BUDGET-AWARE STATE CRITICALITY VIA OPTIMAL INTERVENTIONS UNDER DECAYING AGENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

In reinforcement learning, *critical states* are situations where the chosen action has a disproportionately large impact on the final outcome. Identifying such critical states is a fundamental challenge, with applications in explainability, credit assignment, and human oversight of autonomous agents. We present CODA (**C**riticality via **O**ptimal Interventions under **D**ecaying **A**gency), a framework in which an agent can intervene by overriding the action of a fallback policy, but each intervention can make future ones less likely to take effect. Critical states are those where the agent still chooses to intervene, making criticality explicitly relative to the fallback policy. We show that the resulting problem decomposes into a cascade of coupled optimal stopping problems, and build on this structure to develop algorithms for both tabular and continuous environments. We prove that critical states exist if and only if the fallback policy is suboptimal, and that, unlike in penalty-based approaches, they are invariant to the scale of the reward. Empirically, CODA identifies intuitive critical states in grid worlds and continuous control, and, when used to allocate a limited intervention budget, matches or exceeds the returns achieved with State Importance and Lazy-MDPs.

1 INTRODUCTION AND MOTIVATION

Reinforcement learning (RL) fundamentally aims to learn *policies*, i.e., a probability distribution over actions for each state the agent can be in. However, depending on the purpose, only a select few states may be of true interest. In human-AI collaboration, humans often delegate trivial choices to the machine while retaining control over critical decisions (Balazadeh et al., 2022). Similarly, when requiring an agent to justify its actions (Huber et al., 2021) or when comparing the qualitative behavior of two distinct policies, exhaustive analysis across all states is infeasible. Meaningful evaluation requires restricting focus to a carefully chosen subset of *critical states*.

Defining these critical states is an important challenge. While there is no universally agreed-upon definition of criticality, the most prevalent understanding considers a state critical if the chosen action significantly impacts the total return. The importance of this effect can be quantitatively described by the mutual information between action a in state s and the final total discounted reward. However, this measure is intractable in practice, and computationally efficient alternatives are preferred, such as the variance in the Q-values of an agent's policy (Karino et al., 2020).

In this work, we approach the problem through the lens of budgeted intervention. Consider a thought experiment where an agent performs actions in an environment, but each deliberate action can remove some of their *agency* – the extent to which the result aligns with the agent's intention. As agency dwindles, the likelihood of a completely different, unrelated action being played, increases. To the agent, it appears as though the environment is overriding their decisions. In other terms: they are losing control. Accepting this *agency decay* as an inevitable rule of the game, the agent must choose its actions with deliberation and only take action when truly necessary. The resulting performance of this agent will often be quite bad. However, it is not the performance we care about, but the resulting policy: The states in which the agent chose to act and accept a decrease in agency must have been particularly important. We use this to define a notion of *critical states*.

Our contributions are the following: (1) We introduce CODA (Criticality via Optimal Interventions under Decaying Agency), a framework that defines critical states as those where intervening on a suboptimal policy outweighs the cost of future agency loss; (2) we theoretically prove that CODA

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.

guarantees the identification of at least one critical state and, unlike penalty-based frameworks, is mathematically invariant to the scaling of the reward function; (3) we cast the identification process as a sequence of optimal stopping problems and present algorithms for both tabular and continuous environments; (4) we establish a formal theoretical link to Lazy-MDPs (Jacq et al., 2022) and prove that CODA has better budgeting capacity in states with highly exogenous transition dynamics; (5) we experimentally evaluate CODA, assess the usefulness of its critical states in downstream intervention tasks, and compare it to SI and Lazy-MDP.

2 Related Work

Torrey & Taylor (2013) explore a related teacher-student framework where an expert provides a heavily budgeted number of actions to a student. To select intervention states, they utilize different heuristics, including a metric related to state importance, $\max_a Q(s,a) - \min_a Q(s,a)$. While their setup shares the concept of action budgeting, their focus remains on evaluating heuristic strategies rather than computing an optimal one.

A common measure of state criticality relies on the variance of a Q-function, termed State Importance (SI) (Karino et al., 2020). Defined as $\mathsf{SI}_\pi(s) \coloneqq \mathrm{Var}_a[Q^\pi(s,a)]$, this metric was originally developed to balance exploration and exploitation by highlighting high-variance states with respect to the learner's current policy. In this paper, we only consider SI with respect to an optimal policy, which we denote by SI. For a fixed $q \in (0,1)$, one can extract a set of critical states as the $\lfloor q \cdot |S| \rfloor$ states with highest state importance. The behavior of state importance throughout the course of training has been studied by Ramakrishna et al. (2025) and is dependent on the exploration strategy of the agent being trained. A similar concept based around return-variance has also been explored by Spielberg & Azaria (2019). We use SI as a primary comparative baseline.

Other approaches identify critical states without assuming known Q-values. EDGE (Guo et al., 2021) and RUDDER (Arjona-Medina et al., 2019) make use of recurrent sequence models for return prediction and credit assignment to extract important state-action pairs from episodes. Similarly, Liu et al. (2023) proposed Deep State Identifier (DSI), which learns a probabilistic mask over state sequences to find states most critical for the predicted return.

Most similar to our approach is the framework of *Lazy-MDPs* by Jacq et al. (2022). They augment the original MDP with a "lazy action" that defers action selection to a fallback policy. The key difference lies in how the use of action deferral is encouraged: while Lazy-MDPs penalize non-lazy actions with an explicit reward penalty η, our framework completely avoids direct reward manipulation by instead altering the transition dynamics.

3 Preliminaries

In this paper, we consider a given Markov Decision Process (MDP) $\mathcal{M} = (S, A, R, P, \gamma)$ with finite state and action spaces S and A, reward function $R\colon S \times A \to \mathbb{R}$, transition kernel $P\colon S \times A \to \Delta_S$, where Δ_X denotes the set of probability distributions over a finite set X, and discount factor $\gamma \in [0,1)$. For a policy $\pi\colon S \to \Delta_A$, we denote state and action-value functions by V^π and Q^π as usual. For an arbitrary value function $V\colon S \to \mathbb{R}$, we adopt the shorthand $Q = \mathcal{L}V$ for the *one-step lookahead* operation, $Q(s,a) = (\mathcal{L}V)(s,a) = R(s,a) + \gamma \mathbb{E}_{s' \sim P}[V(s')]$.

Our primary comparative baselines will be the state importance (SI) metric used by Karino et al. (2020) and the Lazy-MDP framework introduced by Jacq et al. (2022). As mentioned in the previous section, we exclusively consider the state importance with respect to an optimal policy and define $\mathsf{SI}(s) = \mathrm{Var}[Q^\star(s,\cdot)]$. Given a quantile parameter q, the critical state set for SI is defined as $C_{\mathrm{SI}}(q) = \{s \in S : \mathsf{SI}(s) > Q(q)\}$, where $Q(q)$ denotes the q-quantile of SI.

Meanwhile, Lazy-MDP features a default policy ρ and a penalty $\eta > 0$. The MDP is augmented by a default action $\bar{a}$ and the reward function is modified to $R_{\mathrm{lazy}}(s,a) = R(s,a) - \eta \mathbb{I}\{a \neq \bar{a}\}$. We denote its critical states by $C_{\mathrm{lazy}}(\rho,\eta) = \{s \in S : \bar{a} \notin \arg\max_{a \in A} Q^\star_{\mathrm{lazy}}(s,\cdot)\}$, dropping the parameters whenever they are clear from context.

4 METHOD

CODA aims to obtain a subset of states $C \subset S$ that are *critical* relative to $\mathcal{M}$ and some fixed baseline policy. It is derived by assuming a system that executes a baseline (or fallback) policy $\rho : S \to \Delta_A$ within the environment whenever the agent plays a dedicated NOOP action. The agent can also choose to intervene and override the baseline policy by playing $a \in A$. Not every intervention is successful: The environment accepts an intervention with some probability $\alpha(h_t)$, where h_t denotes the state-action history at time t. When rejected, the intervention is ignored and the action is selected according to ρ instead, as if the agent had not intervened at all. Formally, the effective policy executed in the environment $\mathcal{M}$ is thus

$$\pi_{\text{eff}}(a_t \mid h_t) = \begin{cases} \alpha(h_t)\mathbb{I}\{a_t = \tilde{a}_t\} + (1 - \alpha(h_t))\rho(a_t \mid s_t), & \tilde{a}_t \in A \\ \rho(a_t \mid s_t), & \tilde{a}_t = \text{NOOP} \end{cases} \tag{1}$$

In the following, we further restrict the form of α by assuming that $\alpha(h_t) = \lambda(k(h_t))$, where $k(h_t)$ is the number of interventions – counted as the number of non-NOOP actions – by the agent contained in the history h_t, and $\lambda \colon \mathbb{N} \to [0, 1]$ is a monotonically decreasing function that satisfies $\lambda_0 > 0$.[1] Thus, acceptance is independent of both the state and proposed intervention, and only depends on the number of interventions so far.

The behavior of λ controls how quickly the success rate of interventions dwindles. While CODA works for any decay function with the aforementioned properties, we particularly focus on three families: (i) exponential decay $\exp(-\beta k)$ with $\beta > 0$, (ii) linear decay $\max\{0, 1 - k/K\}$ with $K \in \mathbb{N}$, (iii) step function $\mathbb{I}\{k < K\}$ with $K \in \mathbb{N}$. The parameter β or K controls the rate of decay in the first two settings, and the support of λ in all three settings.

By augmenting the state space to record the intervention count, $\widetilde{S} = S \times \mathbb{N}$, the effective policy π_{eff} becomes Markovian. This allows us to embed it into the transition dynamics of $\mathcal{M}$, yielding a surrogate MDP, $\mathcal{M}_{\text{crit}}(\rho, \lambda) \coloneqq (\widetilde{S}, \widetilde{A}, \widetilde{R}, \widetilde{P}, \gamma)$. Formal descriptions of $\widetilde{P}$ and $\widetilde{R}$ are provided in Appendix A, and we write $\widetilde{A} = A \cup \{\text{NOOP}\}$. By construction, any deterministic optimal policy $\pi^\star_{\text{crit}}$ of $\mathcal{M}_{\text{crit}}(\rho, \lambda)$ represents an optimal intervention strategy under the described acceptance dynamics. For any $k \in \mathbb{N}$, we define the set of critical states as those where every optimal policy would intervene:

$$C_k(\rho, \lambda) \coloneqq \{s \in S \mid \pi^\star_{\text{crit}}(s, k) \neq \text{NOOP} \ \ \forall \pi^\star_{\text{crit}} \in \Pi^\star(\mathcal{M}_{\text{crit}}(\rho, \lambda))\} \tag{2}$$

Whenever it is clear from context, we drop the parameters and simply write C_k. While these sets are defined for all k, we are often only interested in C_0. The critical state sets for $k > 0$ correspond to scenarios in which k interventions have already been made. Crucially, CODA's critical states are mathematically guaranteed to exist as long as ρ is suboptimal:

Theorem 1. *Let $\lambda \colon \mathbb{N} \to [0, 1]$ be a monotonically decreasing function with $\lambda_0 > 0$. Then, $C_k(\rho, \lambda)$ is non-empty for all $k \in \mathbb{N}$ with $\lambda_k > 0$ if, and only if, ρ is not optimal.*

This guarantee allows us to freely choose λ without having to worry about not recovering any critical states. In contrast, for Lazy-MDPs, such a guarantee only exists for penalty values smaller than some threshold $\eta_{\max}$, which has to be determined with additional computational effort. Later in Section 4.3, we will also see that our parameterization is invariant to the reward scale.

4.1 IMPLEMENTATION AND TRAINING

In practice, the above formulation is computationally undesirable, because it creates a copy of the entire state space for each natural number. The linear decay and the step function are naturally only supported on $\{0, \dots, K - 1\}$, and for the exponential decay, there is a number of interactions after which λ_k is so small that the influence of interventions becomes negligible. Defining this threshold is an arbitrary choice, and we set it to $\tau = 0.01$ in our experiments, introducing a maximum interaction horizon $K(\beta) = \lceil -\ln(\tau)/\beta \rceil$. In general, the modified state space becomes $S \times \{0, \dots, K\}$. Once a state (s, K) has been reached, we truncate the decay, effectively forcing $\lambda_k = 0$ for $k \geq K$. Because the truncation preserves monotonicity, Theorem 1 still applies in the truncated scenario.

[1] Throughout, we use the notation $\lambda_k = \lambda(k)$.

4.1.1 THE SINGLE-INTERVENTION CASE

To understand the problem structure, we begin by analyzing the problem for a simple fallback function. In this case, $\lambda_0 = 1$ and $\lambda_k = 0$ for all $k \geq 1$: Any intervention immediately drops the acceptance probability to zero. The three decay families introduced earlier converge to this λ when picking $K = 1$, or when $\beta \to \infty$ in case of an exponential decay. From the agent's perspective, the episode effectively ends as soon as the first intervention is made, as the environment will simply execute ρ for the remainder of the episode. We can model this by giving the agent a reward of $Q^\rho(s, a)$ and terminating the episode as soon as they intervene for the first time. Under these circumstances, an optimal intervention policy will only suggest $a \in A$, if it has the largest Q^ρ-value in s. This reduces the action space to just two actions: NOOP and INTERVENE. By assumption, the $Q^\star$-values of INTERVENE are known in every state,

$$Q^\star(s, \text{INTERVENE}) = \max_{a \in A} Q^\rho(s, a),$$

leaving only the $Q^\star$-values for NOOP to be learned. Note that this is essentially an *optimal stopping* problem: stopping corresponds to intervening, which ends the episode. Meanwhile, playing NOOP corresponds to continuing the interaction. Once $Q^\star$ is known, it follows from equation 2 that a state s should be deemed *critical* if, and only if, $\max_{a \in A} Q^\rho(s, a) > Q^\star(s, \text{NOOP})$.

4.1.2 MULTIPLE INTERVENTIONS AND CASCADED VALUE ITERATION

If more than one intervention is accepted, CODA unfolds as a cascade of interdependent optimal stopping problems. Denote by $V^{(k)} \coloneqq V^\star_{\text{crit}}(\cdot, k)$ the optimal value function of what we call *stage* k. The key insight is that, because k only ever increases throughout an episode, $V^{(k)}$ can be computed independently of $V^{(\ell)}$ for any $\ell < k$.

Proposition 1. *We have $V^{(K)} = V^\rho$. For $k \in \{0, \ldots, K-1\}$, the function $V^{(k)}$ is the solution of an optimal stopping problem where the value of stopping is*

$$V^{(k)}_{\text{stop}}(s) \coloneqq \lambda_k \max_{a \in A} Q^{(k+1)}(s, a) + (1 - \lambda_k)\mathbb{E}_{a \sim \rho(\cdot|s)}[Q^{(k+1)}(s, a)], \tag{3}$$

and the continuation value is

$$V^{(k)}_{\text{cont}}(s) \coloneqq \mathbb{E}_{a \sim \rho(\cdot|s)}[Q^{(k)}(s, a)]. \tag{4}$$

Consequently, we have $V^{(k)} = \max\{V^{(k)}_{\text{cont}}, V^{(k)}_{\text{stop}}\}$ and the critical state sets can be expressed as $C_k = \{s \in S \mid V^{(k)}_{\text{stop}}(s) > V^{(k)}_{\text{cont}}(s)\}$.

Following Proposition 1, CODA's objective can be solved using a cascade of optimal stopping problems, starting with stage $K - 1$ and ending at stage 0. For a tabular MDP, each stage's optimal stopping problem is solvable with a value iteration. We call the entire procedure the *cascaded value iteration* (CVI). It is formally written down in Algorithm 2 of the appendix.

Proposition 2 (Error propagation)**.** *The approximation $\widetilde{V}^{(k)}$ of the stage-k value function $V^{(k)}$ computed by the cascaded value iteration satisfies $\|\widetilde{V}^{(k)} - V^{(k)}\|_\infty \leq \frac{\varepsilon\gamma(1-\gamma^{K-k+1})}{(1-\gamma)^2}$ for every $k \in \{0, \ldots, K-1\}$, provided that $\gamma < 1$.*

4.1.3 BEYOND TABULAR ENVIRONMENTS WITH DEEP CODA

The CVI solves tabular environments nearly exactly, but relies on the environment being fully known. While $\mathcal{M}_{\text{crit}}(\rho, \lambda)$ is constructed in such a way that it can be simulated whenever $\mathcal{M}$ can be simulated, a naive application of online RL algorithms suffers from severe sample inefficiency: as k advances, the agent's actions rarely impact the environment, which eliminates almost all training signal. However, because λ_k is only an analytical mixture weight, there is no need to sample the surrogate MPD at all. Consequently, we use TD3-like off-policy reinforcement learning (Fujimoto et al., 2018). To design the architecture, we first observe that the key quantity of interest is the difference between stopping and continuation value, which can be re-written as

$$V^{(k)}_{\text{stop}}(s) - V^{(k)}_{\text{cont}}(s) = \lambda_k \underbrace{(\max_a Q^{(k+1)}(s, a) - \mathbb{E}_{a \sim \rho(\cdot|s)}[Q^{(k+1)}(s))]}_{=: A^{(k)}(s)} - \underbrace{(V^{(k)}_{\text{cont}}(s) - V^{(k+1)}_{\text{cont}}(s))}_{=: \delta^{(k)}(s)},$$

so that we only need to learn continuation value increments $\delta^{(k)} \geq 0$ and stopping advantages $A^{(k)} \geq 0$ for every stage. This leads to the following decompositions of the value functions: (i) $V^{(k)}_{\text{cont}}(s) = V^{\rho}(s) + \sum_{i=k}^{K-1} \delta^{(i)}(s)$; (ii) $V^{(k)}_{\text{stop}}(s) = V^{\rho}(s) + \sum_{i=k+1}^{K-1} \delta^{(i)}(s) + \lambda_k A^{(k)}(s)$; (iii) $V^{(k)}(s) = V^{\rho}(s) + \sum_{i=k}^{K-1} \delta^{(i)}(s) + \max\{0, \lambda_k A^{(k)}(s) - \delta^{(k)}(s)\}$. The stage-wise Q-functions (or *critics*) are also expressed using increments, $Q^{(k)}(s,a) = Q^{\rho}(s,a) + \sum_{i=k}^{K-1} \Delta_i(s,a)$. This is for two reasons: (1) the Q-values are monotonically non-increasing in k (proven in Appendix B), and (2) they are used to fit the continuation value, which is itself composed of increments, allowing us to pair the increments during the learning process.

As in TD3, we use twin critics Q_{θ_1} and Q_{θ_2} and maintain a replay buffer of transitions. We also maintain an additional *anchor buffer* that contains (partial) trajectories generated by ρ. The anchor buffer is used to train V^{ρ} using bootstrapped n-step returns (Sutton & Barto, 2018). During the initialization phase, V^{ρ} is fit directly on the discounted returns of Monte Carlo simulations of ρ in the environment, where random starts are used whenever possible.[2] The collected transitions are added to both the replay and anchor buffer. During the main loop, the algorithm alternates for the desired number of iterations between the following steps: (i) Policy rollout / data collection, (ii) Fit the critics (θ_i), (iii) Fit the stage K value using n-step returns from the anchor buffer, (iv) Fit the lower stage continuation value increments by regressing to the expected critic increments under action draws by ρ, (v) Fit the stopping advantages, (vi) Fit the actor (φ) by maximizing the first critic.

We also make use of ensembles, which share the two buffers, but are otherwise initialized and trained independently. Further details on the architecture and the loss terms is given in Appendix C.

4.2 On the Choice of Fallback Policy

A key parameter of CODA is the fallback policy ρ, which is the environment's baseline behavior. We briefly discuss several possible choices for the fallback policy. In our experiments we primarily utilized a uniform fallback.

(i) *Environment-intrinsic baselines*: Fallback policies derived from the MDP alone yield critical states that reflect the environment itself. The uniform policy π_{unif} is a natural choice in the absence of prior information. Since $\rho = \pi^{\star}$ yields no critical states (Theorem 1), a noisy optimal policy $\rho = (1-\varepsilon)\pi^{\star} + \varepsilon\pi_{\mathsf{unif}}$ retains a nontrivial critical set while staying close to optimal behavior. At the other extreme, a reward-minimizing policy actively seeks out failure, extending criticality into otherwise safe regions. (ii) *Agent-specific baselines*: If the policy of a particular agent or user is known or can be inferred, setting ρ to it concentrates interventions on states where that agent is likely to err. Regions where it is already proficient remain non-critical. (iii) *Interpretable baselines*: A simple, transparent, but slightly suboptimal ρ makes the critical states mark the few places where handing control to a stronger, opaque policy pays off most.

Crucially, the dependence of the critical state set on the fallback policy ρ is a deliberate feature of CODA rather than a limitation. We argue that criticality is inherently contextual: a bottleneck that is dangerous under a weak fallback may be trivial under a strong one that is able to navigate it without intervention. This perspective also aligns with the underlying philosophy of the Lazy-MDP framework (Jacq et al., 2022).

4.3 Reward Scale Invariance

A significant theoretical advantage of CODA over penalty-based frameworks like Lazy-MDPs is its invariance to uniform reward scaling. For $c > 0$, let $\mathcal{M}_c$ be an MDP identical to $\mathcal{M}$ but with a scaled reward function $R_c(s,a) \coloneqq cR(s,a)$.

Proposition 3 (Reward scale invariance)**.** *Fix λ and ρ. Let $c, c' > 0$. Then, $C_k(\mathcal{M}_c) = C_k(\mathcal{M}_{c'})$ for every $k \in \mathbb{N}$.*

[2] Random starts are not necessary for CODA to work, but not using them may restrict the parts of the state space where the criticality estimates are trustworthy. This is particularly important when querying criticality on states which are far from ρ's visitation distribution.

This property cannot be satisfied by a penalty-based framework: if the rewards are scaled and the penalty η stays the same, the critical states will inevitably change. Consequently, a practitioner cannot adjust η without implicitly making assumptions about the scale of the underlying MDP.

4.4 Relation to Lazy-MDP Framework

The setup behind CODA and Lazy-MDPs is similar: both frameworks feature a designated meta-action that defers the selection of the "physical" action to a baseline policy. Despite this similarity, both methods differ in the way they motivate the use of this deferring action: Lazy-MDPs hand out a fixed reward penalty η when it is not used, while CODA decreases the acceptance probability of internventions. The following result shows that, in the single-intervention case, CODA is equivalent to a variant of the Lazy-MDP framework, in which the penalties depend in a specific way on the current state and action.

Proposition 4. *Let $\lambda = \mathbb{I}\{k = 0\}$. Then, the set of optimal policies under CODA is the same as the set of optimal policies under the Lazy-MDP framework with penalty $\eta(s, a) = Q^{(0)}(s, a) - Q^{\rho}(s, a)$, where $Q^{(0)}$ is the stage-0 action-value function of CODA.*

However, because this penalty depends on $Q^{(0)}$, which is a function not available prior to applying CODA, this result is only a theoretical bridge between the two frameworks and not of practical interest.

4.5 Budget-Awareness in States with Exogenous Dynamics

We conclude the section by providing a result which shows that, depending on the degree of control in a state, CODA is truly *budget-aware*, while the two baselines are not.

For a state s, let $G(s) \coloneqq \max_{a \in A} R(s, a) - \mathbb{E}_{a \sim \rho(\cdot|s)}[R(s, a)]$ denote the immediate reward gain of the best action over the fallback, and let $\kappa(s) \coloneqq \max_{a,a' \in A} \mathrm{TV}\big(P(\cdot \mid s, a), P(\cdot \mid s, a')\big)$ denote the *degree of endogeneity*: how much the choice of action can influence the next state. When $\kappa(s) = 0$, the dynamics at s are exogenous. Write $P^{\rho}(\cdot \mid s) \coloneqq \sum_a \rho(a \mid s) P(\cdot \mid s, a)$ for the next-state distribution under the fallback, and define the *option value* of stage k as $\Omega_k(s) \coloneqq \gamma\, \mathbb{E}_{s' \sim P^{\rho}(\cdot|s)}[V^{(k)}(s') - V^{(k+1)}(s')]$. Finally, let $\varepsilon(s) \coloneqq \gamma\, \kappa(s)\, (R_{\max} - R_{\min})/(1-\gamma)$.

Proposition 5 (Local decidability)**.** *Let $s \in S$ and $k \in \mathbb{N}$ with $\lambda_k > 0$. Then:*

- *(i)* $\left|\sqrt{\mathsf{SI}(s)} - \mathrm{std}_a R(s, \cdot)\right| \le \varepsilon(s)$;
- *(ii) Lazy-MDP: $G(s) > \eta + \varepsilon(s)$ implies $s \in C_{\mathrm{lazy}}$, and $G(s) < \eta - \varepsilon(s)$ implies $s \notin C_{\mathrm{lazy}}$;*
- *(iii) CODA: $\lambda_k G(s) > \Omega_k(s) + \varepsilon(s)$ implies $s \in C_k$, and $\lambda_k G(s) < \Omega_k(s) - \varepsilon(s)$ implies $s \notin C_k$.*

In particular, if s is exogenous, then $\varepsilon(s) = 0$ and (i)–(iii) hold with equality resp. as equivalences.

The main takeaway of this proposition is that, outside of a gray zone of width $2\varepsilon(s)$, each method's criticality verdict is decided solely by the reward function, either via $G(s)$ or the standard deviation of $R(s, \cdot)$. The size of the gray zone is proportional to $\kappa(s)$, the degree of endogeneity of the state. The difference between the methods lies in the threshold, which forms the center of the gray zone. For SI and Lazy-MDP, it is a global constant, while for CODA it is the option value $\Omega_k(s)$, which depends on the values of downstream states. As a consequence, CODA is capable of separating states with identical reward profiles even when $\kappa(s)$ is very low.

The following example illustrates a consequence of this proposition. There are five states arranged in a column which are visited from top to bottom, regardless of which of the two actions a_1 and a_2 is played. In other words, the transition dynamics are fully exogenous and $\varepsilon(s) = 0$ for all states. The reward function is $R(z_2, a_2) = R(z_4, a_2) = r_-$ and

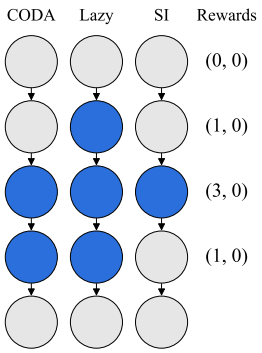


Figure 1: Example construction. Critical states are highlighted in blue.

$R(z_3, a_2) = r_+$ where $0 < r_- \ll r_+$, and 0 everywhere else. In particular, action a_1 always yields 0 reward. The fallback policy ρ always plays a_1. By Proposition 5, both SI and Lazy-MDP are unable to classify states z_2 and z_4 differently, as they share the same reward statistics – they are either both critical or non-critical. In stage $k = 0$, CODA's condition for z_2 is

$$\lambda_0 r_- > V^{(0)}(z_3) - V^{(1)}(z_3),$$

while it is $\lambda_0 r_- > V^{(0)}(z_5) - V^{(1)}(z_5) = 0$ in state z_4. Because $r_- > 0$ and $\lambda_0 > 0$, z_4 will always be critical. Meanwhile, for small enough r_-, state z_2 may be classified as uncritical. Figure 1 compares the critical states identified by CODA with $\lambda = \mathbb{I}\{k = 0\}$, Lazy-MDP with $\eta = 0.25$ and SI with $q = 0.1$ on this example.

5 Experimental Analysis

The goal of the experimental analysis is to answer the following questions: (1) How does the choice of ρ and λ affect the resulting critical states? (2) How different are the critical states found by CODA to those found by SI and Lazy-MDP? (3) Do CODA's critical states translate to meaningful improvements in return, when used to select states to intervene in?

Our evaluation covers several environments: (i) `CliffMaze`, a small gridworld introduced by Karino et al. (2020), as well a continuous variant; the goal is to reach the goal in the bottom right, but there is a cliff in the center column which can only be passed via the center row, (ii) `UnlockPickup` from the MiniGrid suite (Chevalier-Boisvert et al., 2023), another gridworld but with a significantly larger state space, (iii) `LunarLander`, a low-dimensional continuous control environment, and (iv) `Hopper` from the MuJoCo suite. For environments (i)–(ii), we use value iteration as well as its lazy and cascaded variants. For environments (iii)–(iv), we use SAC (Haarnoja et al., 2018) to train the expert policies used for computing SI.

Sparsity of critical states (RQ1) We evaluated the three decay function families *exponential*, *linear* and *step* on CliffMaze and UnlockPickup for different decay horizons $K \in \{1, 2, 3, 5, 10, 20, 30, 50, 100\}$. Across environments, parameters and fallback policies, we find that an exponential decay produces the sparsest results, followed by a linear decay, while a step function produces the highest fraction of critical states. The full table can be found in Appendix D. Based on this result, we adopt the exponential decay as the standard decay function, unless otherwise mentioned, as it produces the sparsest critical state sets among the three families we studied.

Choice of fallback policy (RQ1) Figure 2 (b) shows several examples of how altering the fallback policy affects the resulting critical states on CliffMaze. We evaluated the following fallback policies: (i) Random: $\rho = \pi_{\text{unif}}$, (ii) Noisy optimal: $\rho = 0.8\pi^\star + 0.2\pi_{\text{unif}}$, (iii) Minimizer: ρ *minimizes* total discounted reward, (iv) Careful: ρ prefers to move to states which are further away from the cliff. Along the rows, β varies from ∞ down to 0.05. We observe that the noisy optimal policy's critical states for $\beta \to \infty$ resemble those of the random policy, but restricted to the right half of the environment. The critical states of the minimizing policy reach even further into the "safe" regions of the environment, which matches our intuition of criticality: a baseline policy that seeks the cliffs out more aggressively requires earlier interventions. The opposite happens with the careful fallback: as it actively avoids cells near the cliff and will never step into the cliff, the critical states are mostly located on the left side to encourage the agent to cross the passage to the other side.

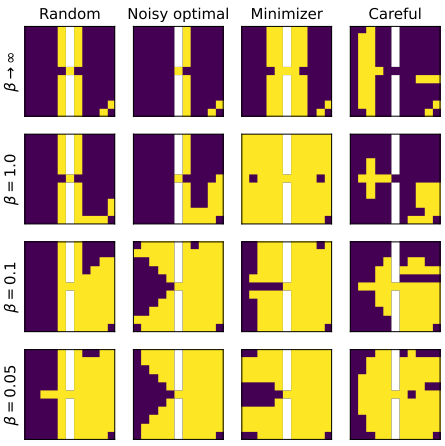


Figure 2: Critical states (yellow) identified by CODA-exp with $\beta \in \{\infty, 1.0, 0.1, 0.05\}$ and various fallback policies on CliffMaze.

Similarity to baselines (RQ2) To assess the similarity of CODA-exp to both SI and Lazy-MDP, we computed confusion matrices resulting from the critical state classifications on

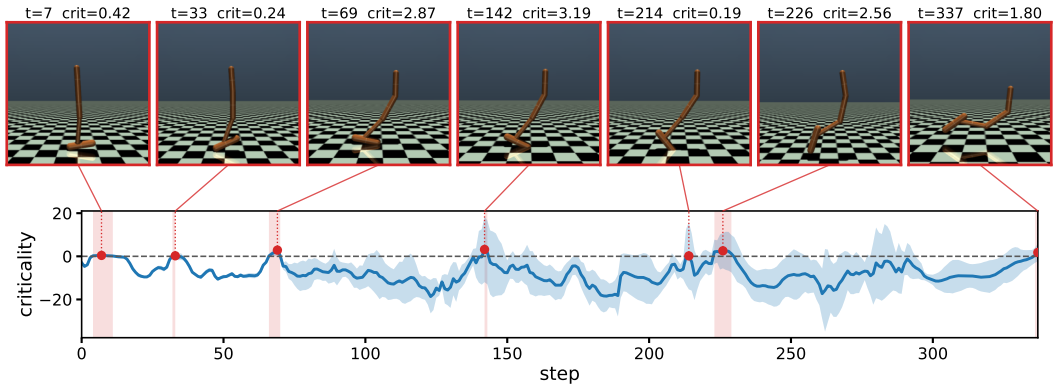


Figure 3: CODA-exp criticality with uniform ρ and $\beta = 0.5$ on an episode generated by a SAC policy. Highlighted regions are critical. Ensemble uncertainty grows as the observations move away from ρ's state distribution.

`UnlockPickup`. In order to equalize the sizes of the critical states, we devised three categories, depending on the fraction of the state set classified as critical: low (5% of all states are critical), medium (10%) and high (20%). For each category, we picked method parameters that achieve the desired percentage. We found CODA to have relatively low similarity to SI (Jaccard 0.26 at 5% and 0.35 at 20%) and a much higher similarity to Lazy-MDP (0.68 at 5% and 0.85 at 20%). This result was consistent with our expectations, given that the core mechanism of deferring to a default policy is shared between CODA and Lazy-MDPs. Nonetheless, there is still a significant difference between CODA and Lazy-MDPs, especially in the low-density regime. The full table can be found in Appendix D.

Validation of DeepCODA against CVI reference To validate the Deep CODA pipeline, we created a continuous variant of the CliffMaze environment. It shares the layout of its tabular counterpart, but the state space is $S = [0, 1]^2$, and actions correspond to two-dimensional vectors of fixed length, which we set to $1/11$ to match the original proportions. Both the cliff and the goal region have the same width as the step length. We discretized this environment into a tabular MDP with 3025 states and 16 actions in order to generate an approximate ground truth using CVI, against which the DeepCODA solution could be compared. Table 3 shows the resulting error rates and F_1 scores after 20 iterations, averaged across 3 seeds, using a uniform fallback and an exponential decay with $\beta = 1.0$. The overall error rate is low at 3.1% ($\sigma = 0.26\%$) while the F_1 score averages 0.95 ($\sigma = 0.0035$).

5.1 Evaluation on Intervention Tasks (RQ3)

In order to assess the quality of the identified critical states, we used them in a downstream intervention problem: The baseline policy ρ, uniform random in all experiments, selects an action by default, but the action can be overridden up to $B \in \mathbb{N} \cup \{\infty\}$ times per episode. Such an intervention is *always* accepted. Each method selects *where* to intervene, but the intervening action is selected from a pre-trained expert policy, enabling a fair comparison. For CODA and Lazy-MDP in particular, it means that their optimal policies are only used for critical state extraction, not for action selection. Also for CODA, only C_0 is used. As such, all three methods are evaluated on a task that differs from their training objective. An additional baseline chooses the intervention states at random.

Tabular environments As compute is cheap, we performed a dense sweep over parameters for all methods. For each method we pick the best configuration by cross-fitting over 200 episodes and report paired 95% bootstrap CIs. On UnlockPickup, when the budget is tight but not trivial ($13 \leq B \leq 17$), CODA gets the highest return at every budget, 19–39% above the best baseline (e.g. 0.49 vs. 0.41 at $B = 17$, where the gap is significant). Close to the unlimited-expert regime ($B \geq 18$), Lazy-MDP catches up and is ahead at $B = 19$. Without a budget, CODA needs the fewest interventions to reach 30–50% of expert return (14.6/15.5/16.4 vs. 15.4/16.2/17.2 for Lazy-MDP and 16.2/16.6/17.5 for SI; all significant), while random gating needs about 130. The gap to

CODA-step(B), the best policy limited to B interventions, shows there is still room to improve. On CliffMaze, CODA and Lazy-MDP are statistically tied at most budgets, with SI falling behind from $B = 9$ onward. The full results are in Appendix D.

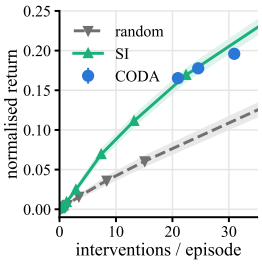


Figure 4: Results on Hopper with infinite budget.

Continuous Environments For the continuous environments, we evaluated CODA-exp at a select few parameters ($\beta \in \{0.5, 1, \infty\}$) and compared against random and SI who had their parameters swept. On Hopper, the CODA models use 21–31 interventions per episode, as seen in Figure 4. We therefore compare the methods only where their intervention counts are matched. There, CODA reaches more than twice the return of random interventions and comes within 0.02 of SI, tying it at $B = 25$. Notably, it does this without SI's access to the expert's critic. In LunarLander at $B = 100$ (Figure 5, the best mixture of swept configurations using as many interventions per episode as CODA does is significantly worse at all three points (e.g. -7.6 vs. -17.3 for random and -62.0 for SI at 39 interventions). At $B = 20$, CODA also attains the highest return outright (-26.4 vs. -45.4). With a loose budget, the baselines reach a similar peak return (4.0 for SI vs. 2.7), but only with about 50% more interventions (76 vs. 51).

6 Discussion and Future Work

Our results provide strong evidence that CODA discovers high-leverage states which can meaningfully improve return when intervening on them. It performed significantly better than random selection on all environments except LunarLander, where it performed marginally better than random selection but still significantly better than SI under lower budgets.

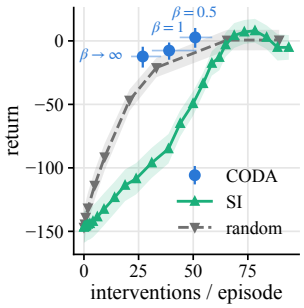


Figure 5: Results on LunarLander with $B = 100$.

While the evidence is strong, our evaluation in the continuous case is sparse, as Lazy-MDP is missing, and CODA is only evaluated for a few parameter points. The latter is partially due to the fact that CODA is compute-intensive: its cost scales with K for both CVI and Deep CODA, although the latter makes it easier to cut corners by sampling k during training, rather than training on all stages. This is something to be investigated in future work. Another avenue for future work is to try combining several parameters into a single model, so that not only the stage k is an input, but also the parameter, such as β for CODA-exp.

7 Conclusion

We have presented CODA, a framework which defines critical states of an environment as those states in which the advantage of intervening on a suboptimal policy outweighs the cost of future loss of agency. By encoding the cost of intervening in the transition dynamics rather than in the reward, CODA makes criticality explicitly relative to a fallback policy while avoiding a reward penalty that must be tuned to the scale of the MDP. We showed that CODA finds at least one critical state whenever the fallback policy is suboptimal, and that its critical states are invariant to scaling of the reward function. Exploiting its decomposition into a cascade of optimal stopping problems, we presented CVI and Deep CODA for tabular and continuous environments, and proved an error bound for the former. We also showed that, in states with weakly endogenous dynamics, SI and Lazy-MDP compare local reward statistics against a global threshold, whereas CODA uses a state-dependent option value, allowing it to separate states with identical local reward statistics. Empirically, on a budgeted intervention task in tabular and continuous environments, CODA was competitive with or outperformed the baselines, particularly under tight budgets. More broadly, we view CODA as a step toward principled tools for deciding *when* a human or a stronger policy should take control of an autonomous agent, a question that becomes increasingly relevant as such agents are deployed with limited oversight.

AI USE STATEMENT

In this work, we used generative AI tools for the formulation of mathematical claims, assistance in the writing of proofs, feedback on research methodology, and implementing methods. We have not used generative AI tools for providing critical ingredients for proving mathematical claims, generating synthetic data sets, developing theoretical models or conceptual frameworks, proposing or refining hypotheses, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. Additionally, we used generative AI tools for creating or modifying scientific figures or images, creating or editing software code, creating artifacts, drafting parts of a research paper, summarizing or analyzing existing literature, discovering research topics or identifying gaps, brainstorming, sourcing/searching for information, editing the paper to improve readability, identifying relevant literature, and proposing a title and keywords for the paper. We have reviewed all AI-assisted work: we verified LLM-generated code and checked ideas for potential plagiarism through a manual literature survey. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have taken several steps to ensure the reproducibility of our results. The source code for CODA and our experiments is provided in the supplementary material. For the theoretical results, the extended MDP $\mathcal{M}_{\mathrm{crit}}(\rho, \lambda)$ is formally defined in Appendix A.1, and complete proofs of all statements are given in Appendix B. Pseudocode for cascaded value iteration and its optimal stopping subroutine is given in Algorithms 1 and 2. The architecture, loss functions, and training procedure of Deep CODA are described in Appendix C. For the experiments, the environments and fallback policies are described in Section 5, with the precise definition of the careful fallback policy in Appendix A.2. The parameter choices for the similarity analysis, the parameter sweep procedure for the budgeted intervention task, and the training details for the continuous environments, including the expert policy configurations taken from rl-baselines3-zoo, are reported in Appendix D.

REFERENCES

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems 32*, 2019.

Vahid Balazadeh, Abir De, Adish Singla, and Manuel Gomez Rodriguez. Learning to switch among agents in a team via 2-layer markov decision processes. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831, 2023.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018.

Wenbo Guo, Xian Wu, Usmann Khan, and Xinyu Xing. Edge: Explaining deep reinforcement learning policies. In *Advances in Neural Information Processing Systems 35*, 2021.

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

Tobias Huber, Katharina Weitz, Elisabeth André, and Ofra Amir. Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artificial Intelligence*, 301: 103571, 2021.

Alexis Jacq, Johan Ferret, Olivier Pietquin, and Matthieu Geist. Lazy-mdps: Towards interpretable rl by learning when to act. In *Proceedings of the European Workshop on Reinforcement Learning 15*, 2022.

Izumi Karino, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Identifying critical states by the action-based variance of expected return. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, 2020. doi: 10.1007/978-3-030-61609-0_29.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times, second edition*. American Mathematical Society, 2017. URL https://pages.uoregon.edu/dlevin/MARKOV/markovmixing.pdf.

Haozhe Liu, Mingchen Zhuge, Bing Li, Yuhui Wang, Francesco Faccio, Bernard Ghanem, and Jürgen Schmidhuber. Learning to identify critical states for reinforcement learning from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1955–1965, 2023. doi: 10.1109/ICCV51070.2023.00187.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

Antonin Raffin. Rl baselines3 zoo. https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

Chidvilas Karpenahalli Ramakrishna, Adithya Mohan, Zahra Zeinaly, and Lenz Belzner. The evolution of criticality in deep reinforcement learning. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART)*, 2025.

Yitzhak Spielberg and Amos Azaria. The concept of criticality in reinforcement learning. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 251–258, 2019. doi: 10.1109/ICTAI.2019.00043.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Lisa Torrey and Matthew E. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, 2013.

A Definitions

A.1 Extended MDP Definition

For a finite set X, denote by Δ_X the set of probability distributions over X. Let $\rho\colon S \to \Delta_A$ and $\lambda\colon \mathbb{N} \to [0,1]$ be given. The extended MDP $\mathcal{M}_{\text{crit}}(\rho,\lambda)$ consists of the state set $\widetilde{S} = S \times \mathbb{N}$, action set $\widetilde{A} = A \cup \{\text{NOOP}\}$. Furthermore, the transition function is defined as

$$\widetilde{P}((s',k') \mid \tilde{a},(s,k)) = \begin{cases} \sum_{a\in A} \rho(a \mid s) P(s' \mid a, s) & \text{if } \tilde{a} = \text{NOOP and } k' = k, \\ \lambda_k P(s' \mid \tilde{a}, s) + (1-\lambda_k) \sum_{a\in A} \rho(a \mid s) P(s' \mid a, s) & \text{if } \tilde{a} \neq \text{NOOP and } k' = k+1, \\ 0 & \text{otherwise.} \end{cases}$$

In a similar vein, the reward function is defined as

$$\widetilde{R}((s,k),\tilde{a}) = \begin{cases} \sum_{a\in A} \rho(a \mid s) R(s,a) & \text{if } \tilde{a} = \text{NOOP}, \\ \lambda_k R(s,\tilde{a}) + (1-\lambda_k) \sum_{a\in A} \rho(a \mid s) R(s,a) & \text{otherwise.} \end{cases} \tag{5}$$

A.2 Definition of Custom Value Function

In Section 5, we analyzed the impact of the fallback policy. One of the fallback policies we picked described a "careful" fallback, which preferred to move toward states that are further away from the cliff. Here, we give the precise definition of this policy. Denote by cliff the set of cell coordinates which are part of the cliff, i.e., $\mathsf{cliff} \coloneqq \{(r, c) \in \{0, \dots, 10\}^2 \mid c = 5 \wedge r \neq 5\}$. Then, the fallback's value or desirability of every cell can be summarized by a matrix $V \in \mathbb{R}^{11\times 11}$ with components

$$v_{i,j} \coloneqq \begin{cases} \exp\left(\frac{1}{2}\min_{(r,c)\in\mathsf{cliff}}(|i-r| + |j-c|)\right) & (r, c) \notin \mathsf{cliff}, \\ -\infty & (r, c) \in \mathsf{cliff}. \end{cases}$$

This matrix is visualized in Figure 6 below.

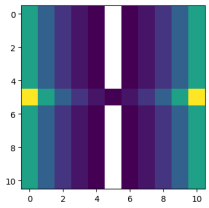


Figure 6: Visualization of V.

Then, for each state $s \in S$, the log probs over the four actions are set to the value of the cell reached by playing that action. If the agent's position would not change (because they are moving into a wall), the log probability for that action is set to 0.

B Proofs

We prove all statements for a general decay function $\lambda\colon \mathbb{N} \to [0, 1]$ with $\lambda_k \geq \lambda_{k+1}$ for every $k \in \mathbb{N}$. Recall that

- $V^{(k)}_{\text{stop}} \coloneqq \lambda_k \max_{a\in A} Q^{(k+1)}(s, a) + (1 - \lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k+1)}(s, a)]$,
- $V^{(k)}_{\text{cont}} \coloneqq \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k)}(s, a)]$, and
- $V^{(k)} \coloneqq \max\{V^{(k)}_{\text{cont}}(s), V^{(k)}_{\text{stop}}(s)\}$.

B.1 Existence of Critical States

The first result links the critical state sets to the value functions at the different stages.

Lemma 1. *Let $k \in \mathbb{N}$. C_k is empty if, and only if, $V^{(k)} = V^{\rho}$.*

Proof. Suppose that C_k is empty. For every $s \in S$, this implies that an optimal solution chooses "continue" in stage k. It follows that

$$\begin{aligned} V^{(k)}(s) &= \max(V^{(k)}_{\text{stop}}(s), V^{(k)}_{\text{cont}}(s)) = V^{(k)}_{\text{cont}}(s) \\ &= \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s, a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^{(k)}(s')]]. \end{aligned}$$

In other words, $V^{(k)}$ satisfies the Bellman equation for ρ. Since it has a unique solution, it follows that $V^{(k)} = V^{\rho}$.

On the other hand, if $V^{(k)} = V^{\rho}$, the same equation tells us that $V^{(k)}(s) = V^{(k)}_{\text{cont}}(s)$ in every state. This, in turn, shows that no state is k-critical. □

Next, we show a key property of the cascade: the value functions are monotonically non-increasing in k.

Lemma 2. *For any $k \in \mathbb{N}$, $V^{(k)} \geq V^{(k+1)} \geq V^{\rho}$.*

Proof. For the second inequality, observe that the intervention policy that plays NOOP in every augmented state has a value of V^ρ. Because $V^{(k+1)}$ is the value of an optimal policy, it cannot be worse than V^ρ.

Next, we prove the first inequality. Without loss of generality, we assume that $\lambda_{k+1} > 0$. If not, then the environment rejects every intervention in stage $k+1$ and onward, such that $V^{(k+1)} = V^\rho$. By the second inequality, $V^{(k)} \geq V^\rho = V^{(k+1)}$.

We use the deterministic optimal policy $\pi^\star$ of $\mathcal{M}_{\text{crit}}$, which produces a value of $V^{(k+1)}$ in stage $k+1$, to construct a policy π that achieves the same value in stage k. Because $\lambda_k \geq \lambda_{k+1}$, we can simulate a rejection by having the intervention policy sample its interventions according to ρ. Thus, we define $\pi(\text{NOOP} \mid s, k) = 1$ if and only if $\pi^\star(s, k+1) = \text{NOOP}$, and $\pi(\tilde{a} \mid s, k) = \frac{\lambda_{k+1}}{\lambda_k}\pi^\star(\tilde{a} \mid s, k+1) + (1 - \frac{\lambda_{k+1}}{\lambda_k})\rho(\tilde{a} \mid s)$ in all remaining states.

Inserting this into the effective policy from equation 1, we find for $a \in A$:

$$\pi_{\text{eff}}(a \mid s, k) = \lambda_k \left(\frac{\lambda_{k+1}}{\lambda_k}\pi^\star(a \mid s, k+1) + \left(1 - \frac{\lambda_{k+1}}{\lambda_k}\right)\rho(a \mid s) \right) + (1 - \lambda_k)\rho(a \mid s) \tag{6}$$

$$= \lambda_{k+1}\pi^\star(a \mid s, k+1) + (1 - \lambda_{k+1})\rho(a \mid s), \tag{7}$$

which is the same as if $\pi^\star$ had been played in stage $k+1$. If $\tilde{a} = \text{NOOP}$, it follows directly from the definition of π that the effective policy behaves the same. □

The following corollary extends the monotonicity of the value functions to the state-action value functions, the stopping values and the continuation values.

Corollary 1. *Let $k \in \mathbb{N}$, $s \in S$ and $a \in A$ be arbitrary. We have*

- $Q^{(k)}(s, a) \geq Q^{(k+1)}(s, a)$,
- $V^{(k)}_{\text{stop}}(s) \geq V^{(k+1)}_{\text{stop}}(s)$,
- $V^{(k)}_{\text{cont}}(s) \geq V^{(k+1)}_{\text{cont}}(s)$.

Proof. From Lemma 2, it follows that

$$Q^{(k)}(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot \mid a, s)}[\underbrace{V^{(k)}(s')}_{\leq V^{(k+1)}(s')}] \leq Q^{(k+1)}(s, a).$$

Then, we also have

$$V^{(k)}_{\text{cont}}(s) = \mathbb{E}_{a \sim \rho(\cdot \mid s)}[\underbrace{Q^{(k)}(s, a)}_{\leq Q^{(k+1)}(s,a)}] \leq V^{(k+1)}_{\text{cont}}(s).$$

And finally,

$$V^{(k)}_{\text{stop}}(s) = \lambda_k \max_{a \in A} \underbrace{Q^{(k+1)}(s, a)}_{\leq Q^{(k+2)}(s,a)} + (1 - \lambda_k) \underbrace{V^{(k+1)}_{\text{cont}}(s)}_{\leq V^{(k+2)}_{\text{cont}}(s)} \leq V^{(k+1)}_{\text{stop}}(s).$$

□

We are now ready to prove the main result.

Theorem 1. *Let $\lambda \colon \mathbb{N} \to [0, 1]$ be a monotonically decreasing function with $\lambda_0 > 0$. Then, $C_k(\rho, \lambda)$ is non-empty for all $k \in \mathbb{N}$ with $\lambda_k > 0$ if, and only if, ρ is not optimal.*

Proof. Suppose that ρ is not optimal, but there exists some $k \in \mathbb{N}$ such that C_k is empty. By Lemma 1, this implies that $V^{(k)} = V^\rho$. By Lemma 2, it follows that $V^\rho = V^{(k)} \leq V^{(k+1)} \leq V^\rho$, also

implying $V^{(k+1)} = V^\rho$ and subsequently $Q^{(k+1)} = Q^\rho$. Since C_k is empty, we have for every state $s \in S$,

$$\begin{aligned}
&V^{(k)}_{\text{stop}}(s) \leq V^{(k)}_{\text{cont}}(s) \\
\iff &\lambda_k \max_{a \in A} Q^{(k+1)}(s,a) + (1-\lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k+1)}(s,a)] \leq \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k)}(s,a)] \\
\iff &\lambda_k \max_{a \in A} Q^{\rho}(s,a) + (1-\lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{\rho}(s,a)] \leq \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{\rho}(s,a)] \\
\iff &\lambda_k \max_{a \in A} Q^{\rho}(s,a) \leq \lambda_k V^\rho(s) \overset{\lambda_k>0}{\iff} \max_{a \in A} Q^{\rho}(s,a) \leq V^\rho(s).
\end{aligned}$$

Because this holds for all $s \in S$, it implies that ρ is already optimal, contradicting our initial assumption.

For the other direction, suppose that ρ is optimal. It follows that $V^\rho = V^\star$, and by Lemma 2, we have $V^{(k)} \geq V^\rho = V^\star$ for any $k \in \mathbb{N}$. But we also have $V^\star \geq V^{(k)}$ by definition of the optimal value. We conclude that $V^{(k)} = V^\star = V^\rho$, and by Lemma 1, C_k is empty for all $k \in \mathbb{N}$. □

B.2 Reward Scale-Invariance

Recall that, for $c > 0$, we denote by R_c the reward function derived from R via the transformation $R_c \coloneqq cR$.

Proposition 3 (Reward scale invariance)**.** *Fix λ and ρ. Let $c, c' > 0$. Then, $C_k(\mathcal{M}_c) = C_k(\mathcal{M}_{c'})$ for every $k \in \mathbb{N}$.*

Proof. Since $V^{(k)} = V^\star_{\text{crit}}(\cdot, k)$, we can argue using $\mathcal{M}_{\text{crit}}$. First, observe from equation 5 that if R is replaced by R_c in $\mathcal{M}$, then $\widetilde{R}$ becomes $c\widetilde{R}$, which we denote by $\widetilde{R}_c$ from now on. Following a standard argument, one can show that $\pi^\star_{\text{crit}}$ remains optimal under the reward transformation. Value functions with a subscript c refer to those in the reward-transformed MDPs.

Therefore,

$$\begin{aligned}
V^{(k)}_c(s) &= \mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty} \gamma^t \widetilde{R}_c((s_t,k_t),\widetilde{a}_t) \mid (s_0,k_0) = (s,k)\right] \\
&= c\mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty} \gamma^t \widetilde{R}((s_t,k_t),\widetilde{a}_t) \mid (s_0,k_0) = (s,k)\right] \\
&= cV^{(k)}(s),
\end{aligned}$$

due to the fact that $\gamma < 1$, which causes the sum to converge, and due to linearity of the expectation.

It follows that

$$\begin{aligned}
Q^{(k)}_c(s,a) &= cR(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^{(k)}_c(s')] \\
&= c(R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^{(k)}(s')]) \\
&= cQ^{(k)}(s,a),
\end{aligned}$$

which in turn implies $V^{(k)}_{c,\text{stop}}(s) = cV^{(k)}_{\text{stop}}(s)$ and $V^{(k)}_{c,\text{cont}}(s) = cV^{(k)}_{\text{cont}}(s)$, since

- $\mathbb{E}_x[cx] = c\mathbb{E}_x[x]$,
- $\max\{cx, cy\} = c\max\{x, y\}$ and
- $\lambda(cx) + (1-\lambda)(cy) = c(\lambda x + (1-\lambda)y)$

for any $x, y \in \mathbb{R}$, $c > 0$ and $\lambda \in [0, 1]$.

We can conclude that

$$V^{(k)}_{c,\text{stop}}(s) > V^{(k)}_{c,\text{cont}}(s) \iff cV^{(k)}_{\text{stop}}(s) > cV^{(k)}_{\text{cont}}(s) \overset{c>0}{\iff} V^{(k)}_{\text{stop}}(s) > V^{(k)}_{\text{cont}}(s).$$

□

Remark 1. If $\mathcal{M}$ contains no terminal states that can cause trajectories to end prematurely, the invariance can be shown to also extend to additive shifts in the reward function, since the additive shift $d \in \mathbb{R}$ can be pulled out of the sum and expectation, yielding an additive term of $\frac{d}{1-\gamma}$ in every value function. The presence of terminal states changes trajectory lengths, which in turn can affect $\pi^\star_{\text{crit}}$.

B.3 Optimal Stopping and Convergence of Cascaded Value Iteration

First, we prove the result which reframes CODA's optimization problem as a cascade of optimal stopping problems.

Proposition 1. *We have $V^{(K)} = V^\rho$. For $k \in \{0, \dots, K-1\}$, the function $V^{(k)}$ is the solution of an optimal stopping problem where the value of stopping is*

$$V^{(k)}_{\text{stop}}(s) \coloneqq \lambda_k \max_{a \in A} Q^{(k+1)}(s,a) + (1-\lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k+1)}(s,a)], \tag{3}$$

and the continuation value is

$$V^{(k)}_{\text{cont}}(s) \coloneqq \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k)}(s,a)]. \tag{4}$$

Consequently, we have $V^{(k)} = \max\{V^{(k)}_{\text{cont}}, V^{(k)}_{\text{stop}}\}$ and the critical state sets can be expressed as $C_k = \{s \in S \mid V^{(k)}_{\text{stop}}(s) > V^{(k)}_{\text{cont}}(s)\}$.

Proof. $V^{(K)} = V^\rho$ follows from Lemma 1, since $\lambda_K = 0$ by definition of the cutoff.

To prove the main statement, we make use of the augmented MDP definition given in Appendix A and rewrite its Bellman equation. The Bellman equation for $V^\star_{\text{crit}}$ at a state s in stage $k \in \{0, \dots, K-1\}$ reads

$$V^\star_{\text{crit}}(s,k) = \max_{\tilde{a}\in\widetilde{A}} \widetilde{R}((s,k),\tilde{a}) + \gamma \mathbb{E}_{(s',k')\in\widetilde{P}(\cdot|(s,k),\tilde{a})}[V^\star_{\text{crit}}(s',k')].$$

If the maximizer is $\tilde{a} = \text{NOOP}$, then

$$\begin{aligned} V^\star_{\text{crit}}(s,k) &= \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a)] + \gamma \mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V^\star_{\text{crit}}(s',k)] \\ &= \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|a,s)}[V^{(k)}(s')]] \\ &= \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k)}(s,a)] = V^{(k)}_{\text{cont}}(s). \end{aligned}$$

Otherwise, if the maximizer is some $\tilde{a} \in A$, then it follows, again from the definitions of $\widetilde{P}$ and $\widetilde{R}$ and a bit of rewriting, that

$$\begin{aligned} V^\star_{\text{crit}}(s,k) &= \max_{\tilde{a}\in A}\{\lambda_k R(s,\tilde{a}) + (1-\lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a)] \\ &\qquad + \gamma(\lambda_k \mathbb{E}_{s'\sim P(\cdot|\tilde{a},s)}[V^\star_{\text{crit}}(s',k+1)] + (1-\lambda_k)\mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V^\star_{\text{crit}}(s',k+1)])\} \\ &= \lambda_k \max_{\tilde{a}\in A}\Big\{R(s,\tilde{a}) + \gamma \mathbb{E}_{s'\sim P(\cdot|\tilde{a},s)}[V^{(k+1)}(s')]\Big\} \\ &\qquad + (1-\lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|a,s)}[V^{(k+1)}(s')]] \\ &= \lambda_k \max_{\tilde{a}\in A} Q^{(k+1)}(s,\tilde{a}) + (1-\lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k+1)}(s,a)] = V^{(k)}_{\text{stop}}(s). \end{aligned}$$

Together, this is expressed as $V^\star(s,k) = \max\{V^{(k)}_{\text{stop}}(s), V^{(k)}_{\text{cont}}(s)\}$.

Finally, the characterization of C_k. If $V^{(k)}_{\text{stop}}(s) > V^{(k)}_{\text{cont}}(s)$, then no deterministic optimal policy $\pi^\star_{\text{crit}}$ would play NOOP. If $V^{(k)}_{\text{stop}}(s) = V^{(k)}_{\text{cont}}(s)$, swapping out the optimal action and NOOP makes no difference, so the condition in the definition of C_k is not fulfilled. Hence the strict inequality. □

The resulting algorithm, the cascaded value iteration is shown in Algorithm 2. It iterates backwards from $k = K-1$ to $k = 0$, invoking a value iteration-based optimal stopping problem solver, OSVI (Algorithm 1), at each step. The latter receives an MDP, a fallback policy, a stopping value

function, an initial guess V_0 and a precision $\varepsilon > 0$, and iterates the Bellman optimality equation until $\|V_n - V_{n-1}\|_\infty < \varepsilon$.[3]

Algorithm 1 Solve optimal stopping problem

Input: MDP $\mathcal{M}$, fallback ρ, stopping values $V_{\text{stop}} \colon S \to \mathbb{R}$, initial guess V_0
Output: value function V, stopping states $C \subset S$

```
V ← V_0
C ← ∅
while not converged do
    for s ∈ S do
        V_cont(s) ← Σ_{a∈A} ρ(a | s) Q(s, a)
        V(s) ← max{V_cont(s), V_stop(s)}
        if V_stop(s) > V_cont(s) then
            C ← C ∪ {s}
        else
            C ← C \ {s}
        end if
    end for
end while
return V, C
```

Algorithm 2 Cascaded Value Iteration (CVI)

Input: MDP $\mathcal{M}$, fallback ρ, decay function λ, threshold $\tau \in (0, 1)$ (default $\tau = 0.01$)
Output: Critical states $C_k \subset S$ w.r.t. ρ

```
K ← ⌈−ln(τ)/β⌉
V^(K) ← V^ρ
k ← K − 1
repeat
    Q^(k+1) ← L V^(k+1)
    for s ∈ S do
        V_acc(s) ← max_{a∈A} {Q^(k+1)(s, a)}
        V_rej(s) ← E_{a∼ρ(·|s)} [Q^(k+1)(s, a)]
    end for
    V_stop ← λ_k V_acc + (1 − λ_k) V_rej
    V^(k), C_k ← OSVI(M, ρ, V_stop, V^(k+1))
    k ← k − 1
until k = 0
return {C_k}_{k=0}^K
```

Finally, we prove the convergence result of the cascaded value iteration, shown in Algorithm 2. In the proof, we denote approximations of value functions with a tilde, while maintaining the same sub- and superscripts.

Proposition 2 (Error propagation)**.** *The approximation $\widetilde{V}^{(k)}$ of the stage-k value function $V^{(k)}$ computed by the cascaded value iteration satisfies $\|\widetilde{V}^{(k)} - V^{(k)}\|_\infty \leq \frac{\varepsilon\gamma(1-\gamma^{K-k+1})}{(1-\gamma)^2}$ for every $k \in \{0, \ldots, K-1\}$, provided that $\gamma < 1$.*

[3]Note that V_n is different from $V^{(k)}$. The index n refers to the number of iterations in the value iteration, while the k refers to the stage in our framework.

Proof. At each stage k, the Q-values resulting from OSVI are combined into an (approximate) stopping value $\widetilde{V}_{\text{stop}}^{(k)}$, which has an error of

$$\begin{aligned}\mathcal{E}_{\text{stop}}^{(k)} &:= \|\widetilde{V}_{\text{stop}}^{(k)} - V_{\text{stop}}^{(k)}\|_\infty \\ &\leq \max_{a\in A}\|\widetilde{Q}^{(k+1)}(\cdot,a) - Q^{(k+1)}(\cdot,a)\|_\infty \\ &\leq \gamma \underbrace{\|\widetilde{V}^{(k+1)} - V^{(k+1)}\|_\infty}_{=:\,\mathcal{E}^{(k+1)}}.\end{aligned}$$

On the other hand, we can also show that $\mathcal{E}^{(k)} \leq \mathcal{E}_{\text{stop}}^{(k)} + \frac{\epsilon\gamma}{1-\gamma}$: For ease of reading, we temporarily drop the superscript (k) from the value functions. Since the stopping values are again fed into Algorithm 1, the result has two sources of error: first, the error in the stopping values, and secondly, the approximation error in the computed value function. To separate the errors clearly in the notation, we distinguish between (i) $V^\star$, the optimal value given exact stopping values, (ii) $\widetilde{V}^\star$, the optimal value given approximate stopping values, and (iii) $\widetilde{V}_n$, the approximately optimal value after n iterations, given approximate stopping values. By the triangle inequality,

$$\mathcal{E}^{(k)} = \|V^\star - \widetilde{V}_n\|_\infty \leq \|V^\star - \widetilde{V}^\star\|_\infty + \underbrace{\|\widetilde{V}^\star - \widetilde{V}_n\|_\infty}_{\leq\, \varepsilon\gamma/(1-\gamma)}. \tag{8}$$

The second term is bounded by $\varepsilon\gamma/(1-\gamma)$, because the value iteration stops as soon as $\|\widetilde{V}_n - \widetilde{V}_{n-1}\| < \varepsilon$. This is a standard bound from the MDP literature Puterman (1994).

We now establish an upper bound for the first term on the right-hand side of equation 8. Consider the state s in which the maximum error $\|V^\star - \widetilde{V}^\star\|_\infty$ occurs. We distinguish several cases:

1. $V_{\text{stop}}(s) \leq V_{\text{cont}}(s)$: We further divide this into two subcases:
   (a) If $V_{\text{cont}}(s) \geq \widetilde{V}^\star(s)$, then
   $$\begin{aligned}\|V^\star - \widetilde{V}^\star\|_\infty &= V_{\text{cont}}(s) - \widetilde{V}^\star(s) \\ &\leq V_{\text{cont}}(s) - \widetilde{V}_{\text{cont}}(s) \\ &= \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^\star(s,a) - \widetilde{Q}^\star(s,a)] \\ &= \gamma\mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V^\star(s') - \widetilde{V}^\star(s')] \leq \gamma\|V^\star - \widetilde{V}^\star\|_\infty,\end{aligned}$$
   which for $\gamma < 1$ implies that $V^\star = \widetilde{V}^\star$, i.e., $\|V^\star - \widetilde{V}^\star\|_\infty = 0$.
   (b) If $V_{\text{cont}}(s) < \widetilde{V}^\star(s)$, then
   $$\|V^\star - \widetilde{V}^\star\|_\infty = \widetilde{V}^\star(s) - V_{\text{cont}}(s) \leq \max\{\widetilde{V}_{\text{cont}}(s) - V_{\text{cont}}(s), \widetilde{V}_{\text{stop}}(s) - V_{\text{cont}}(s)\}.$$
   The first case in the "max" would imply that $V^\star = \widetilde{V}^\star$, as above. Assuming the other case, we have
   $$\|V^\star - \widetilde{V}^\star\|_\infty \leq \widetilde{V}_{\text{stop}}(s) - V_{\text{cont}}(s) \leq \widetilde{V}_{\text{stop}}(s) - V_{\text{stop}}(s) \leq \mathcal{E}_{\text{stop}}^{(k)},$$
   where the second equality follows from the fact that the first case in the "max" would imply that $V^\star = \widetilde{V}^\star$, as above.
2. $V_{\text{stop}}(s) > V_{\text{cont}}(s)$: We again divide this into two subcases:
   (a) If $V_{\text{stop}}(s) \geq \widetilde{V}^\star(s)$, then
   $$\|V^\star - \widetilde{V}^\star\|_\infty = V_{\text{stop}}(s) - \widetilde{V}^\star(s) \leq V_{\text{stop}}(s) - \widetilde{V}_{\text{stop}}(s) \leq \mathcal{E}_{\text{stop}}^{(k)}.$$
   (b) If $V_{\text{stop}}(s) < \widetilde{V}^\star(s)$, then
   $$\begin{aligned}\|V^\star - \widetilde{V}^\star\|_\infty &= \widetilde{V}^\star(s) - V_{\text{stop}}(s) \leq \max\{\widetilde{V}_{\text{cont}}(s) - V_{\text{stop}}(s), \widetilde{V}_{\text{stop}}(s) - V_{\text{stop}}(s)\} \\ &\leq \max\{\widetilde{V}_{\text{cont}}(s) - V_{\text{cont}}(s), \widetilde{V}_{\text{stop}}(s) - V_{\text{stop}}(s)\} \leq \mathcal{E}_{\text{stop}}^{(k)}.\end{aligned}$$
   Once again, the first case in the "max" would have led to $V^\star = \widetilde{V}^\star$.

Each case implies that either $V^\star = \widetilde{V}^\star$, or $\|V^\star - \widetilde{V}^\star\|_\infty \leq \mathcal{E}^{(k)}_{\text{stop}}$. Since the first implies the latter, we can conclude that $\|V^\star - \widetilde{V}^\star\|_\infty \leq \mathcal{E}^{(k)}_{\text{stop}}$ always holds. With equation 8, this implies

$$\mathcal{E}^{(k)} \leq \mathcal{E}^{(k)}_{\text{stop}} + \frac{\varepsilon\gamma}{1-\gamma} \leq \gamma\mathcal{E}^{(k+1)} + \frac{\varepsilon\gamma}{1-\gamma} = \gamma\left(\mathcal{E}^{(k+1)} + \frac{\varepsilon}{1-\gamma}\right).$$

Given that $\mathcal{E}^{(K)} = \frac{\epsilon\gamma}{1-\gamma}$ by assumption, we find that, for any $k \in \{0, \ldots, K-1\}$,

$$\mathcal{E}^{(k)} \leq \frac{\varepsilon\gamma(1+\gamma+\cdots+\gamma^{K-k})}{1-\gamma} = \frac{\varepsilon\gamma(1-\gamma^{K-k+1})}{(1-\gamma)^2}.$$

□

Remark 2. To keep error propagation in check, one has to either lower ε or γ. Lowering γ generally decreases convergence time at the cost of long-term accuracy, while lowering ε increases convergence time. For example, when $\gamma = 0.99$, one needs to set $\varepsilon \leq 10^{-5}$ to ensure errors smaller than 10^{-1}.

Because of the error bound, misclassification in stage k can occur, if the true stop/continue values are closer than $2\gamma\mathcal{E}^{(k)}$. To mitigate this, we can change the condition for criticality using the approximate values to

$$\widetilde{V}^{(k)}_{\text{stop}}(s) > \widetilde{V}^{(k)}_{\text{cont}}(s) + \frac{2\varepsilon\gamma^2}{(1-\gamma)^2},$$

where we have taken the limit $K \to \infty$ for simplicity. This way, one might find fewer critical state sets, but there is a guarantee that every discovered critical state is considered critical even under the true values.

B.4 Connection to Lazy-MDP

Both CODA and Lazy-MDPs employ the same principle. However, whereas Lazy-MDPs modify the reward function directly by applying a penalty when not deferring, CODA modifies the transition kernel instead. The downstream effect is still a modification of the long-term reward that the agent receives, but it is no longer explicit. Additionally, the penalty may depend on the state and action. By slightly extending the lazy-MDP framework and allowing state- and action-dependent penalties $\eta(s,a) \geq 0$, we can show that CODA, in a special case, is equivalent to solving the lazy-MDP with a particular penalty function.

Proposition 4. *Let $\lambda = \mathbb{I}\{k=0\}$. Then, the set of optimal policies under CODA is the same as the set of optimal policies under the Lazy-MDP framework with penalty $\eta(s,a) = Q^{(0)}(s,a) - Q^\rho(s,a)$, where $Q^{(0)}$ is the stage-0 action-value function of CODA.*

Proof. From the perspective of an agent in CODA, the Bellman optimality equation is

$$V^{(0)}(s) = \max\{\max_{a\in A} Q^\rho(s,a), \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(0)}(s,a)]\}. \tag{9}$$

Meanwhile, from the perspective of an agent in the Lazy-MDP framework, the Bellman optimality equation reads Jacq et al. (2022)

$$V'(s) = \max\{\max_{a\in A} Q'(s,a) - (Q^{(0)}(s,a) - Q^\rho(s,a)), \mathbb{E}_{a\sim\rho(\cdot|s)}[Q'(s,a)]\}. \tag{10}$$

If we simply replace Q' by $Q^{(0)}$ in equation 10 (analogously with V' and $V^{(0)}$), we arrive at equation 9. In other words, any solution of equation 9 is also a solution of equation 10. Since both problems have a unique solution, the converse must be true as well. Therefore, both optimization problems are equivalent and thus share the same set of optimal policies. □

Remark 3. We speak of the "set of optimal policies" rather than the "optimal policy". While the optimal value function is unique, there can be multiple policies with this value function, in the case of value ties. In CODA, we always break ties by deferring. If we adopt the same convention for the Lazy-MDP framework, the optimal policy is always unique and the critical state sets will match.

Remark 4. When $K > 1$, an analogous result cannot be derived without performing a state space expansion that keeps track of the number of non-defer actions in Lazy-MDP. Still, it is possible to assign a value to the penalty: just like $\eta = Q(s,a) - (Q(s,a) - \eta)$ can be expressed as the difference of penalized and non-penalized action values, we can do the same with CODA by recognizing that the implicit penalty consists in the stage advancement from k to $k+1$:

$$\eta^{(k)}(s,a) = \underbrace{\lambda_k Q^{(k)}(s,a) + (1-\lambda_k)V^{(k)}_{\text{cont}}(s)}_{\text{no penalty}} - \underbrace{\lambda_k Q^{(k+1)}(s,a) + (1-\lambda_k)V^{(k+1)}_{\text{cont}}(s)}_{\text{with penalty}}$$
$$= \lambda_k(Q^{(k)}(s,a) - Q^{(k+1)}(s,a)) + (1-\lambda_k)(V^{(k)}_{\text{cont}}(s) - V^{(k+1)}_{\text{cont}}(s)) \geq 0,$$

where the non-negativity follows from the monotonicity of the value functions, as shown in Corollary 1.

B.5 Degree of Endogeneity

Denote by $V^\star_L$ the optimal value function of the lazy MDP under consideration.

Fix a state $s \in S$. Let us define

$$f_V(s,a) \coloneqq \gamma(\mathbb{E}_{s'\sim P(\cdot|s,a)}[V(s')] - \mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V(s')])$$

for each $a \in A$ and value function $V\colon S \to \mathbb{R}$. It quantifies the downstream advantage of choosing a over the baseline, ignoring the immediate reward. Observe that $\mathbb{E}_{a\sim\rho(\cdot|s)}[f_V(s,a)] = 0$. Furthermore, we define

$$G_V(s) \coloneqq \max_{a\in A} R(s,a) + f_V(s,a) - \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a) + f_V(s,a)]$$
$$= \max_{a\in A} R(s,a) + f_V(s,a) - \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a)].$$

This is the same reward statistic as $G(s)$ from Section 4.5, except that the reward is now augmented by a value function-dependent bonus, f_V.

Proposition 6. *Let $s \in S$ be an arbitrary state. Then,*

- $C_{\text{SI}}(q) = \{s \in S : \text{Var}_a[R(s,a) + f_{V^\star}(s,a)] > Q(q)\}$,
- $C_{\text{lazy}}(\rho,\eta) = \{s \in S : G_{V^\star_L}(s) > \eta\}$,
- $C^{(k)}_{\text{CODA}}(\rho,\lambda) = \{s \in S : \lambda_k G_{V^{(k+1)}}(s) > \gamma\mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V^{(k)}(s') - V^{(k+1)}(s')]\}$.

Proof. SI:

$$\text{Var}_a[Q^\star(s,a)] = \text{Var}_a[R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^\star(s')]]$$
$$= \text{Var}_a[R(s,a) + f_{V^\star}(s,a) + \gamma\mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V^\star(s')]]$$
$$= \text{Var}_a[R(s,a) + f_{V^\star}(s,a)]$$

Lazy-MDP:

$$\max_{a\in A} R(s,a) - \eta + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^\star_L(s')] > \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}[V^\star_L(s')]]$$
$$\max_{a\in A} R(s,a) - \eta + f_{V^\star_L}(a) > \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a)]$$
$$G_{V^\star_L}(s) > \eta.$$

CODA:

$$V^{(k)}_{\text{stop}}(s) > V^{(k)}_{\text{cont}}(s)$$
$$\iff \quad \lambda_k \max_{a\in A} Q^{(k+1)}(s,a) + (1-\lambda_k)\mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k+1)}(s,a)] > \mathbb{E}_{a\sim\rho(\cdot|s)}[Q^{(k)}(s,a)]$$
$$\iff \quad \lambda_k(\max_{a\in A}\{R(s,a) + f_{V^{(k+1)}}(s,a)\} - \mathbb{E}_{a\sim\rho(\cdot|s)}[R(s,a)]) > \gamma\mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V^{(k)}(s') - V^{(k+1)}(s')]$$
$$\iff \quad \lambda_k G_{V^{(k+1)}}(s) > \gamma\mathbb{E}_{s'\sim P^\rho(\cdot|s)}[V^{(k)}(s') - V^{(k+1)}(s')].$$

□

Next, we will tie the above result a some notion of the "degree of endogeneity" of a state. To this end, let us define $\kappa_\rho(s) \coloneqq \max_{a \in A} \mathrm{TV}(P(\cdot \mid s,a), P^\rho(\cdot \mid s))$ and the ρ-free variant used in the main text, $\kappa(s) \coloneqq \max_{a,a' \in A} \mathrm{TV}(P(\cdot \mid s,a), P(\cdot \mid s,a'))$. It is evident that $\kappa_\rho \leq \kappa$ pointwise. We will also define the *span* of a bounded function $V \colon S \to \mathbb{R}$ as $\mathrm{span}(V) \coloneqq \sup_{s \in S} V(s) - \inf_{s \in S} V(s)$.

Lemma 3. *Let $V \colon S \to \mathbb{R}$ be a bounded value function and $s \in S$. Then,*

$$|G_V(s) - G(s)| \leq \|f_V(s,\cdot)\|_\infty \leq \gamma \kappa_\rho(s) \mathrm{span}(V).$$

Proof. For the first inequality,

$$|G_V(s) - G(s)| = |\max_{a \in A}\{R(s,a) + f_V(s,a)\} - \max_{a \in A} R(s,a)| \leq \|f_V(s,\cdot)\|_\infty$$

due to the fact that the maximum function is 1-Lipschitz in $\|\cdot\|_\infty$.

The second inequality follows from observing that f_V is a (scaled) difference of expectations of the same function V under different distributions. To bound this, subtract the midpoint m of $\mathrm{span}(V)$ to obtain $U = V - m$ which then satisfies $\|U\|_\infty = \frac{1}{2}\mathrm{span}(V)$. Following proposition 4.5 of Levin et al. (2017), we have

$$\begin{aligned}\frac{\|f_V(s,\cdot)\|_\infty}{\|U\|_\infty} &= \frac{\gamma}{\|U\|_\infty} \max_{a \in A} |\mathbb{E}_{s' \sim P(\cdot|s,a)}[U(s')] - \mathbb{E}_{s' \sim P^\rho(\cdot|s)}[U(s')]| \\ &\leq \gamma 2 \mathrm{TV}(P(\cdot \mid s,a), P^\rho(\cdot \mid s)),\end{aligned}$$

so that $\|f_V(s,\cdot)\|_\infty \leq \gamma \kappa_\rho(s) \mathrm{span}(V)$. □

Recall that we defined $\varepsilon(s) = \gamma \kappa(s)(R_{\max} - R_{\min})/(1-\gamma) \geq 0$ in the main text. Observe that $\mathrm{span}(V) \leq (R_{\max} - R_{\min})/(1-\gamma)$ holds for any value function in an MDP with reward function R. Thus, the bound from the previous Lemma can be loosened to $|G_V(s) - G(s)| \leq \varepsilon(s)$. We can now prove Proposition 5 from the main text.

Proposition 5 (Local decidability)**.** *Let $s \in S$ and $k \in \mathbb{N}$ with $\lambda_k > 0$. Then:*

(i) $\left|\sqrt{\mathsf{SI}(s)} - \mathrm{std}_a R(s,\cdot)\right| \leq \varepsilon(s)$*;*

(ii) Lazy-MDP: $G(s) > \eta + \varepsilon(s)$ implies $s \in C_{\mathrm{lazy}}$, and $G(s) < \eta - \varepsilon(s)$ implies $s \notin C_{\mathrm{lazy}}$;

(iii) CODA: $\lambda_k G(s) > \Omega_k(s) + \varepsilon(s)$ implies $s \in C_k$, and $\lambda_k G(s) < \Omega_k(s) - \varepsilon(s)$ implies $s \notin C_k$.

In particular, if s is exogenous, then $\varepsilon(s) = 0$ and (i)–(iii) hold with equality resp. as equivalences.

Proof. From Proposition 6, recall that $\mathsf{SI}(s) = \mathrm{Var}_a[R(s,a) + f_{V^\star}(s,a)]$. Taking the square root on both sides, we can exploit the subadditivity of the standard deviation and obtain

$$\sqrt{\mathsf{SI}(s)} - \mathrm{std}_a R(s,\cdot) = \mathrm{std}_a[R(s,\cdot) + f_{V^\star}(s,\cdot)] - \mathrm{std}_a R(s,\cdot) \leq \mathrm{std}_a f_{V^\star}(s,\cdot).$$

Since $\mathrm{Var}_a[f_V(s,\cdot)] \leq \|f_V(s,\cdot)\|_\infty^2$, we get the desired bound using Lemma 3.

For Lazy-MDP, Lemma 3 yields $|G_{V_L^\star}(s) - G(s)| \leq \varepsilon(s)$, i.e., $G(s) - \varepsilon(s) \leq G_{V_L^\star}(s) \leq G(s) + \varepsilon(s)$. If $G(s) > \eta + \varepsilon(s)$, then $G_{V_L^\star}(s) \geq G(s) - \varepsilon(s) > \eta$, so $s \in C_{\mathrm{lazy}}$ by Proposition 6. If $G(s) < \eta - \varepsilon(s)$, then $G_{V_L^\star}(s) \leq G(s) + \varepsilon(s) < \eta$, so $s \notin C_{\mathrm{lazy}}$.

Statement (iii) can be shown in an analogous manner. □

C Further Details on Deep CODA

C.1 Architecture

In this section, we give further details on the architecture of Deep CODA.

C.1.1 VALUE NETWORK

Our value network consists of a *backbone* $b_\phi \colon \mathbb{R}^{d_s} \to \mathbb{R}^{d_h}$ which computes state features. These are passed to both an *anchor head* $v^\rho_\phi \colon \mathbb{R}^{d_h} \to \mathbb{R}$, which is an affine-linear function, as well as *increment heads* $v^{(k)}_\phi \colon \mathbb{R}^{d_h} \to \mathbb{R}^2$.

Under the hood, the increment heads are implemented using a single head, which is advantageous if K becomes large. It maps k onto $[0,1]$ via $k \mapsto k/K$ and computes Fourier features $(k/K, \sin(fk/K), \cos(fk/K))$ from it, for some fixed frequencies $f \in \mathbb{R}^{n_f}$. We denote this encoding as $F(k)$.

Both the resulting features and the state features are projected into an embedding space and added together. Finally, a single MLP layer (ReLU + linear) maps them to two outputs. For each k, the two outputs are $\tilde{\delta}^{(k)}_\phi$ and $A^{(k)}_\phi$. Thus, $(\tilde{\delta}^{(k)}_\phi, A^{(k)}_\phi) = W_\phi \cdot \mathsf{ReLU}(p_\phi(F(k)) + q_\phi(h)) + b_\phi$. From the anchor and stage heads, all relevant value outputs can be constructed, as described earlier.

Loss The value net is updated using three losses: First, the *distillation loss*

$$\mathcal{L}_{\text{dist}}(\phi) = \text{MSE}_{s\sim\mathcal{D}}\left(v^\rho_\phi(s), \frac{1}{m}\sum_{j=1}^{m} \overline{q}^\rho_{\theta_{\text{targ}}}(s,a)\right) + \frac{1}{K}\sum_{k=0}^{K-1} \text{MSE}_{s\sim\mathcal{D}}\left(\tilde{\delta}^{(k)}_\phi(s), \frac{1}{m}\sum_{j=1}^{m} \overline{\widetilde{\Delta}}^{(k)}_{\theta_{\text{targ}}}(s,a_j)\right),$$

where $a_j \sim \rho(\cdot \mid s)$. It distills the critic increments to the continuation value increments. Second is the *anchor loss*

$$\mathcal{L}_{\text{anchor}}(\phi) = \text{MSE}_{(s,r,s',m,c)\sim\mathcal{D}_A}\left(v^\rho_\phi(s), r + mcv^\rho_{\phi_{\text{targ}}}(s')\right)$$

which is trained on ρ-strands contained in the anchor buffer. Each tuple (s, r, s', m, c) describes a ρ-strand of varying length n, such that s is the starting state, r is the collected return after n steps, s' is the state reached after n steps, and $c = \gamma^n$. The mask $m \in \{0,1\}$ indicates whether this is a terminal ρ-trajectory, in which case no bootstrapping is required. Finally, there is the *advantage loss*

$$\mathcal{L}_{\text{adv}}(\phi) = \frac{1}{K}\sum_{k=0}^{K-1} \text{MSE}_{s\sim\mathcal{D}}\left(A^{(k)}_\phi, q^{(k+1)}_{\theta_2,\text{targ}}(s, a^{(k)}_{\varphi,\text{targ}}(s)) - \frac{1}{m}\sum_{j=1}^{m} \overline{q}^{(k+1)}_{\theta,\text{targ}}(s,a_j)\right),$$

again with $a_j \sim \rho(\cdot \mid s)$.[4] Its purpose is to learn the stopping advantage. Note that the loss term only evaluates the second critic, while the actor, whose loss is specified down below, is trained to maximize the first critic.

C.1.2 CRITIC

Each critic also has a *backbone* $b_\theta \colon \mathbb{R}^{d_s} \times \mathbb{R}^{d_a} \to \mathbb{R}^{d_h}$, an *anchor head* $q^\rho_\theta \colon \mathbb{R}^{d_h} \to \mathbb{R}$, and *increment heads* $q^{(k)}_\theta \colon \mathbb{R}^{d_h} \to \mathbb{R}$. The latter are implemented identically to the value's increment heads, just that each has only a single output, $\widetilde{\Delta}^{(k)}_\theta$.

Loss Both critics are trained on anchor values and value increments:

$$\mathcal{L}_{\text{anchor}}(\theta) = \text{MSE}_{(s,a,r,s',d)\sim\mathcal{D}}\left(q^\rho_\theta(s,a), r + \gamma(1-d)v^\rho_{\phi_{\text{targ}}}(s')\right)$$

and

$$\mathcal{L}_{\text{incr}}(\theta) = \frac{1}{K}\sum_{k=0}^{K-1} \text{MSE}_{(s,a,r,s',d)\sim\mathcal{D}}\left(\widetilde{\Delta}^{(k)}_\theta(s,a), \gamma(1-d)(v^{(k)}_{\text{cont},\phi_{\text{targ}}}(s) - v^{(k+1)}_{\text{cont},\phi_{\text{targ}}}(s))\right).$$

[4] In our implementation, we re-use the mean Q-function outputs from the distillation loss.

C.1.3 ACTOR

Finally, the actor is an MLP $a_\varphi^{(k)} \colon \mathbb{R}^{d_s} \to \mathbb{R}^{d_a}$, where k is incorporated in the same way as for the value and critic networks. As mentioned before, it is trained to maximize the first critic, so that its loss is

$$\mathcal{L}_{\text{actor}}(\varphi) = -\frac{1}{K}\sum_{k=1}^{K} q_{\theta_1}^{(k)}(s, a_\varphi^{(k)}(s)),$$

where we use the convention $q_\theta^{(K)} = q_\theta^\rho$. Note that $a_\varphi^{(0)}$ is never queried and therefore does not need to be trained.

All non-anchoring loss functions are averaged over $k \in \{0, \dots, K-1\}$. If K is very large, this will increase the runtime. In this case, we would sample random subsets of $\{0, \dots, K-1\}$ without replacement.

C.1.4 TRAINING

Policy rollout / data collection Collection is split between pure ρ-rollouts and *faithful* rollouts. Faithful rollouts randomly sample a stage k and play in $\mathcal{M}_{\text{crit}}$: whenever the state s has positive criticality, a suggestion is made and k increased. At the same time, we track ensemble disagreement: if a state is close to the threshold with high disagreement, then an intervention is forced with high probability. This intervention is always accepted and does not increase the stage. Its purpose is to provide data in regions with high uncertainty.

D FURTHER DETAILS ON EXPERIMENTS

D.1 SPARSITY

Table 1 shows the results of the sparsity evaluation of the different decay families.

D.2 DISCUSSION OF EXPONENTIAL DECAY PARAMETER IN TABULAR ENVIRONMENTS

This section contains a bit more discussion on the critical states in CliffMaze and UnlockPickup.

Sensitivity From Figure 2, we generally observe the trend that the overall number of critical states decreases as β increases and vice-versa. However, this behavior is not perfectly monotonic. This can be seen in the column corresponding to the *minimizer* fallback policy: At $\beta = 1.0$, all but 2 states are considered critical, not counting the final state. However, when lowering β to 0.1, fewer states are considered critical. This behavior may at first glance seem to stand in conflict with our understanding of β but makes sense when considering the trade-off at hand. The agent has to decide whether it is more important to act now at the expense of losing agency, or whether the current level of agency should be maintained. While not shown here, a similar behavior can be observed when looking across different values of k, which controls the initial acceptance probability.

As an example of a slightly more complex, but still discrete, environment, we chose an environment from the MiniGrid suite of environments (Chevalier-Boisvert et al., 2023). In particular, we chose the `UnlockPickup` environment with full observability and a fixed seed. The environment, depicted in Figure 7c, consists of two rooms separated by a door. To succeed, the agent has to collect a key, open the door using the key, and interact with the box in the bottom-right corner. This environment may appear simple but is reasonably challenging for reinforcement learning agents and totals 17 284 states. Figure 7c shows all of the critical states identified by CODA with $\beta = 0.5$ and $\rho = \pi_{\mathsf{unif}}$ along an optimal trajectory. The total number of identified critical states was $|C_0| = 759$, which corresponds to around 4.34% of the entire state space. The runtime of cascaded value iteration was around 34 seconds.

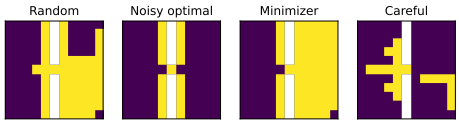
Along the optimal trajectory, states shortly before and after opening the door are deemed critical, as well as the state right after dropping off the key and some states before reaching the goal. Interestingly, the state just before picking up the key is not deemed critical, but instead the state after the key has been picked up and the agent has rotated (top left). A possible cause for this might be that the

Environment	ρ	K	Exp. (%)	Linear (%)	Step (%)
CliffMaze ($\|S\| = 110$)	uniform	1	**20.91**	**20.91**	**20.91**
		2	**20.91**	22.73	23.64
		3	**22.73**	25.45	27.27
		5	**23.64**	28.18	35.45
		10	**26.36**	38.18	55.45
		20	**33.64**	52.73	100.00
		30	**41.82**	60.00	100.00
		50	**52.73**	80.00	100.00
		100	**63.64**	98.18	100.00
	ε-opt. (ε=0.2)	1	**11.82**	**11.82**	**11.82**
		5	**22.73**	32.73	34.55
		20	**70.00**	72.73	100.00
UnlockPickup ($\|S\| = 17164$)	uniform	1	**0.70**	**0.70**	**0.70**
		2	**0.71**	1.94	3.26
		3	**0.89**	2.63	7.80
		5	**1.78**	6.14	17.41
		10	**5.94**	16.72	29.54
		20	**11.80**	29.03	49.53
		30	**19.40**	36.76	51.50
		50	**26.71**	49.09	51.50
		100	**41.09**	51.50	51.50
	ε-opt. (ε=0.2)	1	**0.70**	**0.70**	**0.70**
		5	**0.72**	**0.72**	20.04
		20	**0.72**	25.57	49.95

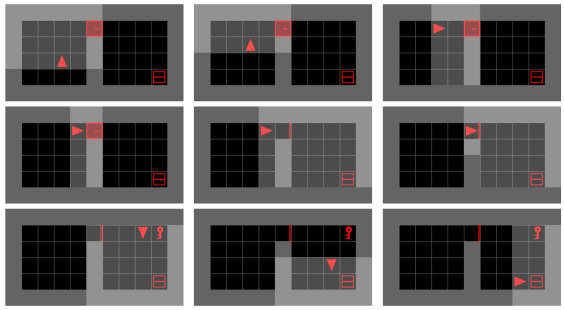
Table 1: Percentage of critical states per decay function and horizon K, evaluated on both tabular environments and two different fallback policies.



(a) CODA with $\beta \to \infty$ and random fallback vs. $\mathsf{SI} > 0.01$. vs. Lazy-MDP with $\eta = 1.37$ and random default policy



(b) Critical states identified by Lazy-MDP with four different default policies and $\eta = 0.07$.



(c) Every state in this figure was identified as critical by CODA-exp with $\beta = 0.5$ and a uniform fallback, and no other states which the optimal policy passes is critical.

Figure 7

exponential decay features too sharp a drop-off. For the agent, it is worth more to wait for ρ to grab the key at random and *only then* use up its intervention power. However, it should be noted that also neither SI nor Lazy-MDP deemed this state critical. When β is lowered, the state before picking up the key does become classified as critical. For this environment, we also find that the critical state sets at different levels exhibit an inclusion chain of the form $C_{k+1} \subseteq C_k$, which is not the case in general.

For $\beta \to \infty$, 120 states were identified as critical, but in all of them, the agent is positioned next to the goal state. The large number simply arises due to the different number of environment configurations (direction and key position). This reflects the discussion from Section 4.1 as an example of an environment where a too large sensitivity β with a random fallback fails to capture any nontrivial critical states.

D.2.1 Comparison to other methods

Because Lazy-MDPs also operate using a fallback/default policy, we can try the four fallback policies from Figure 2. The results are shown in Figure 7b. We used a reward penalty of $\eta = 0.07$ in each case. Interestingly, the result with the "noisy optimal" default policy resembles the results of SI and CODA with the random policy from Figure 7a, whereas CODA only identified states on the right-hand side as critical with this particular fallback. With the reward-minimizing default policy, the result is less symmetrical than ours. Finally, with the *careful* default policy, similar characteristics emerge, such as the pyramid-like shape on the left side.

D.3 Similarity to Baselines

Table 2 shows the confusion matrices between CODA-exp and each of the two baselines on Unlock-Pickup with a uniform fallback.

Table 2: Comparison of critical states identified by CODA vs. SI and Lazy-MDP. **Overlap** denotes the count of states deemed critical by both methods and **Jaccard (IoU)** indicates overall similarity (1.0 = identical).

		State Classification Counts				
Baseline	**Density**	**Overlap (Both)**	**Ours Only**	**Base. Only**	**Neither (TN)**	**Jaccard (IoU)**
SI	Low (5%)	354	509	510	15911	0.26
	Med (10%)	934	769	794	14787	0.37
	High (20%)	1771	1636	1685	12192	0.35
Lazy-MDP	Low (5%)	702	161	170	16251	0.68
	Med (10%)	1413	290	303	15278	0.70
	High (20%)	3142	265	304	13573	0.85

The chosen parameters were as follows:

	q	β	η
Low (5%)	0.05	0.48	0.115
Medium (10%)	0.1	0.27	0.087
High (20%)	0.2	0.15	0.056

D.4 Validation of Deep CODA

Table 3 shows the results of the validation on the CVI ground truth.

k	F1 ↑	Error rate ↓
0	0.968 ± 0.007	0.018 ± 0.004
1	0.916 ± 0.025	0.052 ± 0.017
2	0.929 ± 0.008	0.043 ± 0.005
3	0.938 ± 0.002	0.035 ± 0.000
4	0.990 ± 0.005	0.005 ± 0.002
All	0.946 ± 0.004	0.031 ± 0.003

Table 3: DeepCODA validation: Final-iteration metrics, mean ± std over 3 seeds.

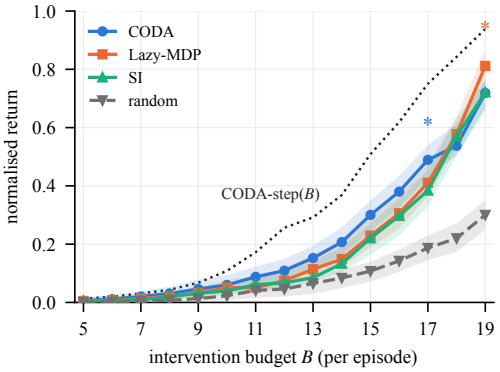


Figure 8: Intervention budget vs. normalised return on UnlockPickup. Stars mark a significant win over the runner-up.

	CODA	Lazy-MDP	SI	random	CODA-step(B)
Normalised return at budget B					
$B = 13$	**0.15**	0.11	0.09	0.07	0.29
$B = 14$	**0.21**	0.15	0.13	0.08	0.37
$B = 15$	**0.30**	0.23	0.22	0.11	0.51
$B = 16$	**0.38**	0.31	0.30	0.14	0.62
$B = 17$	**0.49***	0.41	0.38	0.19	0.75
$B = 18$	0.54	**0.58**	0.57	0.22	0.84
$B = 19$	0.72	**0.81***	0.72	0.30	0.94
Interventions / episode to reach target, $B = \infty$					
30%	**14.6***	15.4	16.2	128.5	–
40%	**15.5***	16.2	16.6	128.5	–
50%	**16.4***	17.2	17.5	128.5	–

Figure 9: Budgeted interventions in UnlockPickup (uniform ρ; 200 episodes, cross-fitted best configuration per method). Bold: best method; *: significantly better than the runner-up (paired 95% bootstrap CI). CODA-step(B) is a reference.

D.5 Budgeted Intervention

D.5.1 Tabular

To get a dense parameter sweep, we started out with an initial parameter grid and progressively refined it via bisection: if two neighboring parameter values produced a gap of more than 1 intervention per episode (on average), we added the midpoint. This was done until the gap in interventions/episode reduced to 1 or lower, or until the gap between parameters was 10^{-4}, in which case we counted it as a genuine discontinuity.

Figure 8 and Table 9 shows the full budgeting results for UnlockPickup. Similarly, Figure 10 and Table 11 show the budgeting results on CliffMaze, which were only briefly mentioned in the text.

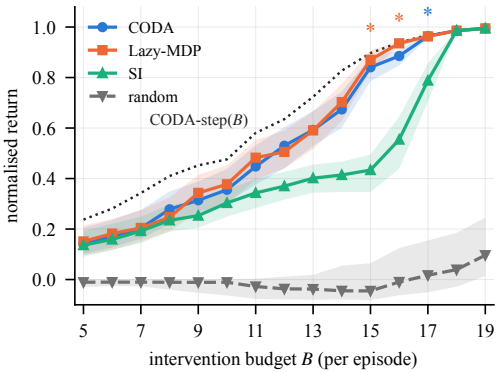


Figure 10: Intervention budget vs. normalised return on CliffMaze. Stars mark a significant win over the runner-up.

	CODA	Lazy-MDP	SI	random	CODA-step(B)
Normalised return at budget B					
$B = 11$	0.45	**0.48**	0.34	−0.03	0.58
$B = 12$	**0.53**	0.51	0.37	−0.04	0.64
$B = 13$	**0.59**	**0.59**	0.40	−0.04	0.72
$B = 14$	0.67	**0.70**	0.41	−0.04	0.83
$B = 15$	0.84	**0.87***	0.43	−0.05	0.90
$B = 16$	0.89	**0.94***	0.55	−0.01	0.94
$B = 17$	**0.96***	**0.96**	0.79	0.02	0.97
Interventions / episode to reach target, $B = \infty$					
70%	**13.9**	**13.9**	17.5	–	–
80%	**13.9**	14.0	17.5	–	–
90%	15.5	**15.0**	17.5	–	–

Figure 11: Budgeted interventions in CliffMaze (uniform ρ; 100 episodes, cross-fitted best configuration per method). Bold: best method; *: significantly better than the runner-up (paired 95% bootstrap CI). CODA-step(B) is a reference, not a contender; –: target not reached.

D.5.2 CONTINUOUS

Our Deep CODA models were trained for 150 iterations, resulting in 1.5M gradient steps. The number of collected episodes was $2000 + 150 \times 200 = 32000$. We optimized all networks with Adam (Kingma & Ba, 2015) with a learning rate of 0.001.

For Hopper and LunarLander, we used the rl-baselines3-zoo (Raffin, 2020) configurations for SAC and PPO.

For Hopper, we averaged across 100 episodes during evaluation, while going up to 500 on LunarLander, as it is more noisy.