law Q, and let A be the joint event that the calibrated adaptive exit violates its tolerance on that deployment point. Under the exchangeable reference law P^{n+1}, Corollary 2 gives $P^{n+1}(A) \le \alpha$. The two joint laws differ only in their final coordinate, so $\|P^n \times Q - P^{n+1}\|_{\mathrm{TV}} \le \|P - Q\|_{\mathrm{TV}} \le \rho$, and the variational characterization of total variation gives $(P^n \times Q)(A) \le P^{n+1}(A) + \rho \le \alpha + \rho$. □

A.6 Population occupancy calibration

Proof of Proposition 6. The score is uniformly bounded above: since $T_i - \hat{\mu}_i \le B + \mu_{\max}$ and $\hat{\sigma}_i \ge \sigma_{\min} > 0$, every positive normalized residual is at most $(B+\mu_{\max})/\sigma_{\min}$, and every nonpositive residual satisfies the same bound. Thus $\mathcal{Q}(\pi) \le (B + \mu_{\max})/\sigma_{\min}$ for every policy. The running-maximum update is nondecreasing and cannot exceed $\bar{q} = \max\{q_0, (B + \mu_{\max})/\sigma_{\min}\}$, so $q_t \uparrow q^\star < \infty$.

We next show that the induced discounted occupancies converge in total variation. For any fixed state, each stopping inequality has the form $\hat{\mu}_i + q\hat{\sigma}_i \le \varepsilon$. Along $q_t \uparrow q^\star$, its truth value eventually equals its truth value at $q^\star$: an inequality that holds at the limit holds throughout, whereas one that fails at the limit fails for all sufficiently large t. Because the index set is finite, the exit depth, and hence the selected action, eventually agrees with that of $\pi(q^\star)$ at each state. The policies share a fixed initial-state law and transition kernel. At each step, the total-variation distance between the next-state laws is bounded by the previous distance plus the probability, under the limiting state law, that the policies select different actions. Induction and dominated convergence therefore give total-variation convergence of every finite-step state law. Truncating the normalized discounted occupancy at any finite horizon and then bounding its remaining mass by the geometric tail gives $\|d^{\pi(q_t)} - d^{\pi(q^\star)}\|_{\mathrm{TV}} \to 0$.

By the definition of the population quantile and the update, $\Pr_{X \sim d^{\pi(q_t)}}(S(X) \le q_{t+1}) \ge 1-\alpha$. Since $q_{t+1} \le q^\star$, the same inequality holds with threshold $q^\star$. Total-variation convergence allows us to pass this fixed event to the limit, yielding $\Pr_{X \sim d^{\pi(q^\star)}}(S(X) \le q^\star) \ge 1 - \alpha$. Hence $\mathcal{Q}(\pi(q^\star)) \le q^\star$, so $q^\star$ is a fixed point and all upper bounds cover simultaneously under its own discounted occupancy. The return bound then follows under its stated reference-utility assumptions. □

The result concerns a deterministic population operator, not empirical quantiles recomputed from finite samples. A sample quantile that falls below the current running maximum leaves an empirical update unchanged without proving that the current policy has population coverage at level $1 - \alpha$. Independent held-out evaluation measures that policy's coverage; its empirical coverage rate alone is not a finite-sample certification procedure.

A.7 Discounted-return corollary

Let π_K be the deterministic maximum-depth policy and π_τ the adaptive policy. The performance-difference lemma gives

$$J(\pi_\tau) - J(\pi_K) = \frac{1}{1-\gamma}\mathbb{E}_{s \sim d^{\pi_\tau}}\left[Q^{\pi_K}(s, \pi_\tau(s)) - V^{\pi_K}(s)\right] \tag{6}$$

$$= \frac{1}{1-\gamma}\mathbb{E}_{s \sim d^{\pi_\tau}}\left[Q^{\pi_K}(s, \pi_\tau(s)) - Q^{\pi_K}(s, \pi_K(s))\right]. \tag{7}$$

Negating and upper bounding the signed difference by its positive part yields

$$J(\pi_K) - J(\pi_\tau) \le \frac{1}{1-\gamma}\mathbb{E}_{d^{\pi_\tau}}\left[Q^{\pi_K}(s, \pi_K(s)) - Q^{\pi_K}(s, \pi_\tau(s))\right]_+.$$

Suppose the fixed deployed policy has simultaneous coverage at level $1 - \alpha$ under d^{π_τ} and every remaining gain is bounded by B. The positive action-value gap is at most the remaining gain, because the maximum-depth action is one of the future prefix actions whenever stopping occurs before maximum depth; at maximum depth the gap is zero. The argument in Proposition 3 therefore gives

Equation (5); uniform critic error adds at most 2η inside the expectation. For a random policy selected through calibration, the same argument gives a bound on $\mathbb{E}[J(\pi_K) - J(\pi_\tau)]$ if coverage holds jointly over calibration and a state drawn conditionally from the selected policy's discounted occupancy. A marginal split-conformal statement does not imply a return guarantee conditional on every realized calibration sample. Calibration on an arbitrary replay distribution, or empirical stabilization of a policy-dependent quantile, does not by itself establish the required coverage under the deployed policy's state law.

B Experimental Details

B.1 Task construction

Each Cycle-20 state specifies a randomly relabeled 20-node cycle with sparse long-range chords, a current node and a goal; the distances range from 1 to 7 during training and from 1 to 10 during evaluation. All neighbors that reduce the shortest-path distance receive equal target probability. Cycle-FQI uses the same graph family, but action-score heads are trained using fitted-Q backups: each transition is generated by uniformly sampling a valid neighboring action, the reward is shortest-path progress plus a terminal bonus, the discount is 0.97, and a slowly updated target network supplies the deepest-prefix bootstrap. All recurrent prefixes receive Bellman loss. After Bellman training, the deepest fitted-Q head is frozen and evaluates every prefix action for certificate training and calibration. Exact distances enter the Bellman updates through the reward and are used for final optimal-action evaluation, but are not supplied directly as action-value or certificate targets.

OpenGrid uses randomly relabeled 7×7 lattices and therefore contains many equally optimal actions. Maze begins from the same lattice and removes edges while preserving connectivity. Tree uses random recursive trees with a few distractor chords, and GeoGraph connects points through a symmetric nearest-neighbor graph. These families vary path multiplicity, bottlenecks and graph diameter. For LightsOut on 3×3 and 4×4 boards, an action toggles one cell and its orthogonal neighbors; we enumerate the reachable state graph from the all-zero goal and compute exact shortest distances and optimal first actions by breadth-first search. The convolutional recurrent block has the same executable-prefix and detached-certificate interfaces as the graph policy.

Table 3: Primary task hyperparameters. All results use five optimizer seeds. Hidden denotes the node embedding dimension for graph tasks and the convolutional channel count for LightsOut.

Task	Nodes/cells	Hidden	K	Train/cal/test	Reference utility
Cycle-20	20	40	8	8000/1000/2500	exact
Cycle-FQI	20	40	8	11000/1000/2500	frozen fitted-Q critic
Geo-32	32	44	9	7500/1000/2500	exact
Grid-7	49	48	10	7000/1000/2500	exact
Maze-7	49	52	11	8000/400/1000	exact
Tree-31	31	44	9	7500/1000/2500	exact
LightsOut-3	9	32	6	6000/1600/2800	exact
LightsOut-4	16	48	8	8000/2000/3500	exact

B.2 Architectures and optimization

Graph node inputs are a current-node indicator, a goal indicator, normalized degree and a validity mask. A gated mean-aggregation block with layer normalization is reused at every depth, and the action head scores each candidate neighbor using the concatenation of its hidden state and the current node's hidden state. LightsOut uses a 3×3 convolutional residual block with 32 channels on 3×3 boards and 48 channels on 4×4 boards. The parameter count is independent of K except for the small future-depth embedding in the structured variant, which the scalar variant does not require.

Policy optimization uses AdamW, gradient clipping, and a normalized linear depth prior from 0.25 to 1. After the policy is frozen, the certificate is fitted using the loss

$$\mathcal{L}_{\text{cert}} = \text{Huber}(\hat{\mu}, T) + 0.15\Big(\frac{|T - \hat{\mu}|}{\hat{\sigma}} + \log \hat{\sigma}\Big),$$

Table 4: RARE^2 against the compute-matched maximum-probability gate. Task means over five seeds. The confidence gate is calibrated by Learn-Then-Test at a 5% risk target; the two methods achieve indistinguishable average depths (paired difference -0.044 steps, $p = 0.916$), making this a compute-matched rather than a risk-matched comparison. Table 6 gives the risk-matched version.

	RARE^2			Max-probability		
Task	Depth	Viol. (%)	Optimal (%)	Depth	Viol. (%)	Optimal (%)
Cycle-20	1.63	0.08	99.83	1.58	3.66	96.26
Cycle-FQI	1.64	0.06	99.74	2.08	3.74	96.07
Geo-32	2.82	0.03	99.27	2.56	3.50	95.80
Grid-7	3.06	0.00	100.00	1.83	3.95	96.05
LightsOut-3	2.48	0.06	82.61	4.29	2.57	75.86
LightsOut-4	2.85	0.74	97.59	4.31	4.03	93.01
Maze-7	4.35	0.12	97.76	3.18	3.44	94.52
Tree-31	1.71	0.06	99.56	1.07	3.74	95.99
Task mean	**2.57**	**0.14**	**97.05**	2.61	3.58	92.95

with positive and zero targets class-balanced during training. Conformalization, not the raw scale, supplies validity. The scalar ablation of Section 5.3 uses the same trunk width, the same loss, the same learning rate, the same number of gradient steps and the same class balancing, and differs only in dropping the future-depth embedding and predicting one value per prefix.

B.3 WHY CONFIDENCE FAILS ON MULTIPLE OPTIMA

Suppose a_1 and a_2 are equally optimal and a prefix assigns probability $1/2$ to each. Maximum confidence is $1/2$ and entropy is one bit, yet the greedy action has zero regret, so any confidence threshold above $1/2$ spends additional computation only to break an irrelevant tie. Conversely, an overconfident prefix can assign probability 0.99 to an inferior action. OpenGrid intentionally contains many such ties, and RARE^2 achieves an optimal-action rate of 100% using 3.06 of 10 steps on this task.

Two further design choices affect reproducibility. The certificate receives detached features and is trained after policy convergence; this separation also affects learning: if the gate loss influences the recurrent trajectory, the network can make halting easier rather than improving actions, and an input-dependent gate can suppress supervision of some prefixes. Separately, replacing normalized residuals with additive residuals preserves validity but increases mean depth in pilot runs, because one global additive margin is set by the hardest index; the learned scale is therefore an efficiency device whose raw values are never interpreted as probabilities.

C ADDITIONAL RESULTS

C.1 COMPUTE-MATCHED COMPARISON

Average simultaneous coverage is 95.14% at a nominal level of 95%, with task means from 94.68% to 95.94% (Figure 5). The coverage event requires every index to be covered on the same state before the depth is selected. The realized violation rate at the selected depth is much smaller, averaging 0.145%, because the stop rule requires the whole future row to lie below ε.

Table 4 reports the compute-matched comparison that the early-exit literature usually makes. RARE^2 uses 29.8% of maximum recurrent depth, saving 70.2% of steps, while the confidence gate calibrated at a 5% risk target saves 67.7%; their task-averaged depths are indistinguishable, with a paired difference of -0.044 steps across eight task means ($p = 0.916$). At this matched compute level, RARE^2 improves the optimal-action rate by 4.10 points and lowers the violation rate by 3.44 points; both differences are significant under paired t and Wilcoxon tests ($p = 0.0078$ for each Wilcoxon test). The two methods operate at different risk levels here, 0.145% violation versus 5%, so Section 5.2 presents the main comparison at matched risk.

C.2 SCALAR-HEAD ABLATION

Our stop rule uses only $\max_{j>k} U_{k,j}$, motivating a controlled comparison with a head that predicts the scalar G_k directly. This head uses the same detached features, Huber plus Laplace loss, optimizer, learning rate, step count, class balancing and conformal calibration; the only difference is that the index set of Equation (3) has K elements instead of $K(K-1)/2$.

The scalar head matches the lattice. Across eight tasks it reaches 97.01% optimal actions compared with the lattice's 97.05%, a difference that is not significant (paired Wilcoxon $p = 0.211$, paired t $p = 0.135$), while using 2.46 steps compared with 2.57; this reduction in depth is significant in favor of the scalar head (paired Wilcoxon $p = 0.023$). On these tasks the structured lattice is therefore not required, and the scalar form is the variant we recommend; the contribution is the certified remaining-gain stopping rule rather than the lattice architecture. The lattice remains the more general representation, preferable when a scheduler needs to know which future depth would help or when the relevant functional of the future row is not its maximum. Tables 4, 6 and 7 report the lattice variant as originally run, so every comparison above uses the more expensive of the two heads.

C.3 PER-TASK RESULTS WITH UNCERTAINTY

Table 5 reports per-task means with standard errors of the mean (SEM) over five optimizer seeds, providing uncertainty estimates omitted from the aggregate results in the main text. Two entries deserve comment. LightsOut-3 has by far the greatest variability (± 6.00 points of optimal-action rate) because its reachable state graph is small enough that differences across seeds in which states are difficult dominate; the overthinking result of Section 5.3 should be interpreted in light of this variability. Maze-7 has only 400 calibration units compared with 1000 elsewhere, which prevents it from certifying a risk target below 0.75% in Table 6.

Table 5: Per-task RARE^2 results, mean $\pm$ SEM over five seeds. Savings are relative to maximum recurrent depth. A violation means some later prefix would improve the chosen action by more than ε.

Task	Coverage (%)	Depth	Saved (%)	Optimal (%)
Cycle-20	94.68 ± 0.41	1.626 ± 0.006	79.67 ± 0.08	99.83 ± 0.03
Cycle-FQI	94.93 ± 0.32	1.643 ± 0.040	79.46 ± 0.51	99.74 ± 0.07
Geo-32	95.17 ± 0.28	2.823 ± 0.043	68.63 ± 0.48	99.27 ± 0.11
Grid-7	95.02 ± 0.34	3.058 ± 0.040	69.42 ± 0.40	100.00 ± 0.00
LightsOut-3	95.70 ± 0.19	2.480 ± 0.139	58.66 ± 2.32	82.61 ± 6.00
LightsOut-4	94.81 ± 0.31	2.852 ± 0.083	64.35 ± 1.04	97.59 ± 0.40
Maze-7	94.90 ± 0.81	4.352 ± 0.108	60.44 ± 0.98	97.76 ± 0.39
Tree-31	95.94 ± 0.20	1.710 ± 0.034	81.00 ± 0.38	99.56 ± 0.18

C.4 DETAILED RISK-MATCHED COMPARISON

To remove the operating-point confound, we recalibrate every scalar baseline gate by Learn-Then-Test at risk targets from 5% down to 0.3%, and RARE^2 at α from 0.2 down to 0.01, on the six graph tasks with five seeds. This traces each method's full compute–risk frontier rather than a single operating point.

At a matched violation rate near 0.05%, RARE^2 needs 2.54 recurrent steps while the best heuristic gate needs 5.29, a factor of 2.09; entropy needs 5.62, logit margin 5.78, latent convergence 7.14 and three-step agreement 9.01. Per-task ratios against the confidence gate range from $1.22\times$ to $2.86\times$. The binary halting head, which uses the same reference utility as RARE^2 but reduces it to one label, needs 3.81 steps, a factor of 1.50 relative to RARE^2; this isolates the additional computation required when the magnitude of the remaining gain is discarded. RARE^2 is also remarkably insensitive to its risk parameter: reducing α from 0.2 to 0.01 changes depth from 2.43 to 2.61, an increase of about 7%, while the violation rate falls by more than a factor of ten. Below a 0.75% risk target, Maze-7's 400 calibration units do not suffice for certification, so every scalar baseline gate falls back to maximum depth, increasing the depths in the corresponding rows; when Maze-7 is excluded, the advantage remains $1.22\times$ to $2.86\times$.

Table 6: Risk-matched comparison on the six graph tasks (five seeds). Every scalar baseline gate is recalibrated by Learn-Then-Test across a range of risk targets and RARE^2 across a range of α, showing the full compute–risk frontier rather than a single operating point. At comparable low violation rates, the best heuristic gate requires roughly twice the recurrent depth of RARE^2, whereas the binary halting head requires 1.50 times its depth. † marks rows in which Maze-7 cannot certify at the target with its 400 calibration units and falls back to maximum depth, which increases the reported mean depths of the scalar baseline gates.

Method	Target	Depth ↓	Violation (%) ↓	Optimal (%) ↑
RARE^2	$\alpha = 0.2$	**2.43**	**0.252**	**99.15**
	$\alpha = 0.1$	**2.50**	**0.092**	**99.32**
	$\alpha = 0.05$	**2.54**	**0.059**	**99.36**
	$\alpha = 0.02$	**2.58**	**0.034**	**99.40**
	$\alpha = 0.01$	**2.61**	**0.020**	**99.42**
Max-probability	0.05	2.05	3.664	95.79
	0.02	3.23	1.105	98.32
	0.01	3.88	0.383	99.05
	0.005†	5.05	0.151	99.27
	0.003†	5.29	0.032	99.38
Entropy	0.05	2.34	3.382	96.08
	0.02	3.77	0.863	98.58
	0.01	4.34	0.352	99.09
	0.005†	5.48	0.108	99.32
	0.003†	5.62	0.036	99.38
Logit margin	0.05	4.28	2.129	97.29
	0.02	5.00	0.645	98.77
	0.01	5.21	0.231	99.18
	0.005†	5.62	0.087	99.32
	0.003†	5.78	0.033	99.37
Latent convergence	0.05	3.27	2.881	96.63
	0.02	4.78	1.062	98.40
	0.01	5.86	0.393	99.04
	0.005†	6.69	0.108	99.30
	0.003†	7.14	0.056	99.35
2-step agreement	0.05	5.81	1.061	98.36
	0.02	7.97	0.323	99.06
	0.01	9.17	0.000	99.41
	0.005†	9.17	0.000	99.41
	0.003†	9.17	0.000	99.41
3-step agreement	0.05	5.36	0.819	98.60
	0.02	7.02	0.267	99.14
	0.01	7.88	0.116	99.28
	0.005†	8.70	0.033	99.35
	0.003†	9.01	0.008	99.39
Binary halting head	0.05	1.76	3.504	95.98
	0.02	2.23	1.183	98.27
	0.01	2.54	0.318	99.12
	0.005†	3.71	0.121	99.30
	0.003†	3.81	0.057	99.36

C.5 ALL COMPUTED BASELINES

Table 7 reports every evaluated gate, including those omitted from the main text for space. The task scope differs between rows: three-step agreement is defined only on the six graph tasks, so its mean is computed over those six tasks, while every other row averages over all eight. The rows are therefore not directly comparable. On the same six graph tasks, RARE^2 achieves an optimal-action rate of

Table 7: All baselines evaluated in the study, with task means over five seeds. All scalar gates use the same Learn-Then-Test calibration at a 5% risk target with a one-sided Clopper–Pearson bound at $\delta = 0.05$. Three-step agreement is defined only on the six graph tasks, so its mean is over those tasks and is not directly comparable with the eight-task rows.

Method	Depth ↓	Violation (%) ↓	Optimal (%) ↑
RARE[2]	**2.57**	**0.14**	**97.05**
Max-probability	2.61	3.58	92.95
Entropy	2.95	3.32	93.28
Logit margin	4.33	2.49	94.03
Latent convergence	3.53	3.13	93.74
2-step agreement	6.11	0.79	95.51
3-step agreement*	5.35	0.83	98.59
Binary halting head	2.06	3.68	93.41
Maximum depth	8.62	0.00	96.29
Oracle (not deployable)	1.32	0.00	98.02

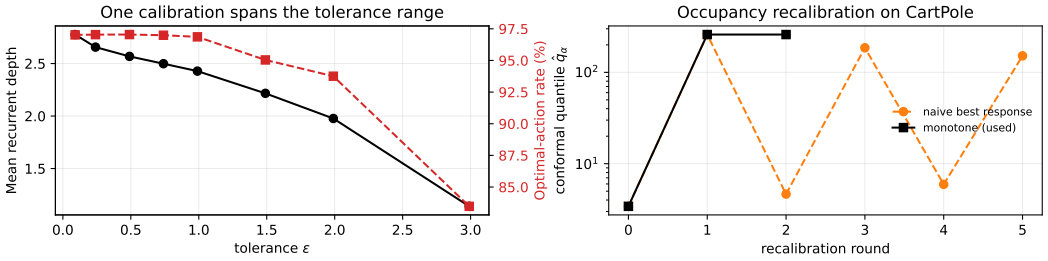


Figure 3: Left: one conformal quantile spans the tolerance range, trading depth against action quality without recalibration. Right: the undamped best-response iteration oscillates between aggressive and conservative quantiles, while the empirical monotone update motivated by Proposition 6 stabilizes after a single step.

99.36% at depth 2.54, compared with 98.59% at depth 5.35 for three-step agreement. The oracle row observes true future gains and is not deployable; it is included to bound the portion of the remaining gap attributable to certificate slack rather than policy limitations.

C.6 One calibration, many tolerances

Corollary 2 establishes that the coverage event holds simultaneously for every ε, so the operating point can be changed after calibration. Table 8 illustrates this property: a single conformal quantile per run, computed once at $\alpha = 0.05$, is reused across the whole tolerance grid. The proportion of steps saved rises from 67.6% at $\varepsilon = 0.09$ to 76.9% at $\varepsilon = 1.99$, while the violation rate rises from 0.11% to 1.62% and remains far below the 5% target. Over the same range, the optimal-action rate gradually decreases from 97.02% to 93.75%. No retraining, relabeling or recalibration is required across these operating points, which distinguishes this approach in practice from a binary halting classifier trained against one fixed threshold.

C.7 Tolerance sweep

Table 8 reports a sweep of the deployment tolerance using a single conformal quantile per run, computed once at $\alpha = 0.05$ and reused throughout, thereby empirically illustrating Corollary 2. The violation rate in each row is measured against the same ε that defines the gate, so these rates are not directly comparable across rows: a loose tolerance makes the violation criterion nearly vacuous, which explains why $\varepsilon = 2.99$ yields a 0% violation rate alongside an optimal-action rate of only 83.5%. The usable range on these integer-valued utilities is roughly $\varepsilon \in [0.09, 1.99]$, over which the proportion of steps saved rises from 67.6% to 76.9% and the optimal-action rate decreases by three points.

Table 8: One calibration, many tolerances. Each run uses a single conformal quantile computed once at $\alpha = 0.05$; only the deployment tolerance ε changes. Task means over eight tasks and five seeds. This illustrates Corollary 2: the operating point can be changed after calibration without recalibrating, retraining or relabeling. Violation is measured against the same ε that defines the gate, so the largest tolerance makes the violation criterion nearly vacuous even as action quality degrades; for this reason, $\varepsilon = 2.99$ is reported but not recommended.

ε	Depth	Steps saved (%)	Violation (%)	Optimal (%)
0.09	2.78	67.64	0.106	97.02
0.24	2.66	69.12	0.104	97.04
0.49	**2.57**	**70.20**	**0.145**	**97.05**
0.74	2.50	71.03	0.206	96.99
0.99	2.43	71.87	0.340	96.86
1.49	2.22	74.35	0.804	95.04
1.99	1.98	76.92	1.620	93.75
2.99	1.14	86.02	0.000	83.47

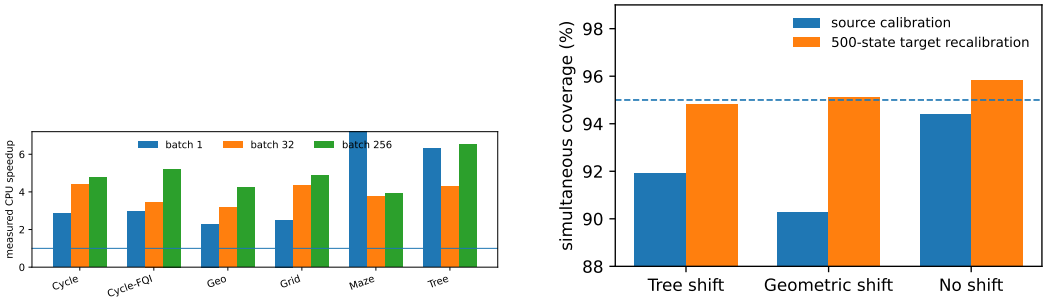


Figure 4: Left: measured active-batch CPU speedup over always executing maximum depth, certificate overhead included. Right: source calibration loses coverage under graph-family shift, while a small target-domain recalibration largely restores it.

C.8 Measured latency

On the six graph tasks, active execution obtains mean CPU speedups of $4.06\times$, $3.91\times$ and $4.91\times$ at batch sizes 1, 32 and 256, with per-task speedups from $2.29\times$ to $7.47\times$ and certificate overhead included (Figure 4 and Table 9). The superlinear relation between average depth and latency at larger batches comes from avoiding repeated message passing on completed states.

We measure latency using PyTorch on a CPU with one intra-op thread for consistent comparisons. These measurements characterize single-threaded CPU execution and do not establish accelerator speedups. The ratio of encoder cost to recurrent-block cost determines how much a reduction in recurrent steps reduces end-to-end latency: on these tasks, the recurrent block dominates, so the two reductions are similar, but they would differ in an architecture with a computationally expensive shared encoder. Measuring accelerator latency across different ratios of encoder cost to recurrent-block cost is a natural direction for future systems work and is beyond the scope of the present study. Each measurement includes the recurrent block, the action head and one certificate evaluation per active prefix, and the batch shrinks after each exit. We report the median of repeated runs after warm-up, and the raw JSON files also include standard deviations.

C.9 Calibration sample size

On Cycle-20, repeated subsampling with calibration sizes from 100 to 2000 yields mean simultaneous coverage between 95.45% and 96.63%, with mean depth changing by less than 0.003 steps, so the learned normalizer already ranks residual difficulty well enough that larger n provides little additional efficiency here.

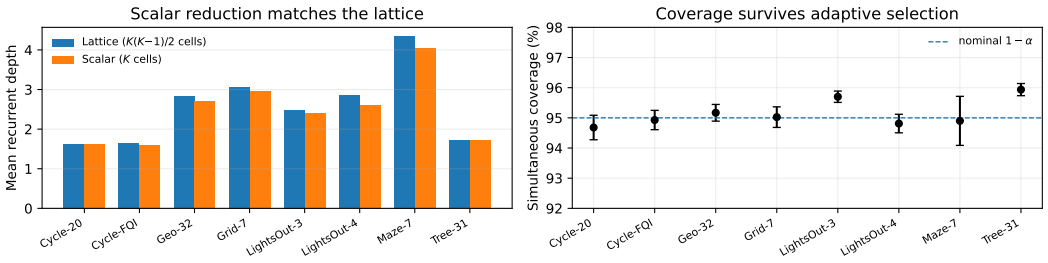


Figure 5: Left: the scalar reduction matches the structured lattice on every task. Right: simultaneous coverage after adaptive depth selection, with SEM over five seeds.

Table 9: Median measured CPU speedup of active RARE^2 execution over maximum depth. Certificate overhead is included.

Task	Batch 1	Batch 32	Batch 256
Cycle-20	2.87×	4.41×	4.77×
Cycle-FQI	2.94×	3.46×	5.21×
Geo-32	2.29×	3.17×	4.24×
Grid-7	2.51×	4.35×	4.86×
Maze-7	7.47×	3.77×	3.90×
Tree-31	6.29×	4.30×	6.51×
Mean	4.06×	3.91×	4.91×

The small non-monotone fluctuations around 95% are expected finite-sample variation. The guarantee applies separately to each randomly chosen calibration split, and averaging many splits is only a visualization of efficiency rather than part of the procedure.

C.10 Distribution-shift details

Exchangeability across graph families does not hold: applying a Cycle-20 quantile to tree or geometric graphs reduces coverage to 91.90% and 90.27%. Recalibrating only the scalar quantile on 500 target-family states raises coverage to 94.83% and 95.13%, without retraining the policy or the certificate head. This is consistent with Proposition 5: the method does not detect distribution shift automatically, but its statistical layer can be recalibrated at low cost.

Cycle-trained certificate predictions were evaluated without retraining on tree and geometric graphs of compatible size. Source calibration lost two to three coverage points. Replacing only the conformal quantile with one estimated from 500 target states restored nominal or conservative coverage, and the no-shift control moved from 94.40% to 95.83%, within ordinary sampling variation. These results support low-cost recalibration rather than invariance to distribution shift.

D CartPole Stress Test

We implemented a dependency-free vectorized CartPole and trained a six-step recurrent DQN. To avoid falsely treating correlated trajectory frames as exchangeable, each calibration and test state came from a fresh environment reset and an independently sampled random stopping time. The reference utility was a deterministic 200-step counterfactual rollout for each first action followed by the frozen maximum-depth policy.

When evaluated on the calibration distribution, the original pilot appeared reliable: the certificate attained 94.89% simultaneous coverage and observed future-gain violations were zero. Over 64 complete episodes, however, RARE^2 used 4.60 of 6 steps and achieved a return of 235.0 ± 23.4, compared with 308.9 ± 1.9 for maximum depth, 100.5 ± 1.1 for a confidence gate and 111.6 ± 3.0 for a binary halting head.

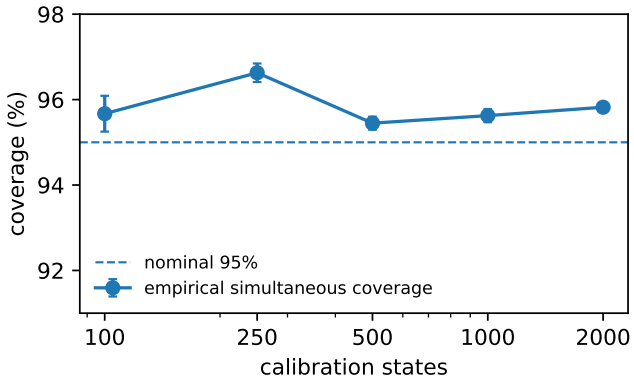


Figure 6: Cycle-20 simultaneous coverage under repeated calibration subsampling. Error bars are SEM across 20 subsamples.

Table 10: Self-consistent occupancy calibration on CartPole. Round 0 is the configuration of a naive deployment: the quantile is calibrated on states visited by the maximum-depth policy and the adaptive policy is then deployed. Coverage on the distribution induced by the adaptive policy is 84.1% against a nominal level of 95%, and the policy loses 70 points of return. One monotone recalibration step yields an empirically stable quantile with high held-out coverage and return exceeding that of full compute. Coverage is measured under the implemented finite stopping-time sampling law, not the discounted occupancy used in the return bound. Episodic returns are means $\pm$ SEM over 160 complete episodes; the maximum-depth policy achieves a return of 309.5 ± 1.3 at depth 6.00.

Round	Quantile $\hat{q}_\alpha$	Coverage on sampled states (%)	Violation (%)	Depth	Episodic return
0 (before recalibration)	3.41	84.1	3.80	4.51	239.5 ± 14.5
1 (stable update)	259.35	99.7	0.00	5.72	328.0 ± 2.8
Held-out replay	259.35	99.8	0.00	5.72	329.3 ± 2.7
Maximum depth	—	—	0.00	6.00	309.5 ± 1.3

Round 0 of Table 10 illustrates deployment without occupancy matching. It uses the same construction as the pilot described above but reports a different run: the pilot used 1800 calibration states and 64 episodes, whereas the run in this table uses 1000 and 160, respectively, so the two report different values for the same failure (94.89% versus 84.1% coverage, 235.0 versus 239.5 return). The coverage discrepancy reflects the evaluation distribution: the pilot measured coverage on the maximum-depth distribution used for calibration, whereas this table measures coverage on the distribution induced by the adaptive policy. Calibrating on the reference policy's states and deploying the adaptive policy yields a certificate that appears valid on the calibration distribution but fails under deployment, with 84.1% coverage against a nominal level of 95%, a 3.80% violation rate, and return falling to 239.5 ± 14.5 compared with 309.5 ± 1.3 for full compute. This illustrates the distribution mismatch discussed in Proposition 5; checks on the original calibration distribution do not assess coverage after the policy changes.

One monotone recalibration step restores empirical coverage. Recalibrating only the scalar quantile, using fresh states sampled under the deployed policy, moves $\hat{q}_\alpha$ from 3.41 to 259.35, and the next update leaves it unchanged, so the empirical iteration stabilizes after a single step. Coverage on fresh states sampled under the resulting policy is 99.7%, the violation rate is zero, and a held-out replay confirms 99.8% coverage. Episodic return is 329.3 ± 2.7, exceeding the full-compute return of 309.5 ± 1.3, at a mean depth of 5.72 out of 6. The compute saving on this seed is only 4.6%, far from the 70% of the planning tasks, because a 200-step survival utility leaves very little certifiable slack; the independently trained policies in Appendix F save substantially more, so this saving is specific to this reference policy rather than to the method. The limit of the population iteration guarantees coverage but can be conservative; we therefore test whether reducing the empirical quantile improves the saving. Appendix F compares it against two alternatives that can decrease the quantile: partial

damping with an empirical coverage guard, and bisection, which accepts a candidate quantile only if measured coverage on fresh states sampled under that candidate meets the nominal level. Bisection brings coverage to the nominal level and yields a quantile several times smaller, confirming that the running maximum is conservative. Relative to the running maximum, both alternatives increase recurrent-step savings by fewer than two percentage points and lower return, because for this utility the binding constraint is the magnitude of the predicted remaining gain rather than the conformal margin added to it. We therefore keep the running maximum.

This experiment uses one seed in one environment and demonstrates improved held-out coverage after empirical recalibration; it does not verify the discounted-return bound.

Two details of the setup affect the interpretation of these results. The reference utility is a deterministic 200-step counterfactual rollout of each first action followed by the frozen maximum-depth policy, so the utility scale is survival steps rather than the distance-based utility scale of the exact-planning tasks; the tolerance $\varepsilon = 0.49$ therefore has a different interpretation here and is expressed in survival steps. In addition, each state is collected using a fresh reset and a stopping time sampled uniformly from a fixed finite range, with trajectories restarted if they terminate before that time. For each frozen policy, this procedure samples its own decision-state distribution; it does not sample the normalized discounted occupancy in Equation (5) or establish joint exchangeability after adaptive policy selection.

E MinAtar Details

The recurrent Q network encodes the $10 \times 10 \times C$ observation with a single convolution, then applies a gated weight-tied block $K = 5$ times with an action head after every application. Every prefix receives a depth-weighted Bellman loss with weights rising linearly from 0.15 to 1, bootstrapping from the deepest head of a periodically synchronized target network. Training uses 32 parallel environments, 5M total frames, one gradient step per four environment steps, a batch size of 64, a replay buffer holding 500k transitions, double-Q bootstrapping from a target network synchronized every 1000 steps, AdamW at 3×10^{-4}, discount 0.99 and ϵ-greedy exploration annealed from 1 to 0.05 over the first half of training. Training on the five games with five seeds per game required 18.2 GPU-hours in total.

Each certificate state is collected using a fresh environment reset and a stopping time sampled uniformly from a fixed finite range; trajectories that terminate before their sampled horizon are restarted. This defines a policy-dependent decision-state distribution rather than the normalized discounted occupancy. Fresh held-out evaluation measures empirical coverage for the selected policy; it does not by itself establish the joint exchangeability required by Theorem 1. The reference utility is $q(x, a) = Q_K(x, a)$ from the frozen deepest head, and the remaining-gain targets are formed from the prefix actions scored under that head. The tolerance is $\varepsilon = 0.05$ on the normalized utility scale described in Section 5.5, corresponding to five percent of the median action-value spread on the certificate-training states.

Empirical coverage is close to nominal on every game. Held-out simultaneous coverage averages 95.26% against a nominal level of 95%, with per-game means from 94.91% to 95.58% and a mean realized violation rate of 1.26%. Mean depth is 2.52 out of 5, a 49.7% reduction, with savings ranging from 40.7% on Asterix to 60.0% on Freeway. The reference policies' returns span 4.19 to 50.23, an order-of-magnitude difference in reference quality to which the guarantee is designed to be insensitive.

We establish return preservation through an equivalence test rather than through failure to reject a difference. Across the twenty-five runs the mean log-ratio of certified to full-compute return corresponds to 99.2% retention with a 90% confidence interval of $[95.5\%, 103.0\%]$. Two one-sided tests establish equivalence within $\pm 5\%$ ($p = 0.041$) and within $\pm 10\%$ ($p = 0.0003$). No per-game difference in return retention relative to full compute is significant under a paired t test over five seeds, with the smallest p value being 0.13 on Seaquest. We report the log-ratio rather than a mean of per-seed ratios because the naive average is dominated by the weakest reference policies. Seaquest is the one game whose point estimate of return retention lies meaningfully below unity, at 91.4%, with retention below one in four of five seeds; the direction is consistent across seeds even though the magnitude is not resolved at this sample size.

The comparison with the confidence gate primarily concerns reduced depth rather than improved return. Calibrated by Learn-Then-Test at the same 5% risk target on the same calibration states, the confidence gate requires a mean depth of 4.49, compared with 2.52 for RARE^2, a factor of 1.82, with per-game ratios from $1.54\times$ on Asterix to $2.42\times$ on Freeway. The depth difference is significant on every game (paired t over five seeds, largest $p = 0.011$ on Seaquest) and across the five game means ($p = 0.001$). The return difference between RARE^2 and the gate is not significant on any game (smallest $p = 0.15$) nor across game means ($p = 0.48$), and equivalence of their returns within a $\pm 10\%$ margin is established using two one-sided tests ($p = 0.022$). RARE^2 therefore reaches the same risk target at substantially lower depth for indistinguishable return. This reproduces on a control benchmark, with a learned critic, the factor of two that Section 5.2 establishes on the planning tasks against an exactly known utility.

The occupancy-matching requirement also affects coverage in this setting. With round-zero calibration on the full-compute policy's states, coverage averages 94.18% and falls below nominal on nineteen of twenty-five runs; twenty-one of twenty-five runs require at least one empirical update motivated by Proposition 6, after which average held-out coverage reaches 95.26%. The largest quantile increase occurs for a Seaquest seed, from 6.56 to 17.19, showing that occupancy recalibration addresses routine cases rather than only exceptional ones. Savings are in turn predictable before any calibration: the zero-gain fraction at depth one, measurable from the certificate training set alone, correlates with realized savings at $r = 0.945$ across the five games, so the potential compute savings in a new domain can be estimated without running the method.

One implementation detail generalizes beyond this benchmark. Reference action values differ by an order of magnitude across these games, and in states with similar action values, an absolute tolerance can fall below the certificate's scale floor, preventing any state from being certified for implementation rather than statistical reasons. We therefore normalize the utility axis per game by the median action-value spread, making ε dimensionless.

One reproducibility constraint is specific to this experiment. The MinAtar build we use obtains its seeds from the global NumPy random state, which our environment constructor reseeds; the random stream therefore depends on the order in which environments are created throughout the pipeline. Exact bitwise reproduction therefore requires a fixed thread count and an uninterrupted run; an interrupted and resumed training run produced a materially different policy despite using the same nominal seed. Seed-to-seed variance in DQN training is large on these games regardless, with full-compute Asterix returns ranging from 2.3 to 9.2 across five seeds, which is why every number in Section 5.5 is a five-seed mean and why the return comparison is reported as a log-ratio rather than an average of per-seed ratios.

F SHARPER DAMPING FOR OCCUPANCY CALIBRATION

The population iteration in Proposition 6 uses a running maximum and is provably convergent; its empirical counterpart also never decreases the quantile. Table 11 examines whether this conservatism limits computational savings on CartPole by comparing the running-maximum iteration with the undamped best response and two schemes that allow the quantile to decrease: partial damping $q \leftarrow q + \rho(\widehat{\mathcal{Q}}(\pi(q)) - q)$, where $\widehat{\mathcal{Q}}$ is the sample quantile and any step failing the empirical coverage check is reverted, and a bisection over q whose feasibility at each probe is decided by held-out coverage on states drawn from $\pi(q)$ itself. Both schemes accept a candidate only if measured coverage on fresh states sampled under that candidate is at least $1 - \alpha$. This empirical acceptance rule does not provide the population guarantee of Proposition 6 or a finite-sample correction for selecting among candidates.

Bisection identifies a smaller quantile at which held-out coverage reaches the nominal level, yielding a quantile several times smaller than that obtained by the running maximum. The running maximum is therefore conservative, but reducing this conservatism offers little computational benefit. Relative to the running maximum, the two alternatives increase recurrent-step savings by fewer than two percentage points and reduce episodic return, because at this tolerance the magnitude of the predicted remaining gain, rather than the added conformal margin, prevents earlier stopping: even with q near zero, most states still fail the stopping condition defined by ε. A 200-step survival utility therefore leaves little scope for certified reductions in computation.

Table 11: Sharper damping schemes for occupancy calibration on CartPole, 2 independently trained seeds. All four schemes deploy $\pi(q)$, collect fresh states using the experiment's finite stopping-time sampling rule and measure coverage on those states; they differ only in how q is updated. Partial damping and bisection accept a candidate only if empirical coverage on fresh states sampled under its policy is at least $1 - \alpha$; this check is not a finite-sample guarantee for the selected candidate. Both alternatives increase recurrent-step savings by fewer than two percentage points relative to the running maximum and lower return.

Scheme	$\hat{q}_\alpha$	Coverage	Depth	Saved (%)	Return
Undamped best response	5.26	0.982	4.30	28.3	265.5
Empirical running maximum	5.58	0.977	4.23	29.5	266.6
Partial damping, coverage-guarded	1.32	0.966	4.15	30.9	250.9
Bisection with coverage check	1.30	0.955	4.20	30.0	236.2
Maximum depth	—	—	6.00	0.0	272.5

The policies in this table are trained from scratch, without reusing the pilot checkpoint in Appendix D, and their full-compute returns vary widely across seeds. The computational savings at near-nominal or conservative empirical coverage for the completed seeds are substantially larger than the 4.6% achieved by the single pilot seed; that limited saving partly reflected a weak reference policy. The 4.6% figure therefore describes the computational savings available for that particular reference policy rather than a general property of CartPole or of the method.

G Depth Stress Test

Table 12: Depth stress test. The policy and certificate are retrained from scratch at each maximum depth K, with the numbers of optimizer updates proportional to K. Training effort is therefore not held fixed. Empirical coverage remains near nominal throughout; under this training schedule, savings grow with K and the zero-gain fraction at depth one remains approximately stable.

Task	K	Seeds	Coverage	Depth	Saved (%)	Optimal (%)	Zero-gain@1
cycle20	5	3	0.950	1.59	68.2	99.68	0.887
cycle20	8	3	0.955	1.66	79.2	99.88	0.885
cycle20	12	3	0.955	1.63	86.4	99.96	0.887
cycle20	16	3	0.953	1.63	89.8	99.95	0.886
grid7	5	1	0.952	2.71	45.8	99.04	0.881
grid7	8	1	0.947	3.36	58.0	100.00	0.876
grid7	12	1	0.955	2.78	76.8	99.76	0.872

We retrain the policy and certificate from scratch at $K \in \{5, 8, 12, 16\}$, with the numbers of optimizer updates proportional to K; training effort is therefore not held fixed. Empirical coverage remains near nominal across depths, and under this training schedule savings increase from 68.2% at $K = 5$ to 89.8% at $K = 16$ on Cycle-20, while the zero-gain fraction at depth one remains near 0.886 throughout. These observations indicate stable opportunities for early stopping across the tested depths, but do not isolate the effect of depth from that of training effort. Non-monotonicity on Cycle-20 remains below 0.1% at every K, limiting the observed overthinking effect in Section 5.3 to the domains where we measure it.

H Jointly Trained Halting Baselines

Both gates use the same backbone, data and step budget as the policy in Table 4 and are trained end to end, so the halting decision influences the trajectory on which it is subsequently based. ACT (Graves, 2016) uses a per-depth halting unit with a ponder penalty $\tau(N + R)$; PonderNet (Banino et al., 2021) uses a geometric halting distribution with the loss taken in expectation over halting depth and a KL term to a geometric prior. We sweep τ because a single ponder cost specifies only one arbitrary operating point; the sweep allows a fair comparison with a method whose tolerance is adjustable.

Table 13: Jointly trained halting baselines. Unlike the binary head of Table 7, which is trained post hoc on frozen features, these gates are trained end to end so the halting decision shapes the trajectory it later judges. Each row reports results averaged over the listed training seeds for a fixed task, halting method and ponder cost τ. Tracing the frontier requires separate training at each cost, whereas RARE^2 obtains the whole frontier from a single calibration (Table 8). Neither baseline controls risk: ACT halts immediately at every τ we tried and violates the 5% target by more than a factor of two, while PonderNet is conservative but provides neither a guarantee nor an adjustable tolerance.

Task	Gate	τ	Seeds	Depth	Saved (%)	Violation (%)	Optimal (%)
cycle20	ACT	0.003	2	1.00	87.5	11.30	88.68
cycle20	ACT	0.01	2	1.00	87.5	11.28	88.70
cycle20	ACT	0.03	2	1.00	87.5	11.32	88.64
cycle20	ACT	0.1	2	1.00	87.5	11.26	88.72
cycle20	PonderNet	0.003	2	8.00	0.0	0.00	99.98
cycle20	PonderNet	0.01	2	4.93	38.3	0.00	99.94
cycle20	PonderNet	0.03	2	3.24	59.5	0.02	99.98
cycle20	PonderNet	0.1	2	2.52	68.5	0.04	99.92

The key distinction concerns the guarantee associated with stopping: a halting probability trained against a task loss does not quantify the cost of stopping, so it provides neither a quantity against which to calibrate a tolerance nor a finite-sample guarantee. ACT illustrates the failure mode directly by halting at the first prefix for every τ while violating the target risk by more than a factor of two. PonderNet performs well but requires separate training at each operating point, whereas one calibration supports the entire range in Table 8.