MENTOR-BASELINE IMPLICIT IMITATION LEARNING: GUIDING REINFORCEMENT LEARNING WITH A SINGLE ACTION-FREE MENTOR TRAJECTORY

Anonymous authors
Paper under double-blind review

ABSTRACT

In online reinforcement learning (RL), we investigate whether a learning agent (observer) can benefit from experience acquired by another agent (mentor) in addition to its own interaction with the environment. The mentor need not be an expert, the two agents may have different state and action spaces, and the observer is given only a single action-free mentor trajectory of states and rewards. In this setting, we propose Mentor-Baseline Implicit Imitation Learning (MB-I2L), which uses the mentor's return-to-go and approximate mentor–observer state correspondence to form state baselines and propagates them to guide the observer's action-value learning. We further formalize mentor-derived baseline optimism, which can drive exploration but can also bias value learning, and bound the resulting policy performance loss. In sparse-reward environments, we evaluate MB-I2L using a simple ball as the mentor and observers with substantially different embodiment and control requirements, including a high-degree-of-freedom Ant for maze navigation and a Hopper for obstacle crossing. Our results show that a single action-free mentor trajectory can facilitate learning across substantial embodiment differences, with MB-I2L being the only compared method to learn to reach the goal in AntMaze. They also show that the observer can achieve a higher return than the mentor, while mentor–observer correspondence too coarse to distinguish states of different values can induce overly optimistic baselines and impede learning, consistent with our theoretical analysis.

1 INTRODUCTION

In online reinforcement learning (RL) (Sutton & Barto, 1998), an agent learns behavior by interacting with its environment and collecting its own experience. While this experience forms the basis of learning, experience acquired by other agents can provide an additional source of information and may enable more efficient learning (Hillier et al., 2024). Existing approaches often exploit such experience in the form of high-quality demonstrations collected under conditions tailored to the learner, for example by having an expert control a robot identical to the learner (An et al., 2025). Leveraging useful experience without such carefully tailored demonstrations could substantially broaden the range of information sources available for behavior acquisition.

As an important application, we focus on sparse-reward RL, where such additional information can be particularly valuable. The learner receives little useful learning signal until it discovers states that yield reward, making exploration particularly difficult (Andrychowicz et al., 2017; Ecoffet et al., 2021). Dense rewards can guide exploration, but designing them for complex tasks is difficult, and poorly designed rewards can induce unintended behavior (Hadfield-Menell et al., 2017; Skalse et al., 2022). An alternative is to leverage experience from other agents that have obtained rewards in corresponding tasks, since such experience may contain information about states that lead to reward and about the reward itself.

We therefore consider an online RL setting with an experience-providing agent (mentor) and a learning agent (observer). The mentor provides auxiliary information while the observer maximizes its own task reward; the two agents have corresponding task objectives but may have different state and action spaces. Before learning, the observer is given only a single action-free mentor trajectory of

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.

states and rewards. The mentor need not be optimal; for the sparse-reward application studied here, we use trajectories containing a success reward so that the observed return-to-go carries task-relevant value information.

The existing framework most closely related to this idea is Implicit Imitation Learning (I2L) (Price & Boutilier, 2003). Rather than directly reproducing the mentor's behavior, I2L uses information obtained from mentor observations to assist the observer's own reinforcement learning. However, conventional I2L assumes one-to-one mentor–observer state correspondence, which prevents its direct use across different state spaces, and relies on estimating the mentor's transition model.

To address this setting, we propose Mentor-Baseline Implicit Imitation Learning (MB-I2L), which uses the mentor's return-to-go and approximate mentor–observer correspondence to form state baselines and propagates them to guide the observer's action-value learning (Figure 1). These baselines may be optimistic: such optimism can drive exploration toward rewarding states, but it can also bias the learned values, so we analyze how it affects the resulting policy. Our contributions are as follows.

- We formulate an implicit imitation setting in which the observer can leverage a single action-free mentor trajectory even when the mentor and observer have different state and action spaces. We further formalize mentor-derived optimism through a state value gap and derive a bound on policy performance degradation in terms of the corresponding propagated value gap.
- Through evaluations on PointMaze and AntMaze from Gymnasium-Robotics (de Lazcano et al., 2024) and on an environment extending Hopper from Gymnasium (Towers et al., 2024), we show that even a single episode of data from a simple mentor, such as a ball, can facilitate sparse-reward learning across substantial embodiment differences; in AntMaze, MB-I2L is the only compared method that learns to reach the goal. We further show that the observer can learn behavior with a higher return than the mentor, while mentor–observer correspondence too coarse to distinguish states of different values can induce overly optimistic baselines and impede learning, consistent with our theoretical analysis.

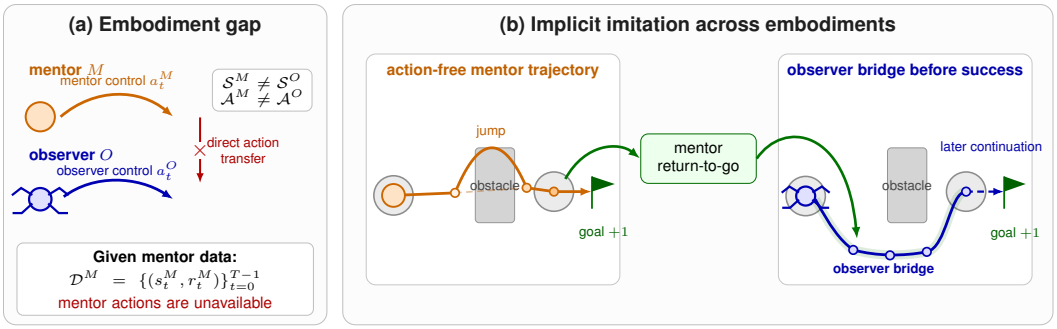


Figure 1: Illustration of implicit imitation across embodiments with MB-I2L. (a) The mentor and observer have different state and action spaces, and the mentor data contain no actions. (b) When mentor and observer trajectory segments connect corresponding neighborhoods, MB-I2L treats them as approximately corresponding and propagates mentor-derived baselines along the observer segment to guide action-value learning.

2 Preliminaries

2.1 Mentor and Observer Processes

We model the mentor and observer environments as discounted Markov decision processes (MDPs) $M^i = (\mathcal{S}^i, \mathcal{A}^i, R^i, P^i, \gamma)$, $i \in \{M, O\}$, where $\mathcal{S}^i$ and $\mathcal{A}^i$ are the state and action spaces, R^i is the reward function defined below, $P^i(\cdot \mid s, a)$ is the transition kernel, that is, the distribution of the next state given the state-action pair (s, a), and $\gamma \in (0, 1)$ is a discount factor shared by the two

MDPs. Let S_t^i, A_t^i, and R_t^i denote the random variables representing the state, action, and reward, respectively, at time t, and let s_t^i, a_t^i, and r_t^i denote their realizations. We define the reward function $R^i : \mathcal{S}^i \times \mathcal{A}^i \to \mathbb{R}$ of MDP M^i as $R^i(s,a) := \mathbb{E}[\mathsf{R}_t^i \mid S_t^i = s,\ A_t^i = a]$. We assume that rewards are bounded: there exists $R_{\max} < \infty$ such that $|\mathsf{R}_t^i| \le R_{\max}$ almost surely for every $i \in \{M, O\}$ and t. In the discounted infinite-horizon setting considered in the paper, policies are assumed to be stationary Markov policies unless otherwise stated. For an observer policy π^O, we define the action-value and state-value functions as

$$Q^{O,\pi^O}(s,a) := \mathbb{E}_{\pi^O}\left[\sum_{t=0}^{\infty} \gamma^t \mathsf{R}_t^O \mid S_0^O = s,\ A_0^O = a\right]$$

and $V^{O,\pi^O}(s) := \mathbb{E}_{\pi^O}[Q^{O,\pi^O}(s, A_0^O) \mid S_0^O = s]$, respectively. We write the optimal value functions as $V^{O,\star}(s) := \sup_{\pi^O} V^{O,\pi^O}(s)$ and $Q^{O,\star}(s,a) := \sup_{\pi^O} Q^{O,\pi^O}(s,a)$. The mentor and observer share a common task objective in the sense that their rewards have corresponding meanings, such as reaching a goal or achieving task success, but their state and action spaces need not be the same. Before the observer begins learning, it is given a realization of an action-free mentor trajectory, $\mathcal{D}^M = \{(s_t^M, r_t^M)\}_{t=0}^{T-1}$, and then interacts online with its own environment M^O. In what follows, we fix the observed $\mathcal{D}^M$ and condition all probabilities and expectations on this realization, while omitting this conditioning from the notation. For simplicity, we consider single-episode mentor data throughout, as in our experiments.

2.2 Neighborhoods and Bridges

In MB-I2L, the mentor data are maintained with indices. The index indicates how many steps after the beginning of the episode a state occurs, with the initial state assigned index 0, and is used to associate each state with its return-to-go. Let $\mathcal{S}_{\text{demo}}^M := \{(\kappa, s_\kappa^M)\}_\kappa$ denote the set of indexed states in the mentor data. For notational simplicity, we henceforth write an indexed element (κ, s_κ^M) simply as s_κ^M. We assume that the feature maps $\psi_{\text{raw}}^M : \mathcal{S}_{\text{demo}}^M \to \mathcal{Z}$ and $\psi_{\text{raw}}^O : \mathcal{S}^O \to \mathcal{Z}$, together with a distance function $\text{dist}_{\mathcal{Z}}$, are fixed in advance. Using these, we define $\rho(s_\kappa^M, s^O) := \text{dist}_{\mathcal{Z}}\big(\psi_{\text{raw}}^M(s_\kappa^M), \psi_{\text{raw}}^O(s^O)\big)$. In this work, ψ_{raw}^M depends only on the state in the mentor MDP; the index κ itself is not used as a feature. Typical choices of the feature maps and distance are coordinates and Euclidean distance, respectively. Given a predefined neighborhood radius $\eta_{\text{nb}} > 0$, we write the neighborhood relation between an observer state and a mentor state as $s^O \sim s_\kappa^M \iff \rho(s_\kappa^M, s^O) \le \eta_{\text{nb}}$. We further define the set of observer states to which a baseline may be applied as $\mathcal{S}_{\text{base}}^O := \{s^O \in \mathcal{S}^O \mid \exists s_\kappa^M \in \mathcal{S}_{\text{demo}}^M :\ s^O \sim s_\kappa^M\}$. For each observer state $s^O \in \mathcal{S}_{\text{base}}^O$, there may be more than one neighboring mentor state. We therefore specify in advance a unique assignment rule $\xi : \mathcal{S}_{\text{base}}^O \to \mathcal{S}_{\text{demo}}^M$ that selects one of these mentor states. For the theoretical analysis, we allow any measurable assignment rule satisfying $s^O \sim \xi(s^O)$. In the implementation, we use nearest-neighbor assignment as a specific choice: $\xi(s^O) \in \arg\min_{s_{\kappa'}^M \in \mathcal{S}_{\text{demo}}^M :\, s^O \sim s_{\kappa'}^M} \rho(s_{\kappa'}^M, s^O)$. If multiple candidates are at the same distance, a predefined tie-breaking rule is used to make the assignment unique. For simplicity, we introduce a terminal state $s_\perp$ that is shared by the mentor and observer and is not a neighbor of any other state. That is, the terminal state is a neighbor only of itself, and we set $\xi(s_\perp) := s_\perp$. For each indexed mentor state $s_\kappa^M \in \mathcal{S}_{\text{demo}}^M$, we define the set of observer states corresponding to later mentor states as

$$B^O(s_\kappa^M) := \big\{\tilde{s}^O \in \mathcal{S}_{\text{base}}^O \mid \exists\, \kappa' > \kappa :\ \xi(\tilde{s}^O) = s_{\kappa'}^M\big\} \cup \{s_\perp\}. \tag{1}$$

The condition $\kappa' > \kappa$ enforces forward progress along the mentor trajectory.

In this paper, a *bridge*, loosely speaking, is a pair of mentor–observer state sequences connecting one neighborhood to another. More precisely, suppose that $\xi(s_t^O) = s_\kappa^M$. A bridge is the pair of mentor–observer state sequences from this point until the observer trajectory first reaches $B^O(s_\kappa^M)$. If the endpoint observer state s_{t+j}^O is nonterminal, then $\xi(s_{t+j}^O) = s_{\kappa+k}^M$ for some $k > 0$, and the bridge can be written as $\big((s_\kappa^M, \dots, s_{\kappa+k}^M), (s_t^O, \dots, s_{t+j}^O)\big)$. We call the pairs (s_κ^M, s_t^O) and $(s_{\kappa+k}^M, s_{t+j}^O)$ the bridge start point and bridge endpoint, respectively. If $s_{t+j}^O = s_\perp$, the shared terminal pair $(s_\perp, s_\perp)$ is taken as the bridge endpoint. Hereafter, we refer to the mentor state and observer state in the bridge start point as the start point mentor state and start point observer state, respectively. Likewise, we refer to those in the bridge endpoint as the endpoint mentor state and endpoint observer

state. A bridge that reaches its endpoint in finite time is called a completed bridge. If the bridge endpoint is nonterminal, both the mentor-side index $\kappa \to \kappa+k$ and the observer-side index $t \to t+j$ advance, but the numbers of steps k and j need not be equal. If $s_0^O \notin \mathcal{S}^O_{\mathrm{base}}$, no bridge start point exists until the observer first reaches either $\mathcal{S}^O_{\mathrm{base}}$ or $s_\perp$; if it first reaches a nonterminal state in $\mathcal{S}^O_{\mathrm{base}}$, its assigned mentor state and that state form the bridge start point.

3 Mentor-Baseline Implicit Imitation Learning

In this section, we formulate the construction of the MB-I2L baselines and how they are injected into value updates. We then define a value gap between the mentor-derived baseline and the observer's optimal value, and analyze the relationship between baseline optimism and policy performance after baseline injection. Finally, we describe the specific update procedure for incorporating MB-I2L into soft actor-critic (SAC).

3.1 Mentor-Derived Baseline Construction and Injection

State and action baselines. For an observer state $s \in \mathcal{S}^O_{\mathrm{base}}$, let $\xi(s) = s^M_\kappa$. We define the state baseline as

$$b_{\mathrm{V}}(s) := \sum_{\ell=0}^{T-\kappa-1} \gamma^\ell r^M_{\kappa+\ell}, \qquad b_{\mathrm{V}}(s_\perp) := 0. \tag{2}$$

That is, $b_{\mathrm{V}}(s)$ is the empirical return-to-go from the mentor state assigned to s. We propagate b_{V} in Eq. (2) through a bridge and the behavior policy π^O_{b} to construct an action baseline. In what follows, we assume that the behavior policy π^O_{b} is fixed. We also use (S^O_u, A^O_u) to denote the pair of random variables corresponding to the observer state and action at time u. For each start point mentor state $m \in \mathcal{S}^M_{\mathrm{demo}}$ of a bridge, we define the action baseline $b^m_{\mathrm{Q}} : \mathcal{S}^O \times \mathcal{A}^O \to \mathbb{R}$ as follows. Along an observer trajectory, define the number of steps from time u until the bridge endpoint set $B^O(m)$ is first reached as $\tau^m(u) := \inf\{n \geq 1 \mid S^O_{u+n} \in B^O(m)\}$. The distribution of $\tau^m(u)$ is determined by the policy under which the following expectation is taken. Under the behavior policy π^O_{b}, we define

$$b^m_{\mathrm{Q}}(s,a) := \mathbb{E}_{\pi^O_{\mathrm{b}}}\left[\sum_{n=0}^{\tau^m(u)-1} \gamma^n \mathsf{R}^O_{u+n} + \mathbf{1}\{\tau^m(u) < \infty\}\gamma^{\tau^m(u)} b_{\mathrm{V}}(S^O_{u+\tau^m(u)}) \mid S^O_u = s,\ A^O_u = a\right]. \tag{3}$$

Baseline injection. Because the same observer state s may appear within bridges with different start point mentor states m, baseline injection marginalizes the effect of the start point mentor state m using $\nu(\cdot \mid s)$. Here, $\nu(\cdot \mid s)$ is a predefined sub-probability kernel on $\mathcal{S}^M_{\mathrm{demo}}$ satisfying $\sum_{m\in\mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s) \leq 1$. For the residual mass $1 - \sum_{m\in\mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)$, no baseline injection is performed, and the standard Bellman backup is used. Let $\mathcal{T}^\star$ denote the Bellman optimality operator for a bounded state-action function $Q : \mathcal{S}^O \times \mathcal{A}^O \to \mathbb{R}$ on the observer MDP. We then define the baseline-injection operator as

$$(\mathcal{T}^{\mathrm{BI}}_\nu Q)(s,a) := (\mathcal{T}^\star Q)(s,a) + \sum_{m\in\mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)\big(b^m_{\mathrm{Q}}(s,a) - (\mathcal{T}^\star Q)(s,a)\big)_+, \tag{4}$$

where $(\cdot)_+ := \max(\cdot, 0)$. Hereafter, we refer to the second term in Eq. (4) as the baseline-induced increment. As is clear from the definition, if $\nu(m \mid s) = 0$ for all $m \in \mathcal{S}^M_{\mathrm{demo}}$, the operator at (s,a) coincides with the Bellman optimality operator.

3.2 Theoretical Properties of Baseline Injection

Here, we define the discrepancy between the state baseline b_{V} and the observer's optimal value as the state value gap, and show how this gap affects policy performance after baseline injection through bridges. We then interpret the value gap in a goal-reward setting.

State value gap, propagated value gap, and performance loss. We define, respectively, the state value gap of the state baseline b_{V} relative to the observer's optimal value, the propagated value gap obtained by propagating this gap under the same fixed behavior policy π_{b}^{O}, and the positive propagated value gap obtained by marginalizing the positive part for each m with respect to ν as follows.

$$\Gamma^{\star}(s) := b_{\mathrm{V}}(s) - V^{O,\star}(s), \tag{5}$$

$$\Gamma_{\mathrm{Q}}^{m,\star}(s,a) := \mathbb{E}_{\pi_{\mathrm{b}}^{O}}\left[\mathbf{1}\{\tau^{m}(u) < \infty\}\gamma^{\tau^{m}(u)}\Gamma^{\star}(S^{O}_{u+\tau^{m}(u)}) \mid S_u^O = s,\ A_u^O = a\right], \tag{6}$$

$$\Gamma_{\mathrm{Q},+}^{\nu,\star}(s,a) := \sum_{m\in\mathcal{S}^{M}_{\mathrm{demo}}} \nu(m \mid s)\big(\Gamma_{\mathrm{Q}}^{m,\star}(s,a)\big)_{+}. \tag{7}$$

Here, we set $\Gamma^{\star}(s_{\perp}) := 0$. In what follows, we assume that a measurable optimal policy exists in the observer MDP. Let $d^{\pi}_{s_0}$ denote the normalized discounted occupancy measure under policy π from initial state s_0. We further assume that there exists a measurable greedy policy $\pi_{\nu}^{\dagger}$ with respect to Q_{ν}^{BI}. We measure the effect of baseline injection by the *performance loss* $V^{O,\star}(s_0) - V^{O,\pi_{\nu}^{\dagger}}(s_0)$ of $\pi_{\nu}^{\dagger}$ from an initial state s_0, that is, its suboptimality in the original observer MDP, and establish a bound on it.

Theorem 1 (Performance loss under bridge-conditioned optimistic baselines)**.** *For a predefined sub-probability kernel ν, $\mathcal{T}_{\nu}^{\mathrm{BI}}$ has a unique fixed point Q_{ν}^{BI}. Then, for any initial state s_0,*

$$V^{O,\star}(s_0) - V^{O,\pi_{\nu}^{\dagger}}(s_0) \le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_{\nu}^{\dagger}}_{s_0}}\left[\mathbb{E}_{a\sim\pi_{\nu}^{\dagger}(\cdot|s)}\left[\Gamma_{\mathrm{Q},+}^{\nu,\star}(s,a)\right]\right]. \tag{8}$$

Theorem 1 shows that the performance loss after baseline injection is bounded by the discounted cumulative amount of $\Gamma_{\mathrm{Q},+}^{\nu,\star}$. This bound is conservative because it also counts increases that do not change the action ordering, but we present it in the main text because of its direct connection to the optimism of b_{V}. A tighter oracle-based refinement is provided in Appendix C.2. Moreover, when the baseline does not exceed the observer's optimal value, optimality is preserved as follows.

Corollary 1 (Optimality preservation under nonoptimistic baselines)**.** *If $\Gamma^{\star}(s) \le 0$ holds for every $s \in \mathcal{S}^{O}_{\mathrm{base}} \cup \{s_{\perp}\}$, then $Q_{\nu}^{\mathrm{BI}} = Q^{O,\star}$. Therefore, any greedy policy with respect to Q_{ν}^{BI} is also optimal in the original observer MDP.*

Interpretations of the state value gap. We clarify the meaning of $\Gamma^{\star}$ through a concrete example. Here, we consider a setting with deterministic state transitions in which a reward of 1 is received once upon reaching the goal and all other rewards are 0. Let k_M denote the number of remaining actions from the mentor state $\xi(s)$ until the goal reward is received, and let k_O denote the minimum number of actions required to reach the goal from the observer state s. Then, $\Gamma^{\star}(s) = \gamma^{k_M-1} - \gamma^{k_O-1}$. Therefore, if $k_M < k_O$, the mentor-derived baseline overestimates the observer's optimal value, and a positive value gap can arise even between neighboring states because of differences in dynamics or embodiment. Such optimism is not necessarily detrimental during learning: in PointMaze, faster mentors, including ones faster than the observer can move, yielded larger baseline-induced increments and generally better performance (Appendix **??**), whereas Theorem 1 bounds the loss it can leave in the converged policy. A more conservative bound based on an arbitrary reference policy is given in Appendix C.3.

3.3 MB-I2L Implementation with Soft Actor-Critic

Rollout and bridge tracking. In this work, we incorporate MB-I2L into SAC (Haarnoja et al., 2018), a representative reinforcement learning algorithm. During rollout, as in standard SAC, observer transitions $(s_t^O, a_t^O, r_t^O, s_{t+1}^O)$ are stored in the replay buffer. In addition, MB-I2L maintains bridge information and associates it with the corresponding transitions. Specifically, when the observer reaches the neighborhood of a mentor state (denoted by m), we retain this mentor state. When the observer state first reaches $B^O(m)$, the bridge is completed, and its start point and endpoint are associated with each transition from the bridge start point up to immediately before the endpoint in a form that can be referenced during replay.

Sampled soft action baseline and critic-target update. The SAC critic estimates an action value for the entropy-regularized return, whereas b_{V}, which is based on the mentor return-to-go, and the propagated action baseline b_{Q}^m are constructed using only environmental rewards, resulting in a mismatch between these quantities. To address this mismatch, for the soft action baseline corresponding to SAC, we add to b_{Q}^m the observer's action-level discounted entropy-to-go, that is, the expected discounted sum of future entropy-regularization terms, which is represented in the implementation by an auxiliary predictor E_θ. The predictor E_θ is the average of two entropy-to-go heads attached to the two SAC critics and is learned jointly with the critics under the current actor using a Bellman-style temporal-difference (TD) update, similarly to a standard critic (Appendix E.1). For a completed bridge with a nonterminal endpoint, start point mentor state $m = s_\kappa^M$, and observer sequence $(s_t^O, \ldots, s_{t+j}^O)$, we construct the sampled action baseline for each $\ell = 0, \ldots, j-1$ as

$$\widehat{b}_{\mathrm{Q}}^m(s_{t+\ell}^O, a_{t+\ell}^O) := \sum_{n=0}^{j-\ell-1} \gamma^n r_{t+\ell+n}^O + \gamma^{j-\ell} b_{\mathrm{V}}(s_{t+j}^O). \tag{9}$$

This is a sample-based estimate of b_{Q}^m. We then construct the sampled soft action baseline to be compared with the critic target as

$$\widehat{b}_{\mathrm{Q},\alpha}^m(s,a) := \widehat{b}_{\mathrm{Q}}^m(s,a) + E_\theta(s,a). \tag{10}$$

A bridge whose endpoint observer state is $s_\perp$ terminates at the shared terminal state rather than at a later mentor state, so its sampled action baseline would consist only of the observer's own rewards and carry no mentor-derived information because $b_{\mathrm{V}}(s_\perp) = 0$; such a bridge is therefore excluded from baseline injection in the implementation. Except for such bridges, each observer state-action pair from the bridge start point up to immediately before the endpoint is subject to baseline injection for that bridge. Let the standard SAC critic target be y_t^{RL}. For a state-action pair (s_t^O, a_t^O) subject to baseline injection, let m denote the start point mentor state of the corresponding bridge. We define the baseline-induced increment and the MB-I2L critic target y_t^{MB} as

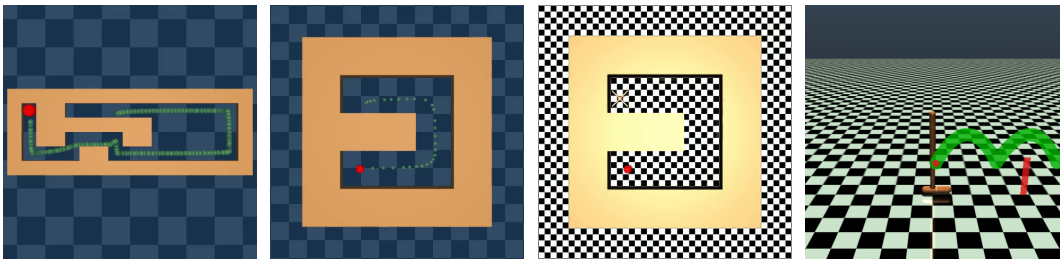
$$y_t^{\mathrm{MB}} := y_t^{\mathrm{RL}} + L_t, \qquad L_t := \min\left\{ \left[\widehat{b}_{\mathrm{Q},\alpha}^m(s_t^O, a_t^O) - y_t^{\mathrm{RL}}\right]_+, c_{\mathrm{inc}} \right\}. \tag{11}$$

Here, $c_{\mathrm{inc}} > 0$ is the clipping threshold for the baseline-induced increment in a single critic update. For transitions that are not subject to baseline injection, we set $L_t = 0$ and $y_t^{\mathrm{MB}} = y_t^{\mathrm{RL}}$. In the experiments, we set $c_{\mathrm{inc}} = 0.01$. An analogous performance bound to Theorem 1 also holds for the entropy-regularized objective with a fixed entropy coefficient α (Appendix D).

4 Related Work

Implicit imitation and demonstration-guided reinforcement learning. Our work is most closely related to Implicit Imitation Learning (I2L) (Price & Boutilier, 2003; Chrysomallis et al., 2025). Rather than directly reproducing the mentor policy, I2L uses information obtained from mentor observations to facilitate the observer's own reinforcement learning, allowing the mentor and observer to have different action spaces and the mentor to be suboptimal. The original I2L framework exploits observed mentor transitions through an estimated mentor transition model, whereas recent deep variants provide model-free mechanisms for using mentor observations. These approaches nevertheless assume a shared mentor–observer state space, whereas MB-I2L uses approximate correspondence across different state spaces and mentor return-to-go to guide the observer's value learning. Although they consider settings different from ours, LOGO (Rengarajan et al., 2022) is also related in that it uses demonstrations to guide online RL rather than directly imitating them. Our method is further related to Monte Carlo Augmented Actor-Critic (MCAC) (Wilcox et al., 2022) and self-imitation learning (Oh et al., 2018), as it uses return-to-go information in critic updates.

Action-free and cross-embodiment learning. A substantial body of work has studied imitation from observation using action-free demonstrations (Stadie et al., 2017; Torabi et al., 2018a;b; Ma et al., 2022), as well as methods that leverage experience across different domains or embodiments (Zakka et al., 2022; Fickinger et al., 2022; Fu et al., 2024; Huey et al., 2025). In particular, Goal-Contrastive Rewards (GCR) (Biza et al., 2025) is closely related to our work in that it uses



(a) PointMaze (b) AntMaze (mentor) (c) AntMaze (observer) (d) HopperHurdle-Long

Figure 2: Environments and mentor trajectories used in the experiments.

action-free passive videos to facilitate online RL under sparse rewards and also addresses cross-embodiment transfer. However, whereas GCR learns a transferable reward representation from multiple videos, including videos from the target embodiment, MB-I2L does not assume access to such diverse demonstrations or a learned reward model. Human2Sim2Robot (Lum et al., 2025), which constructs a reward for cross-embodiment RL from a single human demonstration, and a variant of Primal Wasserstein Imitation Learning (PWIL) (Dadashi et al., 2021) and ORCA (Huey et al., 2025), which use imitation rewards based on state-only trajectories, are also related.

Cross-MDP correspondence and value mismatch. From the perspective of transferring knowledge across different MDPs, our work is related to transfer learning (Taylor & Stone, 2009; Ammar et al., 2014; 2015) and MDP homomorphisms (Givan et al., 2003; Ravindran & Barto, 2003; Soni & Singh, 2006; Taylor et al., 2008). In addition, because our method addresses value differences between states, it is related to bisimulation metrics (Ferns et al., 2004) and their extensions (Gelada et al., 2019; Zhang et al., 2021; Castro et al., 2021; Tao et al., 2025).

5 Experiments

We evaluate MB-I2L in three environments: one in which the mentor and observer have the same state space, and two in which their state spaces differ. Through these experiments, we answer the following questions:

- **Q1.** Can a single action-free mentor trajectory aid sparse-reward learning across embodiments?
- **Q2.** Can the observer achieve a higher return than the mentor?
- **Q3.** When can mentor-derived optimism become detrimental?

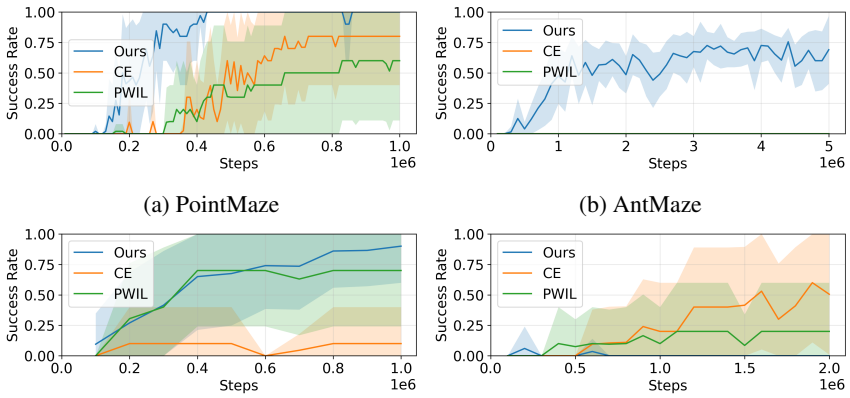
5.1 Experimental Setup

Environments. We conducted experiments on PointMaze and AntMaze from Gymnasium-Robotics, as well as on HopperHurdle, which extends the Hopper environment from Gymnasium by adding a hurdle and a goal (Figure 2). In each environment, the mentor data consisted of a single trajectory from a simple ball mentor, and reward was provided only upon reaching the goal, which also terminated the episode. Because PointMaze has the simplest dynamics, we used it for most ablations and detailed analyses. In Figure 2, the PointMaze panel shows the mentor trajectory overlaid on the observer environment, the AntMaze panels show the mentor trajectory and observer environment separately, and the green region in HopperHurdle-Long indicates neighborhoods around the mentor trajectory. AntMaze additionally terminates when the Ant falls over, while HopperHurdle terminates when the Hopper enters the hurdle or becomes unhealthy under the original Gymnasium termination condition; all environments also terminate at their maximum episode length. For HopperHurdle, we considered Short and Long variants with different goal distances. Mentor–observer neighborhoods used center-of-mass xy coordinates in PointMaze and AntMaze, and center-of-mass xz coordinates together with the sign of the vertical velocity in HopperHurdle; the neighborhood radius was $\eta_{\text{nb}} = 1$ for AntMaze and $\eta_{\text{nb}} = 0.1$ for PointMaze and HopperHurdle. The AntMaze and HopperHurdle environments probe how far the shared features determine

Table 1: Normalized AUC of evaluation success rate (mean ± SD; 10 seeds for Ours, CE, CE', and PWIL and 5 seeds otherwise; bold: best mean). Point, Ant, Hopper-S, and Hopper-L denote PointMaze, AntMaze, HopperHurdle-Short, and HopperHurdle-Long.

	Ours	SAC	CE	CE'
Point	$\mathbf{0.77 \pm 0.076}$	0.00 ± 0.00	0.40 ± 0.022	0.22 ± 0.28
Ant	$\mathbf{0.53 \pm 0.054}$	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Hopper-S	$\mathbf{0.63 \pm 0.27}$	0.00 ± 0.00	0.077 ± 0.23	0.039 ± 0.12
Hopper-L	0.0050 ± 0.016	0.00 ± 0.00	$\mathbf{0.25 \pm 0.28}$	0.046 ± 0.096

	PBRS	HER	PWIL	ORCA	Ori. Rew.
Point	0.00 ± 0.00	0.00 ± 0.00	0.29 ± 0.29	0.050 ± 0.082	n/a
Ant	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	n/a
Hopper-S	0.00 ± 0.00	0.00 ± 0.00	0.58 ± 0.41	0.00 ± 0.00	0.25 ± 0.30
Hopper-L	0.00 ± 0.00	0.00 ± 0.00	0.13 ± 0.29	0.00 ± 0.00	0.048 ± 0.066



(a) PointMaze (b) AntMaze

(c) HopperHurdle-Short (d) HopperHurdle-Long

Figure 3: Learning curves of evaluation success rate for selected methods.

what matters for the task, but in different ways. In AntMaze, the correspondence is position-based while the observer's locomotion dynamics are high-dimensional, so the question is whether such a coarse correspondence still provides useful guidance. In HopperHurdle, the shared features are not sufficient even locally: two observer states that agree on them can differ in whether the hurdle can be cleared at all, because posture and joint states are not represented.

Comparison methods. In addition to standard SAC, we compared MB-I2L (denoted as Ours in the tables) with count-based exploration using hash-based counts (CE) (Tang et al., 2017), its variant restricting the bonus to mentor neighborhoods (CE'), Hindsight Experience Replay (HER) (Andrychowicz et al., 2017), potential-based reward shaping (PBRS) (Ng et al., 1999), a state-based variant of PWIL (Dadashi et al., 2021), and ORCA (Huey et al., 2025). For HopperHurdle, we additionally included SAC trained with the original dense Hopper reward (Ori. Rew.), which remains meaningful for this task.

5.2 Main Results

We summarize learning performance by the normalized area under the curve (AUC) of evaluation success rate, scaled so that success rates of 0 and 1 throughout training correspond to AUCs of 0 and 1, respectively. Table 1 reports the mean and standard deviation across seeds, and Figure 3 shows learning curves for selected methods. Regarding Q1, on AntMaze, MB-I2L was the only compared method that learned to reach the goal despite using only position-based correspondence and mentor return-to-go from the ball mentor. MB-I2L also performed strongly on PointMaze and HopperHurdle-Short, but largely failed on HopperHurdle-Long. On HopperHurdle-Short, its

Table 2: Normalized AUC of evaluation success rate for different η_{nb} (mean $\pm$ SD over 5 seeds; 10 seeds for $\eta_{\mathrm{nb}} = 0.1$).

	$\eta_{\mathrm{nb}} = 0.01$	$\eta_{\mathrm{nb}} = 0.1$	$\eta_{\mathrm{nb}} = 1$	$\eta_{\mathrm{nb}} = 2$	$\eta_{\mathrm{nb}} = 2'$
Point	0.59 ± 0.33	$\mathbf{0.77 \pm 0.076}$	0.62 ± 0.36	0.00 ± 0.00	0.56 ± 0.23

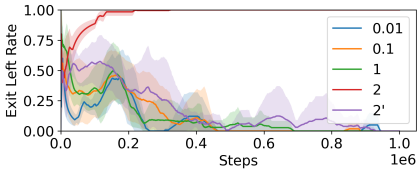


Figure 4: Rate of rollouts that initially moved to the left.

mean normalized AUC also exceeded that of SAC trained with the original dense reward. Together, these results show that partial mentor–observer correspondence can provide useful guidance even across substantial embodiment differences. Regarding Q2, after 1M training steps on PointMaze, the MB-I2L policy reached the goal in 326.12 ± 8.03 steps (mean $\pm$ SD), whereas the mentor required 720 steps, showing that MB-I2L learned behavior with a higher discounted return rather than merely reproducing the mentor trajectory. The failure on HopperHurdle-Long suggests that embodiment-specific information unavailable from the ball mentor matters more when multiple jumps are required; its partial recovery when baseline injection is disabled (Appendix G.8) indicates that optimistic baselines may also contribute.

5.3 Failure Analysis

To address Q3, we use PointMaze to examine how the neighborhood definition affects failures of MB-I2L. Because MB-I2L leverages mentor information through predefined features and their induced neighborhoods, overly optimistic baselines can arise from the neighborhood definition, and our theoretical analysis shows that they can degrade policy performance. We vary the default neighborhood radius ($\eta_{\mathrm{nb}} = 0.1$) over $\eta_{\mathrm{nb}} = 0.01, 0.1, 1, 2$ and additionally consider a wall-aware variant, denoted $\eta_{\mathrm{nb}} = 2'$, which keeps the radius at 2 but excludes mentor–observer pairs separated by a wall. Since the narrowest wall has width 1, $\eta_{\mathrm{nb}} = 2$ can associate states on opposite sides of a wall. We evaluate both the normalized AUC and the rate at which rollouts initially move left toward the incorrect direction. The AUC results (Table 2) show substantial degradation at $\eta_{\mathrm{nb}} = 2$, whereas performance recovers under $\eta_{\mathrm{nb}} = 2'$ despite the same radius. Moreover, only $\eta_{\mathrm{nb}} = 2$ substantially increases the rate of initial movement toward the left dead end (Figure 4), suggesting that inappropriate correspondences can cause baseline injection to guide the observer in the wrong direction. These results show that MB-I2L depends not only on neighborhood radius but also on whether the induced mentor–observer correspondences are meaningful. This dependence is an inherent limitation of our setting: with only a single mentor trajectory and no given mentor–observer correspondences, cross-space correspondence generally cannot be identified uniquely without additional prior information, making some prespecified correspondence structure necessary.

6 Conclusion

In this work, we proposed Mentor-Baseline Implicit Imitation Learning (MB-I2L), a method applicable even when the mentor and observer have different state and action spaces. MB-I2L leverages the mentor return-to-go as a baseline for the observer's critic target. Through theoretical analysis, we bounded the performance loss that mentor-induced optimism can cause, although such optimism can also drive exploration. Our experiments showed that MB-I2L can facilitate the observer's sparse-reward learning from only a single action-free mentor trajectory even when there is a substantial embodiment gap between the mentor and observer. In particular, on AntMaze, MB-I2L was the only compared method that learned to reach the goal from a ball mentor using only position-based mentor–observer correspondence. At the same time, our empirical results highlighted the importance of neighborhood design for the performance of MB-I2L.

AI DISCLOSURE

In this work, we used generative AI tools for the following tasks: help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, implement methods, assist with translation. We have not used generative AI tools for the following tasks: generate synthetic data sets, design or provide feedback on research methodology or experiments, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools for the following tasks: create or modify scientific figures or images, create or edit software code, draft parts of a research paper, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature, propose a title or keywords for a research paper.

We have reviewed all AI-assisted work. All AI-assisted mathematical claims, derivations, and proofs were independently checked by the authors for correctness and consistency with the stated assumptions. AI-generated or AI-edited code was reviewed by the authors and tested to verify that it implemented the intended methods. References and factual information suggested through AI-assisted searching were checked against the original sources before being included in the paper. AI-assisted drafts, translations, edits, and figures were manually reviewed and revised by the authors to ensure technical accuracy and consistency with the intended meaning. Research ideas and hypotheses suggested with AI assistance were critically evaluated by the authors before being incorporated into the work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

CODE OF ETHICS

This work studies implicit imitation learning in simulation and does not involve human participants, personal or sensitive data, or experiments on physical robots. A potential concern specific to implicit imitation is that inaccurate or coarse mentor–observer correspondence may transfer misleading value information and thereby promote undesirable observer behavior, as demonstrated in our failure analysis. The results of this work should therefore not be interpreted as establishing the safety of MB-I2L for physical or safety-critical deployment. Such applications would require additional validation of the mentor–observer correspondence and appropriate task-specific safety mechanisms.

REPRODUCIBILITY

We provide detailed information to support the reproducibility of both our theoretical and empirical results. The assumptions and notation underlying the theoretical analysis are specified in Appendix A, and the proofs of the results stated in the main text are given in Appendix B. Interpretations and refinements of these results are provided in Appendix C, the entropy-regularized extension corresponding to our SAC implementation in Appendix D, and the treatment of the finite-horizon episodic setting used in our experiments in Appendix F. The implementation details of MB-I2L with SAC are described in Section 3.3 and Appendix E. Experimental settings, including the mentor–observer correspondence and neighborhood specification, are described in Section 5, while further details on the environments, mentor-trajectory construction, comparison methods, evaluation protocols, and additional analyses are provided in Appendix G. We additionally provide the source code, configuration files used for the reported experiments, and the mentor trajectory data as supplementary material.

REFERENCES

Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of MDP similarity for transfer in reinforcement learning. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 31–37, 2014.

Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Shan An, Ziyu Meng, Chao Tang, Yuning Zhou, Tengyu Liu, Fangqiang Ding, Shufang Zhang, Yao Mu, Ran Song, Wei Zhang, et al. Dexterous manipulation through imitation learning: A survey. *IEEE Transactions on Automation Science and Engineering*, 23:1760–1792, 2025.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in Neural Information Processing Systems*, 30, 2017.

Ondrej Biza, Thomas Weng, Lingfeng Sun, Karl Schmeckpeper, Tarik Kelestemur, Yecheng Jason Ma, Robert Platt, Jan-Willem van de Meent, and Lawson L. S. Wong. On-robot reinforcement learning with goal-contrastive rewards. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4797–4805, 2025. doi: 10.1109/ICRA55743.2025.11128466.

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. MICo: Improved representations via sampling-based state similarity for Markov decision processes. *Advances in Neural Information Processing Systems*, 34:30113–30126, 2021.

Iason Chrysomallis, Georgios Chalkiadakis, Ioannis Papamichail, and Markos Papageorgiou. Deep implicit imitation reinforcement learning in heterogeneous action settings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16055–16063, 2025.

Robert Dadashi, Leonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal Wasserstein imitation learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=TtYSU29zgR.

Rodrigo de Lazcano, Andreas Kallinteris, Jun Jet Tai, Seungjae Ryan Lee, and Jordan Terry. Gymnasium Robotics, 2024. URL http://github.com/Farama-Foundation/Gymnasium-Robotics.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pp. 162–169. AUAI Press, 2004.

Arnaud Fickinger, Samuel Cohen, Stuart Russell, and Brandon Amos. Cross-domain imitation learning via optimal transport. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xP3cPq2hQC.

Yuwei Fu, Haichao Zhang, Di Wu, Wei Xu, and Benoit Boulet. Robot policy learning with temporal optimal transport reward. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 122078–122103. Curran Associates, Inc., 2024. doi: 10.52202/079017-3879. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/dcd297696d0bb304ba426b3c5a679c37-Paper-Conference.pdf.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pp. 2170–2179. PMLR, 2019.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in Markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

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

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. *Advances in Neural Information Processing Systems*, 30, 2017.

Dylan Hillier, Cheston Tan, and Jing Jiang. Social learning through interactions with other agents: A survey. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pp. 8067–8076. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/892.

William Huey, Huaxiaoyue Wang, Anne Wu, Yoav Artzi, and Sanjiban Choudhury. Imitation learning from a single temporally misaligned video. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26283–26310. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huey25a.html.

Takahisa Imagawa and Shuichi Enokida. Biased exploration q-learning: A simple method for embedding knowledge into reinforcement learning. In *ICAART (2)*, pp. 1242–1253, 2026.

Tyler Ga Wei Lum, Olivia Y. Lee, Karen Liu, and Jeannette Bohg. Crossing the human-robot embodiment gap with sim-to-real RL using one human demonstration. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4418–4441. PMLR, 2025.

Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 278–287. Morgan Kaufmann, 1999.

Junhyuk Oh, Xiaoxiao Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 3878–3887. PMLR, 2018.

Bob Price and Craig Boutilier. Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research*, 19:569–629, 2003.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL https://jmlr.org/papers/v22/20-1364.html.

Balaraman Ravindran and Andrew G Barto. SMDP homomorphisms: an algebraic approach to abstraction in semi-Markov decision processes. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 1011–1016, 2003.

Desik Rengarajan, Gargi Vaidya, Akshay Sarvesh, Dileep Kalathil, and Srinivas Shakkottai. Reinforcement learning with sparse rewards using guidance from offline demonstration. In *International Conference on Learning Representations*, 2022.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Vishal Soni and Satinder Singh. Using homomorphisms to transfer options across continuous reinforcement learning domains. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pp. 494–499, 2006.

Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third person imitation learning. In *International Conference on Learning Representations*, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. #Exploration: A study of count-based exploration for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 30, 2017.

Zhenyu Tao, Wei Xu, and Xiaohu You. A generalized bisimulation metric of state similarity between Markov decision processes: From theoretical propositions to applications. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jonathan Taylor, Doina Precup, and Prakash Panangaden. Bounding performance loss in approximate MDP homomorphisms. *Advances in Neural Information Processing Systems*, 21, 2008.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4950–4957, 2018a.

Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018b.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Hado Van Hasselt. Estimating the maximum expected value: an analysis of (nested) cross validation and the maximum sample average. *arXiv preprint arXiv:1302.7175*, 2013.

Albert Wilcox, Ashwin Balakrishna, Jules Dedieu, Wyame Benslimane, Daniel S. Brown, and Ken Goldberg. Monte Carlo augmented actor-critic for sparse reward deep reinforcement learning from suboptimal demonstrations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=FLzTj4ia8BN.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. XIRL: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pp. 537–546. PMLR, 2022.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

Table 3: Notation used in this paper.

Symbol	Meaning
M, O	Mentor and observer MDP.
$\mathcal{S}^i, \mathcal{A}^i$	State space and action space of MDP M^i.
$S_t^i, A_t^i, \mathsf{R}_t^i$ / s_t^i, a_t^i, r_t^i	Random variables for the state, action, and reward at time t / their realizations.
V, Q	Italic V and Q denote state-value and action-value functions.
$s_\perp$	Absorbing terminal state shared by the mentor and the observer. It is a neighbor only of itself, $\xi(s_\perp) := s_\perp$, $b_{\mathrm{V}}(s_\perp) := 0$, and $s_\perp \in B^O(m)$ for every m.
T	Length of the mentor trajectory.
π_{b}^O	Behavior policy of the observer. It is held fixed in the theory and is used to construct the action baseline and the propagated value gap.
$d_{s_0}^\pi$	Normalized discounted occupancy measure induced by policy π from initial state s_0 (Eq. (14)).
$\mathcal{D}^M$	Realization of the action-free mentor trajectory, given before learning and kept fixed thereafter.
$\mathcal{S}_{\mathrm{demo}}^M$, κ	Set of indexed mentor states in the mentor data, and the index of a mentor state. The initial mentor state has index 0, and the index associates each mentor state with its return-to-go.
ρ, η_{nb}, $s \sim m$	Distance between a mentor state and an observer state in the shared feature space; the neighborhood radius; and the neighborhood relation $\rho(m, s) \leq \eta_{\mathrm{nb}}$ between observer state s and mentor state m.
$\mathcal{S}_{\mathrm{base}}^O$	Set of observer states having at least one neighboring mentor state, that is, the states to which a baseline may be applied.
$\xi(s)$	Unique assignment rule that selects one neighboring mentor state for an observer state $s \in \mathcal{S}_{\mathrm{base}}^O$. Any measurable assignment is allowed in the theory; the implementation uses nearest-neighbor assignment.
$B^O(m)$	Set of observer states that ξ assigns to a mentor state with an index later than that of the current mentor state m, together with $s_\perp$.
$\tau^m(u)$	Number of steps after time u until the observer trajectory first reaches $B^O(m)$. Its distribution is determined by the policy under which the expectation is taken.
b_{V}, b_{Q}^m, $b_{\mathrm{Q},\alpha}^m$	The state baseline in the main text, defined on $\mathcal{S}_{\mathrm{base}}^O \cup \{s_\perp\}$; the action baseline constructed along the fixed behavior policy conditioned on the start point mentor state m; and its fixed-α soft counterpart. Both b_{Q}^m and $b_{\mathrm{Q},\alpha}^m$ are defined on all of $\mathcal{S}^O \times \mathcal{A}^O$.
$\mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O}(s)$, $\mathcal{E}_\alpha^{\pi_{\mathrm{b}}^O}(s, a)$	State-level and action-level discounted entropy-to-go under the fixed behavior policy π_{b}^O. They satisfy $b_{\mathrm{Q},\alpha}^m = b_{\mathrm{Q}}^m + \mathcal{E}_\alpha^{\pi_{\mathrm{b}}^O}$.
$\mathcal{T}^\star$, $\mathcal{T}_\alpha^\star$	Bellman optimality operator on the observer MDP and its fixed-α soft counterpart.
$\nu(m \mid s)$	Weight used at observer state s to marginalize over the start point mentor state $m \in \mathcal{S}_{\mathrm{demo}}^M$ of a bridge. $\nu(\cdot \mid s)$ is a measurable sub-probability kernel on $\mathcal{S}_{\mathrm{demo}}^M$ (see Appendix A.3).
$\mathcal{T}_\nu^{\mathrm{BI}}$, Q_ν^{BI}	Baseline-injection operator (Eq. (4)) and its unique fixed point.
$\pi_\nu^\dagger$	Measurable greedy observer policy that assigns probability 1 to the argmax set of Q_ν^{BI}.
$\Gamma^\star$	State value gap, defined as the difference between the state baseline b_{V} and the observer's optimal value $V^{O,\star}$ (Eq. (5)).
$\Gamma_{\mathrm{Q}}^{m,\star}$	Propagated value gap obtained by propagating $\Gamma^\star$ to the action level along a bridge, conditioned on the start point mentor state m.
$\Gamma_{\mathrm{Q},+}^{\nu,\star}$	Positive propagated value gap, obtained by marginalizing the positive part of the propagated value gap with respect to ν.
ζ_ν	Quantity defined in Eq. (18). It satisfies $Q_\nu^{\mathrm{BI}} = (\mathcal{T}^\star Q_\nu^{\mathrm{BI}}) + \zeta_\nu$.
Z_ν^π	Discounted cumulative amount of ζ_ν along a policy π. Used in Appendix C.2 for the oracle-based refinement that isolates the conservativeness of the bound in the main text.

Table 3 (continued)

Symbol	Meaning
θ_i, θ, $E_{\theta_i}(s,a)$, $E_\theta(s,a)$	Parameters of the i-th SAC critic ($i \in \{1,2\}$), including its shared trunk, its action-value head Q_{θ_i}, and its entropy-to-go head E_{θ_i}; $\theta := (\theta_1, \theta_2)$; and the predictor $E_\theta := \frac{1}{2}(E_{\theta_1} + E_{\theta_2})$ used in the SAC implementation. The heads predict the action-level discounted entropy-to-go under the current actor by a Bellman-style TD update, jointly with the critics. Target parameters are denoted by $\bar{\theta}_i$.
$\widehat{b}^m_{\mathrm{Q}}(s,a)$, $\widehat{b}^m_{\mathrm{Q},\alpha}(s,a)$	Sampled action baseline constructed from the environmental rewards of an observed completed bridge whose endpoint observer state is not $s_\perp$ and from the state baseline b_{V} at the endpoint, and the sampled soft action baseline obtained by adding $E_\theta(s,a)$ to it.
y_t^{RL}, y_t^{MB}, L_t	Standard SAC critic target, the MB-I2L critic target, and the sample-level baseline-induced increment that relates them through $y_t^{\mathrm{MB}} = y_t^{\mathrm{RL}} + L_t$.
c_{inc}	Clipping threshold that limits the size of the baseline-induced increment L_t in a single update. Setting $c_{\mathrm{inc}} = \infty$ recovers the unclipped target.

A Notation and Technical Preliminaries

This section collects the notation used throughout the paper and states the standing assumptions of the theoretical analysis.

A.1 Standing Assumptions

Throughout the following theoretical analysis, the state space $\mathcal{S}^i$ and action space $\mathcal{A}^i$ of each MDP M^i are assumed to be standard Borel spaces equipped with their respective Borel σ-algebras. For a standard Borel space $\mathcal{X}$, let $\mathcal{B}_b(\mathcal{X})$ denote the set of bounded Borel-measurable real-valued functions on $\mathcal{X}$. Equipped with the sup norm

$$\|f\|_\infty := \sup_{x \in \mathcal{X}} |f(x)|,$$

$\mathcal{B}_b(\mathcal{X})$ is a Banach space. We assume that the reward function satisfies $R^i \in \mathcal{B}_b(\mathcal{S}^i \times \mathcal{A}^i)$, and that P^i and all policies used below are measurable stochastic kernels. We further assume that, for each $m \in \mathcal{S}^M_{\mathrm{demo}}$, the neighborhood set

$$\{s \in \mathcal{S}^O \mid s \sim m\}$$

and the bridge endpoint set $B^O(m)$ are Borel measurable, and that the assignment rule ξ is measurable. We also assume the measurability conditions required for the standard Bellman optimality operator $\mathcal{T}^\star$ to be well-defined on $\mathcal{B}_b(\mathcal{S}^O \times \mathcal{A}^O)$. By the assumptions in the main text, rewards are uniformly bounded, with $|\mathsf{R}^O_t| \le R_{\max}$ almost surely. In the setting of the main text, the fixed behavior policy π^O_{b} is a stationary Markov policy, and the transition kernel and reward function of the observer MDP do not depend on time. Hence, given $(S^O_u, A^O_u) = (s,a)$, the conditional distribution of the subsequent observer trajectory, and thus that of the hitting time $\tau^m(u)$ of the fixed set $B^O(m)$, does not depend on u. Consequently, the conditional expectation on the right-hand side of Eq. (3) does not depend on the time index u, and b^m_{Q} can be defined as a state-action function without time as an argument.

A.2 Additional Details on Bridge Construction

Because no finite-horizon assumption is imposed in the main text, the hitting times involved in bridge construction need not be finite. If $s^O_0 \notin \mathcal{S}^O_{\mathrm{base}}$ and the observer does not reach $\mathcal{S}^O_{\mathrm{base}} \cup \{s_\perp\}$ in finite time, no bridge is initiated in finite time. Likewise, after a bridge starts, the observer may fail to reach its endpoint set $B^O(m)$ in finite time; in this case, $\tau^m(u) = \infty$ and the bridge is not completed.

A.3 Construction of the Marginalization Kernel ν

As stated in the main text, the same observer state s may occur within bridges with different start point mentor states, so baseline injection marginalizes over m using $\nu(\cdot \mid s)$. There is freedom

in how ν is constructed, but, as described below, Theorem 1 and Proposition 2 hold for any such construction.

In the theory, $\nu(\cdot \mid s)$ is taken to be a predefined measurable sub-probability kernel on the finite set $\mathcal{S}^M_{\mathrm{demo}}$. The residual mass $1 - \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)$ corresponds to the case in which no start point mentor state exists yet, that is, to occurrences before the observer first reaches $\mathcal{S}^O_{\mathrm{base}}$.

Even when the observer MDP and the behavior policy π^O_{b} are time-homogeneous, the state distribution generated from the initial state distribution is in general not stationary, so different kernels are obtained depending on which time step the occurrence is taken from. As one construction, fix the initial state distribution and π^O_{b}, choose a time step $u_0 \geq 0$, and let $\nu_{u_0}(m \mid s)$ be a regular version of the conditional probability, given $S^O_{u_0} = s$, that the start point mentor state of the bridge containing the occurrence at time u_0 is m; such a version exists because $\mathcal{S}^M_{\mathrm{demo}}$ is finite. As another construction, occurrences across multiple time steps can be aggregated according to a predefined sampling rule, and this sampling rule can be put in correspondence with the transition sampling of a replay-based implementation. In what follows, we do not fix a particular construction and simply write ν for a given measurable sub-probability kernel.

A.4 Operators and Policy Notation

For readability, we first recall the standard Bellman optimality operator. For $Q \in \mathcal{B}_b(\mathcal{S}^O \times \mathcal{A}^O)$, the Bellman optimality operator is given by

$$(\mathcal{T}^\star Q)(s,a) := R^O(s,a) + \gamma \mathbb{E}_{s' \sim P^O(\cdot \mid s,a)} \left[\sup_{a'} Q(s',a') \right]. \tag{12}$$

Using the identity $x + (b-x)_+ = \max\{x, b\}$, the baseline-injection operator introduced in Eq. (4) can also be written in the following more intuitive form:

$$\begin{aligned}(\mathcal{T}^{\mathrm{BI}}_\nu Q)(s,a) = &\left(1 - \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)\right) (\mathcal{T}^\star Q)(s,a) \\ &+ \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s) \max\{(\mathcal{T}^\star Q)(s,a),\, b^m_{\mathrm{Q}}(s,a)\}.\end{aligned} \tag{13}$$

More precisely, the quantity $d^\pi_{s_0}$ used in the main text is the normalized discounted occupancy measure associated with the initial state s_0 and policy π, defined for any Borel set $C \subseteq \mathcal{S}^O$ by

$$d^\pi_{s_0}(C) := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(S^O_t \in C \mid S^O_0 = s_0, \pi). \tag{14}$$

A measurable greedy policy $\pi^\dagger_\nu$ with respect to Q^{BI}_ν is a policy satisfying

$$\pi^\dagger_\nu \left(\arg\max_{a \in \mathcal{A}^O} Q^{\mathrm{BI}}_\nu(s,a) \,\middle|\, s \right) = 1$$

for every $s \in \mathcal{S}^O$. For simplicity, we assume throughout this paper that such a policy exists.

B Proofs

This section provides proofs of the theorem and corollary presented in the main text. As an interpretation of the theoretical results, Appendix C.1 shows that partitioning the return at a bridge endpoint does not change the value of a fixed observer policy.

B.1 Proof of Theorem 1

Before proving Theorem 1, we establish the following lemma. The lemma provides an action-level upper bound on the extent to which $\Gamma^\star$ can induce overestimation in b^m_{Q} through a bridge.

Lemma 1 (Action baseline overestimation bound)**.** *For any start point mentor state m and observer state-action pair (s, a),*

$$b_{\mathrm{Q}}^{m}(s,a) \le Q^{O,\star}(s,a) + \Gamma_{\mathrm{Q}}^{m,\star}(s,a) \tag{15}$$

holds.

Proof of Lemma 1. By definition (5),

$$b_{\mathrm{V}}(s) = V^{O,\star}(s) + \Gamma^{\star}(s).$$

Fix a start point mentor state m and an observer state-action pair (s, a). Substituting the above identity into definition (3) and using definition (6) gives

$$b_{\mathrm{Q}}^{m}(s,a) = \mathbb{E}_{\pi_{\mathrm{b}}^{O}}\left[\sum_{n=0}^{\tau^{m}(u)-1} \gamma^{n}\mathsf{R}_{u+n}^{O} + \mathbf{1}\{\tau^{m}(u) < \infty\}\gamma^{\tau^{m}(u)}V^{O,\star}(S_{u+\tau^{m}(u)}^{O}) \mid S_{u}^{O} = s, A_{u}^{O} = a\right] + \Gamma_{\mathrm{Q}}^{m,\star}(s,a).$$

The first term is the action value obtained by taking action a at time u, following π_{b}^{O} from time $u+1$ until just before the bridge endpoint at time $u + \tau^{m}(u)$, and following an optimal observer policy thereafter. Therefore, this term is at most $Q^{O,\star}(s, a)$. This yields Eq. (15). □

Proof of Theorem 1. We first express $\mathcal{T}_{\nu}^{\mathrm{BI}}$ as the composition of the standard Bellman optimality operator $\mathcal{T}^{\star}$ and a real-valued mapping defined separately for each state-action pair. Using the 1-Lipschitz property of the latter, we establish the contraction property of $\mathcal{T}_{\nu}^{\mathrm{BI}}$ and the uniqueness of its fixed point. We then use the monotonicity of this mapping to show that $Q^{O,\star} \le Q_{\nu}^{\mathrm{BI}}$.

Consider $\mathcal{T}_{\nu}^{\mathrm{BI}}$ on the Banach space $(\mathcal{B}_b(\mathcal{S}^{O} \times \mathcal{A}^{O}), \|\cdot\|_{\infty})$. First, because b_{V} is constructed from the discounted return-to-go of a fixed finite-length mentor trajectory,

$$\|b_{\mathrm{V}}\|_{\infty} \le \frac{R_{\max}}{1-\gamma}.$$

Moreover, along every trajectory in Eq. (3),

$$\left|\sum_{n=0}^{\tau^{m}(u)-1} \gamma^{n}\mathsf{R}_{u+n}^{O} + \mathbf{1}\{\tau^{m}(u) < \infty\}\gamma^{\tau^{m}(u)}b_{\mathrm{V}}(S_{u+\tau^{m}(u)}^{O})\right| \le \frac{R_{\max}}{1-\gamma},$$

and hence $\|b_{\mathrm{Q}}^{m}\|_{\infty} \le R_{\max}/(1-\gamma)$. Furthermore, since $B^{O}(m)$ is Borel measurable and P^{O} and π_{b}^{O} are measurable stochastic kernels, the conditional expectation on the right-hand side of Eq. (3) is a Borel-measurable function of (s, a). Therefore,

$$b_{\mathrm{Q}}^{m} \in \mathcal{B}_b(\mathcal{S}^{O} \times \mathcal{A}^{O})$$

for every m.

For any $Q \in \mathcal{B}_b(\mathcal{S}^{O} \times \mathcal{A}^{O})$, the assumptions imply that $\mathcal{T}^{\star}Q \in \mathcal{B}_b(\mathcal{S}^{O} \times \mathcal{A}^{O})$. In addition, sums, products, and maxima of Borel-measurable functions are Borel measurable. Thus, by Eq. (13),

$$\mathcal{T}_{\nu}^{\mathrm{BI}}Q \in \mathcal{B}_b(\mathcal{S}^{O} \times \mathcal{A}^{O}),$$

so $\mathcal{T}_{\nu}^{\mathrm{BI}}$ maps this Banach space into itself.

For each (s, a), define

$$\Phi_{s,a}(x) := \left(1 - \sum_{m \in \mathcal{S}_{\mathrm{demo}}^{M}} \nu(m \mid s)\right)x + \sum_{m \in \mathcal{S}_{\mathrm{demo}}^{M}} \nu(m \mid s)\max\{x, b_{\mathrm{Q}}^{m}(s,a)\}.$$

By Eq. (13),

$$(\mathcal{T}_{\nu}^{\mathrm{BI}}Q)(s,a) = \Phi_{s,a}((\mathcal{T}^{\star}Q)(s,a)). \tag{16}$$

Because $\nu(\cdot \mid s)$ is a sub-probability kernel and both $x \mapsto x$ and $x \mapsto \max\{x, b_{\mathrm{Q}}^m(s,a)\}$ are 1-Lipschitz, for any x, y,

$$\begin{aligned}|\Phi_{s,a}(x) - \Phi_{s,a}(y)| &\le \left(1 - \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)\right) |x - y| \\ &\quad + \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s) \left|\max\{x, b_{\mathrm{Q}}^m(s,a)\} - \max\{y, b_{\mathrm{Q}}^m(s,a)\}\right| \\ &\le \left[1 - \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s) + \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)\right] |x - y| \\ &= |x - y|.\end{aligned}$$

In the penultimate inequality, we used the 1-Lipschitz property of each mapping and the fact that the weight

$$1 - \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)$$

of the identity component and the total probability mass

$$\sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)$$

of the finite-sum component add up to 1.

Moreover, for any $Q_1, Q_2 \in \mathcal{B}_b(\mathcal{S}^O \times \mathcal{A}^O)$,

$$\left|\sup_{a'} Q_1(s', a') - \sup_{a'} Q_2(s', a')\right| \le \|Q_1 - Q_2\|_\infty.$$

Therefore, Eq. (12) gives

$$\|\mathcal{T}^\star Q_1 - \mathcal{T}^\star Q_2\|_\infty \le \gamma \|Q_1 - Q_2\|_\infty.$$

Combining this with Eq. (16),

$$\begin{aligned}\|\mathcal{T}_\nu^{\mathrm{BI}} Q_1 - \mathcal{T}_\nu^{\mathrm{BI}} Q_2\|_\infty &\le \|\mathcal{T}^\star Q_1 - \mathcal{T}^\star Q_2\|_\infty \\ &\le \gamma \|Q_1 - Q_2\|_\infty.\end{aligned}$$

Hence, $\mathcal{T}_\nu^{\mathrm{BI}}$ is a γ-contraction and, by the Banach fixed-point theorem, has a unique fixed point Q_ν^{BI}.

We next prove Eq. (8). Because $x \mapsto x$ and $x \mapsto \max\{x, b_{\mathrm{Q}}^m(s,a)\}$ are both monotone, $\Phi_{s,a}$ is monotone. The operator $\mathcal{T}^\star$ is also monotone, and hence Eq. (16) implies that $\mathcal{T}_\nu^{\mathrm{BI}}$ is monotone. Moreover, because the second term in Eq. (4) is nonnegative,

$$\mathcal{T}_\nu^{\mathrm{BI}} Q \ge \mathcal{T}^\star Q.$$

Therefore, from $Q^{O,\star} = \mathcal{T}^\star Q^{O,\star}$,

$$Q^{O,\star} \le \mathcal{T}_\nu^{\mathrm{BI}} Q^{O,\star} \le (\mathcal{T}_\nu^{\mathrm{BI}})^2 Q^{O,\star} \le \cdots.$$

Since $\mathcal{T}_\nu^{\mathrm{BI}}$ is a γ-contraction,

$$\|(\mathcal{T}_\nu^{\mathrm{BI}})^k Q^{O,\star} - Q_\nu^{\mathrm{BI}}\|_\infty \to 0.$$

Taking $k \to \infty$ in the above chain of inequalities yields

$$Q^{O,\star}(s,a) \le Q_\nu^{\mathrm{BI}}(s,a) \qquad (\forall (s,a)). \tag{17}$$

For the remainder of the proof, define

$$\zeta_\nu(s,a) := \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s) \left(b_{\mathrm{Q}}^m(s,a) - (\mathcal{T}^\star Q_\nu^{\mathrm{BI}})(s,a)\right)_+. \tag{18}$$

By the fixed-point equation for Q_ν^{BI},

$$Q_\nu^{\mathrm{BI}}(s,a) = (\mathcal{T}^\star Q_\nu^{\mathrm{BI}})(s,a) + \zeta_\nu(s,a). \tag{19}$$

Let

$$V_\nu^{\mathrm{BI}}(s) := \sup_a Q_\nu^{\mathrm{BI}}(s,a).$$

Since $\pi_\nu^\dagger$ assigns probability 1 to the argmax set of Q_ν^{BI} at every state,

$$V_\nu^{\mathrm{BI}}(s) = \mathbb{E}_{A\sim\pi_\nu^\dagger(\cdot|s)}\left[R^O(s,A) + \gamma \mathbb{E}_{s'\sim P^O(\cdot|s,A)}[V_\nu^{\mathrm{BI}}(s')] + \zeta_\nu(s,A)\right].$$

Comparing this with $V^{O,\pi_\nu^\dagger}$ gives

$$V_\nu^{\mathrm{BI}}(s_0) - V^{O,\pi_\nu^\dagger}(s_0) = \mathbb{E}_{\pi_\nu^\dagger}\left[\sum_{t=0}^{\infty} \gamma^t \zeta_\nu(S_t^O, A_t^O) \mid S_0^O = s_0\right]. \tag{20}$$

On the other hand, Eq. (17) and the monotonicity of $\mathcal{T}^\star$ imply

$$(\mathcal{T}^\star Q_\nu^{\mathrm{BI}})(s,a) \geq Q^{O,\star}(s,a).$$

Combining this with Lemma 1, for each m we obtain

$$\begin{aligned} b_{\mathrm{Q}}^m(s,a) - (\mathcal{T}^\star Q_\nu^{\mathrm{BI}})(s,a) &\leq b_{\mathrm{Q}}^m(s,a) - Q^{O,\star}(s,a) \\ &\leq \Gamma_{\mathrm{Q}}^{m,\star}(s,a). \end{aligned}$$

By monotonicity of the positive-part operator, multiplying by $\nu(m \mid s)$ and summing over $m \in \mathcal{S}_{\mathrm{demo}}^M$ then gives, by the definition of $\zeta_\nu(s,a)$,

$$0 \leq \zeta_\nu(s,a) \leq \Gamma_{\mathrm{Q},+}^{\nu,\star}(s,a). \tag{21}$$

Finally, Eq. (17) implies $V^{O,\star}(s_0) \leq V_\nu^{\mathrm{BI}}(s_0)$. Using Eqs. (20) and (21) in turn, we obtain

$$\begin{aligned} V^{O,\star}(s_0) - V^{O,\pi_\nu^\dagger}(s_0) &\leq V_\nu^{\mathrm{BI}}(s_0) - V^{O,\pi_\nu^\dagger}(s_0) \\ &= \mathbb{E}_{\pi_\nu^\dagger}\left[\sum_{t=0}^{\infty} \gamma^t \zeta_\nu(S_t^O, A_t^O) \mid S_0^O = s_0\right] \\ &\leq \mathbb{E}_{\pi_\nu^\dagger}\left[\sum_{t=0}^{\infty} \gamma^t \Gamma_{\mathrm{Q},+}^{\nu,\star}(S_t^O, A_t^O) \mid S_0^O = s_0\right]. \end{aligned}$$

Because $\Gamma_{\mathrm{Q},+}^{\nu,\star} \geq 0$, Tonelli's theorem allows us to interchange the expectation and the infinite sum. Moreover, for any nonnegative measurable function $g : \mathcal{S}^O \to [0,\infty]$, the definition in Eq. (14) gives

$$\sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi_\nu^\dagger}\left[g(S_t^O) \mid S_0^O = s_0\right] = \frac{1}{1-\gamma} \mathbb{E}_{s\sim d_{s_0}^{\pi_\nu^\dagger}}[g(s)],$$

where

$$g(s) := \mathbb{E}_{a\sim\pi_\nu^\dagger(\cdot|s)}\left[\Gamma_{\mathrm{Q},+}^{\nu,\star}(s,a)\right].$$

By using $A_t^O \mid S_t^O = s \sim \pi_\nu^\dagger(\cdot \mid s)$, we obtain

$$\begin{aligned} &\mathbb{E}_{\pi_\nu^\dagger}\left[\sum_{t=0}^{\infty} \gamma^t \Gamma_{\mathrm{Q},+}^{\nu,\star}(S_t^O, A_t^O) \mid S_0^O = s_0\right] \\ &= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{\pi_\nu^\dagger}\left[\Gamma_{\mathrm{Q},+}^{\nu,\star}(S_t^O, A_t^O) \mid S_0^O = s_0\right] \\ &= \frac{1}{1-\gamma} \mathbb{E}_{s\sim d_{s_0}^{\pi_\nu^\dagger}}\left[\mathbb{E}_{a\sim\pi_\nu^\dagger(\cdot|s)}\left[\Gamma_{\mathrm{Q},+}^{\nu,\star}(s,a)\right]\right]. \end{aligned}$$

This proves Eq. (8). □

B.2 Proof of Corollary 1

Proof. By assumption, $\Gamma^\star(s) \le 0$ for every $s \in \mathcal{S}^O_{\mathrm{base}} \cup \{s_\perp\}$, and hence $\Gamma^{m,\star}_{\mathrm{Q}}(s,a) \le 0$ for every $m \in \mathcal{S}^M_{\mathrm{demo}}$ and every (s,a). Lemma 1 therefore gives

$$b^m_{\mathrm{Q}}(s,a) \le Q^{O,\star}(s,a).$$

By the uniqueness of the fixed point of $\mathcal{T}^{\mathrm{BI}}_\nu$ established in Theorem 1, we conclude that

$$Q^{\mathrm{BI}}_\nu = Q^{O,\star}.$$

□

C Interpretation and Refinements

To complement Theorem 1 in the main text, this section addresses three points. First, we show that partitioning a trajectory according to bridges does not change the value of a fixed observer policy. Second, we isolate the conservativeness of the bound in the main text that arises from counting increases that do not alter the action ordering, and derive a tighter oracle bound. Third, we show that a valid bound can also be obtained using a value gap defined with respect to an arbitrary reference policy instead of an optimal policy, and characterize the resulting conservativeness.

C.1 Bridge-Partition Invariance of Policy Value

Suppose that the observer follows π^O from $s \in \mathcal{S}^O_{\mathrm{base}}$. Using $\tau^m(u)$ defined in the main text, we locally write the first hitting time of the bridge endpoint set $B^O(\xi(s))$ as

$$\tau_{\mathrm{hit}}(s) := \tau^{\xi(s)}(0).$$

Under a fixed Markov policy π^O, the Markov property of the discrete-time MDP also applies at the hitting time. Thus, partitioning the discounted return at $\tau_{\mathrm{hit}}(s)$ gives, for any $s \in \mathcal{S}^O_{\mathrm{base}}$,

$$V^{O,\pi^O}(s) = \mathbb{E}_{\pi^O}\left[\sum_{t=0}^{\tau_{\mathrm{hit}}(s)-1} \gamma^t \mathsf{R}^O_t + \mathbf{1}\{\tau_{\mathrm{hit}}(s) < \infty\}\gamma^{\tau_{\mathrm{hit}}(s)} V^{O,\pi^O}\left(S^O_{\tau_{\mathrm{hit}}(s)}\right) \mid S^O_0 = s\right]. \quad (22)$$

On trajectories for which $\tau_{\mathrm{hit}}(s) = \infty$, the first term is interpreted as the infinite discounted sum and the second term as 0. Thus, a bridge endpoint only specifies where the observer return is partitioned and does not change the value of a fixed observer policy π^O.

C.2 Oracle-Based Refinement of the Performance-Loss Bound

The bound in Theorem 1 accounts for all occurrences of $\Gamma^{\nu,\star}_{\mathrm{Q},+}$ along the baseline-injected greedy policy and therefore counts even value increases that do not actually change the policy choice as potential sources of performance loss. In this section, we derive a tighter bound based on an indirect quantity that is more difficult to estimate, thereby clarifying the source of the conservativeness in Theorem 1. As a simple example illustrating this conservativeness, suppose that for some function $c : \mathcal{S}^O \to \mathbb{R}$,

$$Q^{\mathrm{BI}}_\nu(s,a) = Q^{O,\star}(s,a) + c(s) \qquad (\forall (s,a)). \quad (23)$$

Because $c(s)$ is shared by all actions at the same state, the action ordering is unchanged, and any greedy policy with respect to Q^{BI}_ν is also optimal in the original observer MDP. Hence, the actual performance loss is 0, whereas the right-hand side of Eq. (8) can be positive. Let

$$\Pi^{O,\star} := \{\pi^O \mid V^{O,\pi^O} = V^{O,\star}\}$$

denote the set of optimal policies in the observer MDP. By the assumptions in the main text, $\Pi^{O,\star} \neq \emptyset$. To distinguish baseline-induced increments that are compatible with the original optimal policies, define the discounted cumulative baseline-induced increment along a policy π by

$$Z^\pi_\nu(s) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t \zeta_\nu(S^O_t, A^O_t) \mid S^O_0 = s\right]. \quad (24)$$

Here, ζ_ν is the quantity defined in Eq. (18).

Proposition 1 (Oracle-based refinement of the performance-loss bound)**.** *For any initial state s_0,*

$$\begin{aligned} V^{O,\star}(s_0) - V^{O,\pi_\nu^\dagger}(s_0) &\le Z_\nu^{\pi_\nu^\dagger}(s_0) - \sup_{\pi \in \Pi^{O,\star}} Z_\nu^{\pi}(s_0) \\ &\le Z_\nu^{\pi_\nu^\dagger}(s_0) \\ &\le \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{s_0}^{\pi_\nu^\dagger}} \left[\mathbb{E}_{a \sim \pi_\nu^\dagger(\cdot|s)} \left[\Gamma_{\mathrm{Q},+}^{\nu,\star}(s,a) \right] \right]. \end{aligned} \tag{25}$$

In particular, if $\pi_\nu^\dagger \in \Pi^{O,\star}$, the right-hand side of the first inequality is 0*.*

Proof. By Eq. (19), Q_ν^{BI} is the optimal action-value function of the MDP obtained by adding ζ_ν as an additional reward to the original reward function R^O, and $\pi_\nu^\dagger$ is an optimal policy for this MDP. Therefore, for any $\pi \in \Pi^{O,\star}$,

$$V^{O,\pi_\nu^\dagger}(s_0) + Z_\nu^{\pi_\nu^\dagger}(s_0) \ge V^{O,\pi}(s_0) + Z_\nu^{\pi}(s_0) = V^{O,\star}(s_0) + Z_\nu^{\pi}(s_0).$$

Hence,

$$V^{O,\star}(s_0) - V^{O,\pi_\nu^\dagger}(s_0) \le Z_\nu^{\pi_\nu^\dagger}(s_0) - Z_\nu^{\pi}(s_0).$$

Taking the supremum over $\pi \in \Pi^{O,\star}$ gives the first inequality in Eq. (25). The second inequality follows from $\zeta_\nu \ge 0$. For the third inequality, by definition,

$$\begin{aligned} Z_\nu^{\pi_\nu^\dagger}(s_0) &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{s_0}^{\pi_\nu^\dagger}} \left[\mathbb{E}_{a \sim \pi_\nu^\dagger(\cdot|s)} [\zeta_\nu(s,a)] \right] \\ &\le \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{s_0}^{\pi_\nu^\dagger}} \left[\mathbb{E}_{a \sim \pi_\nu^\dagger(\cdot|s)} \left[\Gamma_{\mathrm{Q},+}^{\nu,\star}(s,a) \right] \right], \end{aligned}$$

where the last inequality follows from Eq. (21). Finally, suppose that $\pi_\nu^\dagger \in \Pi^{O,\star}$. Since $V^{O,\pi_\nu^\dagger}(\cdot) = V^{O,\star}(\cdot)$, for any $\pi \in \Pi^{O,\star}$ we have

$$Z_\nu^{\pi_\nu^\dagger}(s_0) \ge Z_\nu^{\pi}(s_0).$$

Therefore,

$$Z_\nu^{\pi_\nu^\dagger}(s_0) = \sup_{\pi \in \Pi^{O,\star}} Z_\nu^{\pi}(s_0),$$

and the right-hand side of the first inequality is 0. □

Regarding Proposition 1, the quantity

$$Z_\nu^{\pi_\nu^\dagger} - \sup_{\pi \in \Pi^{O,\star}} Z_\nu^{\pi}$$

is 0 in cases such as Eq. (23), and thus exactly captures the actual performance loss in such cases. Note, however, that this is an oracle quantity requiring not only $\Pi^{O,\star}$ but also Q_ν^{BI}.

C.3 Conservative Reference-Policy Bound

The bound in Theorem 1 uses a value gap defined relative to the optimal value $V^{O,\star}$. Although $V^{O,\star}$ may be estimated in limited settings, it is generally unavailable. We therefore consider a conservative reference-policy bound based on a value gap defined with respect to an arbitrary reference policy $\tilde{\pi}^O$. From Eq. (5),

$$\begin{aligned} \Gamma^{\tilde{\pi}^O}(s) &:= b_{\mathrm{V}}(s) - V^{O,\tilde{\pi}^O}(s) \\ &= \Gamma^\star(s) + V^{O,\star}(s) - V^{O,\tilde{\pi}^O}(s) \ge \Gamma^\star(s), \end{aligned} \tag{26}$$

where the final inequality follows from $V^{O,\star} \ge V^{O,\tilde{\pi}^O}$. Thus, the value gap with respect to a reference policy can be interpreted as the gap relative to the optimal value plus the suboptimality of the reference policy. Analogously to the main text, define the following propagated value gap:

$$\Gamma_{\mathrm{Q}}^{m,\tilde{\pi}^O}(s,a) := \mathbb{E}_{\pi_{\mathrm{b}}^O} \left[\mathbf{1}\{\tau^m(u) < \infty\} \gamma^{\tau^m(u)} \Gamma^{\tilde{\pi}^O}(S_{u+\tau^m(u)}^O) \mid S_u^O = s, A_u^O = a \right]. \tag{27}$$

We further define the quantity obtained by marginalizing the positive part for each m with respect to ν as

$$\Gamma^{\nu,\tilde{\pi}^O}_{\mathrm{Q},+}(s,a) := \sum_{m\in\mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s)\big(\Gamma^{m,\tilde{\pi}^O}_{\mathrm{Q}}(s,a)\big)_+. \tag{28}$$

Corollary 2 (Conservative reference-policy bound)**.** *For any reference policy $\tilde{\pi}^O$ and initial state s_0,*

$$V^{O,\star}(s_0) - V^{O,\pi^\dagger_\nu}(s_0) \le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi^\dagger_\nu}_{s_0}}\left[\mathbb{E}_{a\sim\pi^\dagger_\nu(\cdot|s)}\left[\Gamma^{\nu,\tilde{\pi}^O}_{\mathrm{Q},+}(s,a)\right]\right]. \tag{29}$$

In particular, if $\tilde{\pi}^O$ is an optimal policy, the right-hand side coincides with the bound in Theorem 1.

Proof. By Eq. (26),

$$\Gamma^\star \le \Gamma^{\tilde{\pi}^O}.$$

The bridge propagation defined in Eqs. (6) and (27), the positive-part operator, and the weighted sum with the nonnegative kernel ν are all monotone. Therefore,

$$\Gamma^{\nu,\star}_{\mathrm{Q},+}(s,a) \le \Gamma^{\nu,\tilde{\pi}^O}_{\mathrm{Q},+}(s,a).$$

Applying this inequality to Theorem 1 gives Eq. (29). □

This corollary shows that replacing the difficult-to-evaluate gap based on the optimal value with a reference-policy gap yields a valid but generally more conservative upper bound. The suboptimality of the reference policy is reflected in the propagated value gap through the bridge endpoint. In particular, by choosing $\tilde{\pi}^O = \pi^O_{\mathrm{b}}$, $V^{O,\pi^O_{\mathrm{b}}}(s)$ is, in principle, estimable from observer rollouts.

D Entropy-Regularized Extension

The SAC critic estimates soft action values corresponding to returns that include entropy-regularization terms in addition to environmental rewards, whereas the state baseline b_{V}, based on the mentor return-to-go, is computed solely from environmental rewards. In this section, we fix $\alpha > 0$ and the behavior policy π^O_{b} and show that an analogous performance bound holds when the baseline-injection operator in the main text is extended to the entropy-regularized objective. We omit the dependence on π^O_{b} from the notation where convenient.

D.1 Soft Counterpart under a Fixed Behavior Policy

We assume that the observer's action space $\mathcal{A}^O$ has positive finite volume and that $\pi(a \mid s)$ denotes a measurable probability density with respect to Lebesgue measure $\lambda_{\mathcal{A}}$. For each policy π, we assume that there exists a constant $C^\pi_{\mathrm{ent}} < \infty$ such that

$$\sup_{s\in\mathcal{S}^O\setminus\{s_\perp\}} \mathbb{E}_{a\sim\pi(\cdot|s)}[|-\alpha\log\pi(a \mid s)|] \le C^\pi_{\mathrm{ent}}$$

holds. Below, suprema and argmax operations over policies range over all measurable stationary Markov policies satisfying this condition. Since $s_\perp$ is absorbing and represents episode termination, the entropy-regularization terms are set to 0 from $s_\perp$ onward. For the fixed behavior policy π^O_{b}, we define the state-level discounted entropy-to-go as the expected discounted sum of entropy-regularization terms starting from state s:

$$\mathcal{H}^{\pi^O_{\mathrm{b}}}_\alpha(s) := \mathbb{E}_{\pi^O_{\mathrm{b}}}\left[\sum_{n=0}^{\infty}\gamma^n\big(-\alpha\log\pi^O_{\mathrm{b}}(A^O_{u+n} \mid S^O_{u+n})\big) \mid S^O_u = s\right], \qquad \mathcal{H}^{\pi^O_{\mathrm{b}}}_\alpha(s_\perp) := 0. \tag{30}$$

Similarly, we define the action-level discounted entropy-to-go from a fixed state-action pair (s,a) as

$$\mathcal{E}^{\pi^O_{\mathrm{b}}}_\alpha(s,a) := \mathbb{E}_{\pi^O_{\mathrm{b}}}\left[\sum_{n=1}^{\infty}\gamma^n\big(-\alpha\log\pi^O_{\mathrm{b}}(A^O_{u+n} \mid S^O_{u+n})\big) \mid S^O_u = s,\ A^O_u = a\right]. \tag{31}$$

Note that the sum in Eq. (30) starts at $n = 0$, whereas the sum in Eq. (31) starts at $n = 1$. Indeed,

$$\mathcal{E}_\alpha^{\pi_{\mathrm{b}}^O}(s,a) = \gamma\, \mathbb{E}_{s' \sim P^O(\cdot|s,a)}\Big[\mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O}(s')\Big] \tag{32}$$

holds, and this one-step offset is consistent with the SAC critic target, in which entropy regularization begins with the next action. We define the theoretical soft action baseline conditioned on the start point mentor state m as

$$\begin{aligned} b^m_{\mathrm{Q},\alpha}(s,a) &:= b^m_{\mathrm{Q}}(s,a) + \mathcal{E}_\alpha^{\pi_{\mathrm{b}}^O}(s,a) \\ &= \mathbb{E}_{\pi_{\mathrm{b}}^O}\Bigg[\sum_{n=0}^{\tau^m(u)-1} \gamma^n \mathsf{R}^O_{u+n} + \sum_{n=1}^{\tau^m(u)-1} \gamma^n \big(-\alpha \log \pi_{\mathrm{b}}^O(A^O_{u+n} \mid S^O_{u+n})\big) \\ &\quad + \mathbf{1}\{\tau^m(u) < \infty\} \gamma^{\tau^m(u)} \Big[b_{\mathrm{V}}\big(S^O_{u+\tau^m(u)}\big) + \mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O}\big(S^O_{u+\tau^m(u)}\big)\Big] \mid S^O_u = s,\ A^O_u = a \Bigg]. \end{aligned} \tag{33}$$

The second equality follows by splitting the action-level entropy-to-go at the hitting time and applying the tower property and the Markov property at that hitting time. Thus, under a fixed behavior policy, explicitly accounting for the entropy-regularization terms within the bridge and the entropy-to-go from the bridge endpoint is equivalent to adding the full entropy-to-go from the initial state-action pair. The assumption on the one-step entropy-regularization term, together with $\gamma < 1$, ensures absolute integrability of the discounted sums and yields the bounds

$$\|\mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O}\|_\infty \le \frac{C_{\mathrm{ent}}^{\pi_{\mathrm{b}}^O}}{1-\gamma}, \qquad \|\mathcal{E}_\alpha^{\pi_{\mathrm{b}}^O}\|_\infty \le \frac{\gamma C_{\mathrm{ent}}^{\pi_{\mathrm{b}}^O}}{1-\gamma}.$$

Moreover, these functions are uniform limits of expectations of finite sums under measurable transition and policy kernels, so $\mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O} \in \mathcal{B}_b(\mathcal{S}^O)$ and $\mathcal{E}_\alpha^{\pi_{\mathrm{b}}^O} \in \mathcal{B}_b(\mathcal{S}^O \times \mathcal{A}^O)$. Consequently, each $b^m_{\mathrm{Q},\alpha}$ is also bounded and measurable.

For an observer policy π^O, we define the soft state value and the optimal soft state value as

$$V_\alpha^{O,\pi^O}(s) := \mathbb{E}_{\pi^O}\left[\sum_{n=0}^{\infty} \gamma^n \left(\mathsf{R}^O_{u+n} - \alpha \log \pi^O(A^O_{u+n} \mid S^O_{u+n})\right) \mid S^O_u = s\right], \tag{34}$$

$$V_\alpha^{O,\star}(s) := \sup_{\pi^O} V_\alpha^{O,\pi^O}(s). \tag{35}$$

We define the state value gap for the soft objective as

$$\Gamma^\star_\alpha(s) := b_{\mathrm{V}}(s) + \mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O}(s) - V_\alpha^{O,\star}(s), \qquad \Gamma^\star_\alpha(s_\perp) := 0. \tag{36}$$

We then define the state value gap propagated under the behavior policy as

$$\Gamma^{m,\star}_{\mathrm{Q},\alpha}(s,a) := \mathbb{E}_{\pi_{\mathrm{b}}^O}\Big[\mathbf{1}\{\tau^m(u) < \infty\} \gamma^{\tau^m(u)} \Gamma^\star_\alpha\big(S^O_{u+\tau^m(u)}\big) \mid S^O_u = s,\ A^O_u = a\Big], \tag{37}$$

and set

$$\Gamma^{\nu,\star}_{\mathrm{Q},\alpha,+}(s,a) := \sum_{m \in \mathcal{S}^M_{\mathrm{demo}}} \nu(m \mid s) \big(\Gamma^{m,\star}_{\mathrm{Q},\alpha}(s,a)\big)_+. \tag{38}$$

For $Q \in \mathcal{B}_b(\mathcal{S}^O \times \mathcal{A}^O)$, we define the soft Bellman optimality operator with fixed α as

$$(\mathcal{T}^\star_\alpha Q)(s,a) := R^O(s,a) + \gamma \mathbb{E}_{s' \sim P^O(\cdot|s,a)}\left[\sup_{\pi(\cdot|s')} \mathbb{E}_{a' \sim \pi(\cdot|s')}\left[Q(s',a') - \alpha \log \pi(a' \mid s')\right]\right]. \tag{39}$$

Here, the supremum is taken over all measurable policies satisfying the conditions of this section, and the continuation value inside the outer brackets is set to 0 when $s' = s_\perp$. For a nonterminal state s, the Gibbs variational formula gives

$$\sup_{\pi(\cdot|s)} \mathbb{E}_{a \sim \pi(\cdot|s)}\left[Q(s,a) - \alpha \log \pi(a \mid s)\right] = \alpha \log \int_{\mathcal{A}^O} e^{Q(s,a)/\alpha}\, \lambda_{\mathcal{A}}(da).$$

The right-hand side is measurable because it is obtained by integrating a measurable function and taking the logarithm, and its absolute value is bounded by $\|Q\|_\infty + \alpha|\log \lambda_{\mathcal{A}}(\mathcal{A}^O)|$. The supremum is attained by the measurable Gibbs policy with density proportional to $e^{Q(s,a)/\alpha}$, which also satisfies the integrability condition on the entropy-regularization term above. Hence, $\mathcal{T}_\alpha^\star$ maps $\mathcal{B}_b(\mathcal{S}^O \times \mathcal{A}^O)$ into itself. Moreover, the log-partition expression above is monotone and 1-Lipschitz in Q under the sup norm, so $\mathcal{T}_\alpha^\star$ is a monotone γ-contraction and has a unique fixed point by Banach's fixed-point theorem. We denote this fixed point by $Q_\alpha^{O,\star}$ and the corresponding Gibbs policy by $\pi_\alpha^{O,\star}$. Standard Bellman evaluation and comparison arguments show that $Q_\alpha^{O,\star}$ is the optimal soft action value and that $\pi_\alpha^{O,\star}$ attains the supremum in Eq. (35). The optimal soft state value defined in Eq. (35) can therefore be written equivalently as

$$V_\alpha^{O,\star}(s) = \sup_{\pi(\cdot|s)} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q_\alpha^{O,\star}(s,a) - \alpha \log \pi(a \mid s)\right]. \tag{40}$$

The soft baseline-injection operator $\mathcal{T}_{\nu,\alpha}^{\mathrm{BI}}$ is defined by replacing $\mathcal{T}^\star$ and b_{Q}^m in Eq. (4) with $\mathcal{T}_\alpha^\star$ and $b_{\mathrm{Q},\alpha}^m$, respectively.

Proposition 2 (Entropy-regularized performance-loss bound)**.** *Under the assumptions of this section, fix α and π_{b}^O. For a predefined sub-probability kernel ν, the operator $\mathcal{T}_{\nu,\alpha}^{\mathrm{BI}}$ has a unique fixed point $Q_{\nu,\alpha}^{\mathrm{BI}}$. Let $\pi_{\nu,\alpha}^\dagger$ be a measurable soft-greedy policy satisfying*

$$\pi_{\nu,\alpha}^\dagger(\cdot \mid s) \in \arg\max_{\pi(\cdot|s)} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q_{\nu,\alpha}^{\mathrm{BI}}(s,a) - \alpha \log \pi(a \mid s)\right]$$

at every state s; its existence follows from the Gibbs representation above. Then, for any initial state s_0,

$$V_\alpha^{O,\star}(s_0) - V_\alpha^{O,\pi_{\nu,\alpha}^\dagger}(s_0) \le \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{s_0}^{\pi_{\nu,\alpha}^\dagger}}\left[\mathbb{E}_{a\sim\pi_{\nu,\alpha}^\dagger(\cdot|s)}\left[\Gamma_{\mathrm{Q},\alpha,+}^{\nu,\star}(s,a)\right]\right]. \tag{41}$$

Furthermore, if the baseline is nonoptimistic relative to $V_\alpha^{O,\star}$, in the sense that $\Gamma_\alpha^\star(s) \le 0$ for all $s \in \mathcal{S}_{\mathrm{base}}^O \cup \{s_\perp\}$, then $Q_{\nu,\alpha}^{\mathrm{BI}} = Q_\alpha^{O,\star}$ and $\pi_{\nu,\alpha}^\dagger$ is optimal.

Proof. With α and π_{b}^O fixed, $\mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O}$, $\mathcal{E}_\alpha^{\pi_{\mathrm{b}}^O}$, and $b_{\mathrm{Q},\alpha}^m$ can be treated as fixed functions. The operator $\mathcal{T}_\alpha^\star$ is a monotone γ-contraction under the sup norm, and the scalar mapping $\Phi_{s,a}$ used for baseline injection is 1-Lipschitz and monotone, as in the proof of Theorem 1. Thus, the contraction and fixed-point uniqueness arguments apply directly after replacing $\mathcal{T}^\star$ and b_{Q}^m with $\mathcal{T}_\alpha^\star$ and $b_{\mathrm{Q},\alpha}^m$, respectively.

It remains to establish the soft counterpart of Lemma 1. Equation (36) gives $b_{\mathrm{V}} + \mathcal{H}_\alpha^{\pi_{\mathrm{b}}^O} = V_\alpha^{O,\star} + \Gamma_\alpha^\star$. Substituting this identity into the bridge endpoint term in Eq. (33) shows that the part excluding the contribution from $\Gamma_\alpha^\star$ is the soft action value of the policy that takes action a at time u, follows π_{b}^O until the bridge endpoint, and follows $\pi_\alpha^{O,\star}$ thereafter. Therefore,

$$b_{\mathrm{Q},\alpha}^m(s,a) \le Q_\alpha^{O,\star}(s,a) + \Gamma_{\mathrm{Q},\alpha}^{m,\star}(s,a).$$

Using this inequality, the remainder of the proof of Theorem 1 applies with $(\mathcal{T}^\star, b_{\mathrm{Q}}^m, Q_\nu^{\mathrm{BI}})$ replaced by $(\mathcal{T}_\alpha^\star, b_{\mathrm{Q},\alpha}^m, Q_{\nu,\alpha}^{\mathrm{BI}})$, yielding Eq. (41). Moreover, if $\Gamma_\alpha^\star \le 0$, then $\Gamma_{\mathrm{Q},\alpha}^{m,\star} \le 0$, and the same fixed-point uniqueness argument as in Corollary 1 yields $Q_{\nu,\alpha}^{\mathrm{BI}} = Q_\alpha^{O,\star}$. □

E Implementation Details

This section describes how MB-I2L is implemented on top of SAC.

E.1 SAC Critic Target and Entropy-to-Go Prediction

For completeness, we spell out here the standard SAC critic target denoted by y_t^{RL} in Section 3.3. Let $Q_{\theta_1}, Q_{\theta_2}$ denote the two critics in SAC with parameters θ_1, θ_2, let $\bar\theta_1, \bar\theta_2$ denote the parameters of the corresponding target critics, and write $\theta := (\theta_1, \theta_2)$. As described below, each critic network also has an entropy-to-go head, so θ_i collects the parameters of both heads of the i-th critic and of

their shared trunk. For a replay transition $(s_t^O, a_t^O, r_t^O, s_{t+1}^O, d_t)$, we sample $a'_{t+1} \sim \pi^O(\cdot \mid s_{t+1}^O)$ from the current actor and set

$$y_t^{\mathrm{RL}} := r_t^O + \gamma(1-d_t)\left[\min_{i\in\{1,2\}} Q_{\bar{\theta}_i}(s_{t+1}^O, a'_{t+1}) - \alpha \log \pi^O(a'_{t+1} \mid s_{t+1}^O)\right]. \tag{42}$$

Here, $d_t \in \{0,1\}$ is the episode-end indicator: $d_t = 1$ if the episode ends at this transition because the goal is reached, because an environment-specific failure termination occurs, or because the maximum episode length is reached, and $d_t = 0$ otherwise. Reaching the maximum episode length is thus treated as termination. Because the current time step is included in the observer state, this is consistent with the augmented MDP of Appendix F, in which $d_t = 1$ holds if and only if the process transitions to the absorbing terminal state $s_\perp$.

Corresponding to Eq. (33), the implementation trains an auxiliary predictor to approximate

$$E_\theta(s,a) \approx \mathbb{E}_{\pi^O}\left[\sum_{n=1}^{\infty} \gamma^n \big(-\alpha \log \pi^O(A_{u+n}^O \mid S_{u+n}^O)\big) \mid S_u^O = s,\ A_u^O = a\right]. \tag{43}$$

The expectation is taken under the current actor π^O rather than under the fixed behavior policy used in Eq. (31), so that the predictor corresponds to the policy under which the SAC critic target in Eq. (42) is defined. The predictor is implemented as an additional output head of each critic network rather than as a separate network. We write E_{θ_i} for the head of the i-th critic, which shares its trunk with Q_{θ_i}, and $E_{\bar{\theta}_i}$ for the corresponding head of the target critic, and we use the average $E_\theta := \frac{1}{2}\sum_{i\in\{1,2\}} E_{\theta_i}$ as the predictor value in Eq. (10). Using the same next-action sample a'_{t+1} as above, we set

$$y_t^E := \gamma(1-d_t)\left[\frac{1}{2}\sum_{i\in\{1,2\}} E_{\bar{\theta}_i}(s_{t+1}^O, a'_{t+1}) - \alpha \log \pi^O(a'_{t+1} \mid s_{t+1}^O)\right], \tag{44}$$

and define the loss of the heads as the mean squared error (MSE) between each prediction and this target,

$$\mathcal{L}_E(\theta) := \frac{1}{2}\sum_{i\in\{1,2\}} \mathrm{MSE}\big(E_{\theta_i}(s_t^O, a_t^O), y_t^E\big), \tag{45}$$

where the MSE is averaged over the sampled minibatch. The heads are trained jointly with the critics: θ_1 and θ_2 are updated by a gradient step on

$$\mathcal{L}_{\mathrm{critic}}(\theta) := \frac{1}{2}\sum_{i\in\{1,2\}} \mathrm{MSE}\big(Q_{\theta_i}(s_t^O, a_t^O), y_t^{\mathrm{MB}}\big) + \mathcal{L}_E(\theta), \tag{46}$$

in which the standard SAC critic loss, with y_t^{MB} in place of y_t^{RL}, and $\mathcal{L}_E$ have equal weight. The gradient of $\mathcal{L}_E$ is not stopped at the shared trunk, so the representation shared by Q_{θ_i} and E_{θ_i} is shaped by both objectives. The value $E_\theta(s_t^O, a_t^O)$ used in $\widehat{b}^m_{\mathrm{Q},\alpha}$ is computed with the parameters before this update and, like all targets, is treated as a constant. Unlike Eq. (42), which takes the minimum over the two target critics, Eq. (44) averages the two target heads. The backup in Eq. (44) involves no maximization over actions, so it is not subject to the maximization bias that the minimum in Eq. (42) is intended to mitigate. Equation (44) corresponds to the Bellman recursion in Eq. (32), and the heads therefore predict the action-level entropy-to-go by bootstrapping, in the same way as a standard critic.

E.2 Bridge Tracking and Replay Association

This section describes how the bridge tracking in Section 3.3 is implemented. At most one bridge is tracked at a time. For the bridge in progress, the implementation keeps its start point mentor state m and references to the replay transitions stored since the bridge started. When the bridge is completed at a nonterminal endpoint, m and the endpoint observer state are attached to these transitions, and the endpoint immediately becomes the start point of the next bridge; each replay transition is therefore associated with at most one bridge, and all transitions of a bridge share the same m. A sampled transition is subject to baseline injection only if it belongs to a completed bridge with a nonterminal

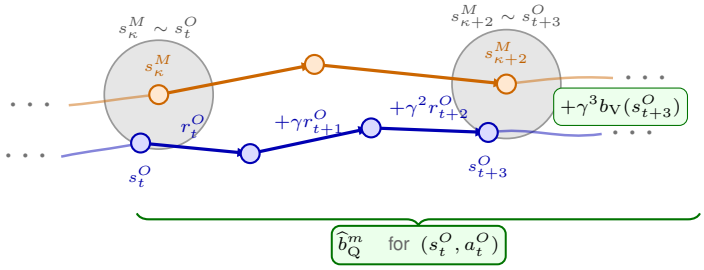


Figure 5: Construction of the sampled action baseline $\widehat{b}^m_{\mathrm{Q}}(s^O_t, a^O_t)$ in MB-I2L. The sampled action baseline $\widehat{b}^m_{\mathrm{Q}}$ is constructed from the discounted cumulative environmental rewards up to the bridge endpoint and the state baseline b_{V} at the endpoint.

Algorithm 1 MB-I2L critic update with SAC

1: Sample a minibatch $\mathcal{B}$ of replay transitions $(s^O_t, a^O_t, r^O_t, s^O_{t+1}, d_t)$ together with their associated bridge information, if available.
2: **for** each transition in $\mathcal{B}$ **do**
3: Sample $a'_{t+1} \sim \pi^O(\cdot \mid s^O_{t+1})$.
4: Compute y^{RL}_t and y^E_t with the target parameters $\bar{\theta}_1, \bar{\theta}_2$ using Eqs. (42) and (44).
5: **if** the transition satisfies the condition for baseline injection (Section 3.3) **then**
6: Construct $\widehat{b}^m_{\mathrm{Q}}$ using Eq. (9) and $\widehat{b}^m_{\mathrm{Q},\alpha}$ using Eq. (10), where $E_\theta = \frac{1}{2}(E_{\theta_1} + E_{\theta_2})$ is evaluated at the current parameters.
7: Compute L_t and y^{MB}_t using Eq. (11).
8: **else**
9: Set $L_t \leftarrow 0$ and $y^{\mathrm{MB}}_t \leftarrow y^{\mathrm{RL}}_t$.
10: **end if**
11: **end for**
12: Update θ_1, θ_2 by a gradient step on $\mathcal{L}_{\mathrm{critic}}(\theta)$ in Eq. (46), which trains the action-value heads toward y^{MB}_t and the entropy-to-go heads toward y^E_t jointly.
13: Update the actor and the entropy coefficient as in standard SAC, and update the target parameters $\bar{\theta}_1, \bar{\theta}_2$, including their entropy-to-go heads, by Polyak averaging as in standard SAC.

endpoint; transitions before the observer first reaches $\mathcal{S}^O_{\mathrm{base}}$, transitions of the bridge still in progress, and transitions of bridges that end at $s_\perp$ receive no baseline injection. The sampled action baseline in Eq. (9) is evaluated from the associated transitions when a transition is sampled for a critic update; since the replay buffer evicts transitions in insertion order, the later transitions of the bridge required for this evaluation are still stored at that time. Algorithm 1 summarizes the critic update procedure of MB-I2L, in which all targets are treated as constants with respect to the parameters being updated.

E.3 Relation to the Theoretical Operator

In the theoretical soft baseline-injection operator $\mathcal{T}^{\mathrm{BI}}_{\nu,\alpha}$, the effect of the start point mentor state m is marginalized out at each observer state s using $\nu(m \mid s)$. In the implementation, by contrast, this marginalization is not computed explicitly; sampling a transition from the replay buffer implicitly samples the start point mentor state m associated with that transition. As described in Appendix A.3, a kernel ν can be constructed from a predefined sampling rule, and the transition sampling used here can be placed in correspondence with such a construction. Moreover, whereas the theoretical operator defines the baseline-induced increment from the soft Bellman backup $(\mathcal{T}^\star_\alpha Q)(s,a)$ and the expected soft action baseline $b^m_{\mathrm{Q},\alpha}(s,a)$, the implementation performs the same comparison sample-wise, between the standard SAC critic target y^{RL}_t and the sampled soft action baseline $\widehat{b}^m_{\mathrm{Q},\alpha}(s,a)$. Consequently, the expectation of y^{MB}_t over this sampling does not in general coincide with the value produced by the theoretical operator. In particular, even if each sampled quantity were an unbiased estimator of the corresponding expected quantity, the expectation of a sample-wise maximum is in general at least the maximum of the expectations, so the sample-wise comparison is biased toward

optimism relative to the theoretical operator. More strongly, it is known that, for general families of distributions, no unbiased estimator of the maximum expected value can be constructed from finite samples (Van Hasselt, 2013). This discrepancy therefore reflects a general statistical difficulty in estimating a maximum expected value from samples, rather than an approximation error specific to MB-I2L. We accordingly do not assume that an unbiased estimator of the theoretical operator is available, and instead apply the same baseline-injection rule to the sampled quantities. We also note that the clipping in Eq. (11) is an implementation-level difference introduced for stabilization and is not part of the theoretical operator in Proposition 2. Moreover, whereas the expectation in Eq. (3) also covers bridges ending at $s_\perp$, such bridges are excluded from baseline injection, so the sampled action baseline is drawn only from bridges with nonterminal endpoints.

F Finite-Horizon Setting

Our experiments use episodes of maximum length H that may also terminate earlier, whereas the theory is stated for discounted infinite-horizon MDPs. As is standard, such a problem is a time-homogeneous discounted MDP on the augmented state space $\widetilde{\mathcal{S}}^O := (\mathcal{S}^O \times \{0, \dots, H-1\}) \cup \{s_\perp\}$, in which (s, t) moves to $(s', t+1)$ under the original dynamics and moves to the absorbing state $s_\perp$ upon early termination or after an action at $t = H-1$; time-dependent policies on the original problem correspond to stationary Markov policies on this MDP with the same values, and the standing assumptions are preserved. Since the mentor–observer correspondence does not use time, we set

$$\widetilde{\mathcal{S}}^O_{\mathrm{base}} := \mathcal{S}^O_{\mathrm{base}} \times \{0, \dots, H-1\}, \quad \widetilde{\xi}((s,t)) := \xi(s), \quad \widetilde{\xi}(s_\perp) := s_\perp,$$
$$\widetilde{b}_{\mathrm{V}}((s,t)) := b_{\mathrm{V}}(s), \quad \widetilde{b}_{\mathrm{V}}(s_\perp) := 0,$$

and define the other components of MB-I2L analogously. The mismatch caused by using the same baseline for different remaining horizons is thus absorbed into the value gap; for nonnegative rewards, $\widetilde{V}^{O,\star}((s,t))$ is nonincreasing in t, so the baseline becomes more optimistic as fewer steps remain. Theorem 1, Corollary 1, and, with entropy-regularization terms set to 0 from $s_\perp$ onward, Proposition 2 therefore apply to the augmented MDP and hence to the corresponding time-dependent policies.

G Additional Experimental Details and Results

This section reports experiments and details that could not be included in the main text. Through these experiments, in addition to supplementing Q2 and Q3 of the main text, we answer the following questions about MB-I2L.

- **Q4.** Are the mechanisms of MB-I2L justified?
  - **Q4.1.** Is the neighborhood relation alone sufficient, or does the bridge mechanism contribute?
  - **Q4.2.** Is clipping effective?
  - **Q4.3.** Is the accuracy of the predictor E_θ sufficient?
- **Q5.** How does the quality of the mentor data relate to the performance of MB-I2L?
- **Q6.** Can excessive optimism be counteracted dynamically during learning?
- **Q7.** What kinds of failures does MB-I2L exhibit?

Our implementation is based on Stable-Baselines3 (SB3) (Raffin et al., 2021); we provide the hyperparameters used in our experiments in Table 4.

G.1 Comparison Methods

We describe the reason for including each comparison method and the corresponding implementation details. CE was included because it is a representative method for assisting exploration under sparse rewards. CE' was included in order to examine what happens when the mentor data are prioritized, by giving the bonus only in the neighborhoods of mentor states. (Because CE' uses

Table 4: SAC and MB-I2L hyperparameters.

Hyperparameter	Value	Note
Optimizer	Adam	SB3 default
Replay buffer size	1M	SB3 default
Soft update (Polyak) coefficient	0.005	SB3 default
Discount factor γ	0.99	SB3 default
Train frequency	1 step	SB3 default
Gradient steps per update	1	SB3 default
Target entropy	$-\dim(\mathcal{A})$	SB3 default (auto)
Target update interval	1	SB3 default
Network architecture	$[256, 256]$	SB3 default
Activation	ReLU	SB3 default
Learning rate	1×10^{-4}	
Learning starts	5000	
Batch size	32	
Entropy coefficient α	auto (initial value 0.001)	
c_{inc}	0.01	

the same counts as CE, hash collisions can increase counts at states unrelated to the neighborhoods and thereby reduce the bonus; we retain this implementation for simplicity.) HER was included because it is a representative method for mitigating the difficulty of sparse rewards in goal-reward tasks. The pseudo goals in HER were matched to the goal representation of each environment, that is, the xy coordinates in the maze environments and the x coordinate in HopperHurdle. PBRS was included because it is widely used and has the desirable property of preserving the optimal policy. In this work, we use the state baseline $b_{\mathrm{V}}(s^O)$, that is, the mentor return-to-go from the assigned mentor state $\xi(s^O)$, as the potential when the observer state s^O is in a neighborhood of a mentor state, and set the potential to 0 otherwise. Although PBRS uses the same b_{V}, its shaping terms sum to $-\Phi(s_0^O)$ along every trajectory (Ng et al., 1999), and their effect has been observed to vanish as learning progresses (Imagawa & Enokida, 2026). In contrast, the increment L_t in Eq. (11) vanishes only when $y_t^{\mathrm{RL}} \geq \widehat{b}^m_{\mathrm{Q},\alpha}$ (cf. Figure 8b). The parameters of PWIL follow its official implementation (`https://github.com/google-research/google-research/tree/master/pwil`). For ORCA, the parameters were adjusted because of the reward-scale issue described in the ORCA paragraph below. Note that CE' and PBRS use neighborhoods, whose definition and radius are the same as in MB-I2L. Likewise, the distance between mentor and observer state features used in PWIL and ORCA is the same as in MB-I2L. As for Ori. Rew., the original reward of the Gymnasium Hopper environment combines a reward for increasing the x coordinate of the center of mass, a survival reward, and a small control penalty. Moving forward while surviving without falling is therefore the condition for obtaining a higher reward, so the behavior of heading toward the goal while clearing the hurdle can in principle be reproduced; we therefore included it as a comparison method.

Let r_t^{int} denote the additional reward at time t in CE, CE', PWIL, and ORCA, and let r_t denote the ordinary reward. We used the modified reward $r_t + \lambda r_t^{\mathrm{int}}$ for CE, CE', PWIL, and ORCA, in place of the ordinary reward. In the reward computation of these four methods, $r_t^{\mathrm{int}} \in [0, 1]$. For PWIL and ORCA we set $\lambda = 0.001$, so that the scales match in the sense that, even if an intrinsic reward of 1 were obtained at every step, its undiscounted episode total would not exceed the extrinsic reward (PointMaze and HopperHurdle) or would exceed it by at most a factor of five (AntMaze). For CE and CE' we set $\lambda = 0.01$, which is larger because the count-based bonus decays as states are revisited.

PWIL. In the variant of PWIL that we use (PWIL-state, referred to simply as PWIL in this work), the Wasserstein distance is computed from distances between states only, and the intrinsic reward is computed from that distance. In this work, we use the feature distance $\rho(s_j^M, s_t^O)$ between the mentor state s_j^M with index j and the observer state s_t^O as the distance between states. The rest of the computation of r_t^{int} follows the original work.

ORCA. Our setting differs from that of the original ORCA in two respects, which makes adjustments necessary. First, ORCA is designed to be used as the sole reward, whereas we combine it with the sparse extrinsic reward; its scale relative to the extrinsic reward is handled by the weight λ described above. Second, the original experiments use demonstrations of roughly 100–200 frames, whereas the mentor data for PointMaze in this work have more than 700 steps. Because the ORCA reward accumulates the terms $\exp(-\rho^2)$ multiplicatively along the entire mentor trajectory (see the equations below), it becomes extremely small when the mentor trajectory is long and when many mentor states are far from the observer, as is the case early in training.

To prevent the intrinsic reward from vanishing, we introduce a scale-adjustment parameter c_{ORCA}, which controls how quickly each term decays with distance, and we subsample the mentor data used in the reward computation. The parameter c_{ORCA} corresponds to the temperature parameter of Huey et al. (2025) through $1/c_{\mathrm{ORCA}}^2$, with the squared feature distance ρ^2 playing the role of their distance d, and $c_{\mathrm{ORCA}} = 1$ recovers the official implementation at `https://github.com/portal-cornell/orca`. Because the mentor steps are indexed by $0, 1, \ldots, T-1$, the intrinsic reward of ORCA is given by

$$P_{t,j} = \exp\bigl(-\rho(s_j^M, s_t^O)^2 / c_{\mathrm{ORCA}}^2\bigr), \tag{47}$$

$$C_{t,j} = \max\{C_{t-1,j}, C_{t,j-1} P_{t,j}\}, \tag{48}$$

$$r_t^{\mathrm{int}} = C_{t,T-2} P_{t,T-1}. \tag{49}$$

Following Huey et al. (2025), the recursion is initialized by $C_{0,0} = P_{0,0}$, by $C_{0,j} = C_{0,j-1} P_{0,j}$ for $j \geq 1$, and by $C_{t,0} = \max\{C_{t-1,0}, P_{t,0}\}$ for $t \geq 1$. Also following Huey et al. (2025), r_t^{int} combines $C_{t,T-2}$ with $P_{t,T-1}$ rather than using $C_{t,T-1}$ directly, so that the reward is obtained only while the observer remains at the last mentor state; since $C_{t,j}$ is nondecreasing in t, using $C_{t,T-1}$ would retain the reward after the observer has left it.

On PointMaze, we tried $(128, 16)$, $(64, 16)$, $(32, 16)$, $(32, 8)$, and $(32, 4)$ for the pair consisting of the number of subsampled mentor steps and c_{ORCA}, and adopted $(32, 16)$, which performed best. This tuning is applied only to ORCA and therefore favors it. Note that ORCA can be affected by neighborhood matches across a wall, but because its reward takes the order of the mentor states into account, we expect no such effect unless the subsampling is coarse enough to make a detour negligible.

G.2 Additional Details on the Environments

As in the usual off-policy reinforcement learning procedure, evaluation rollouts are performed every fixed number of steps during training, and we mainly report their success rate. Each evaluation uses 20 episodes and the deterministic policy given by the mean action, and the evaluation interval is given for each environment below. The initial state of the observer is randomized at each reset, which is why each evaluation uses multiple episodes.

G.2.1 PointMaze

In this environment, both the observer and the mentor are balls, and a goal is placed at a fixed position. A reward of 1 is given when the goal is reached, that is, when the Euclidean distance between the center of mass and the goal falls below 0.45. The episode length is 1000 steps, and an episode ends either after 1000 steps or when the goal is reached. The mentor trajectory is shown in Figure 2 and was generated by proportional-derivative (PD) control. With the mentor data we used, the mentor reached the goal in 720 steps. In PointMaze, a mentor state and an observer state are neighbors when the distance between the xy coordinates of their centers of mass is below the neighborhood radius $\eta_{\mathrm{nb}} = 0.1$. Training was run for 1M steps, and evaluation was performed every 10k steps.

G.2.2 AntMaze

In this environment, the observer agent is an Ant robot, whereas the mentor is a ball as in PointMaze. The mentor reached the goal in 164 steps. The scale is four times that of PointMaze, and the goal is reached when the Euclidean distance between the center of mass and the goal falls below 1.8. The Ant robot can fall over, but the original environment does not terminate an episode upon falling.

In this work, for training efficiency, we terminate the episode at the moment the Ant falls over. Specifically, we terminate the episode (i.e., set $d_t = 1$) when the torso height z leaves the range $0.2 \leq z \leq 1.0$, or when the absolute value of the scalar component of the quaternion representing the torso orientation falls below 0.3. The episode length is 5000 steps, and an episode ends after 5000 steps, when the Ant falls over, or when the goal is reached. In AntMaze, a mentor state and an observer state are neighbors when the distance between the xy coordinates of their centers of mass is below the neighborhood radius $\eta_{\mathrm{nb}} = 1$. Since the maze of the Gymnasium-Robotics AntMaze is four times the scale of PointMaze, the radius was enlarged accordingly; we used the round value 1 rather than the exactly scale-matched value 0.4. This value is still well below the width of the narrowest wall at this scale, so it does not induce the across-wall correspondences examined in the failure analysis in the main text. Training was run for 5M steps, and evaluation was performed every 100k steps. Maximizing the reward in this environment requires learning the motion of the complex Ant robot and navigation toward the goal at the same time, while a reward is obtained only upon reaching the goal; AntMaze is therefore an extremely difficult sparse-reward environment.

G.2.3 HopperHurdle

This environment wraps the Gymnasium Hopper task and adds a hurdle and a goal. The hurdle is a rectangular region in the xz plane that the center of mass of the Hopper must not enter; it has no collision geometry and does not interact with the rest of the Hopper's body. In addition to the original Gymnasium termination condition, under which an episode terminates when the Hopper becomes unhealthy, an episode terminates when the center of mass enters the hurdle. An episode also terminates when the goal is reached or when the maximum episode length is reached. The goal is regarded as reached when the x coordinate of the center of mass exceeds the x coordinate of the goal, and a reward of 1 is then given; no reward is given otherwise. The maximum episode length is 1000 steps. We prepared two variants, Short and Long, which differ in the distance to the goal. The Hopper starts at $x = 0$, and the x coordinate of the goal is 1 for Short and 3 for Long. A single hurdle of width 0.1 and height 0.7 is placed with its center midway between the initial position and the goal position, that is, at $x = 0.5$ for Short and $x = 1.5$ for Long. Since the height of the center of mass of the standing Hopper is about 0.6, the hurdle is low enough to be cleared with a small jump or on tiptoe. The mentor data are generated by a ball of radius 0.6 that bounces with a coefficient of restitution of 1, with the bounce set so that the apexes occur exactly at $x = 0.5, 1.5, \ldots$. The mentor does not pass the hurdle by a narrow margin but with some clearance: the center of mass of the ball follows a trajectory that passes 0.4 above the top of the hurdle. With this construction, the mentor bounces once before reaching the goal in Short and three times in Long, and the hurdle lies at the first apex in Short and at the second apex in Long. The Long variant therefore requires the observer to jump repeatedly, which is consistent with the discussion of the failure on HopperHurdle-Long in the main text. A mentor state and an observer state are neighbors when the distance between the xz coordinates of their centers of mass is below the neighborhood radius $\eta_{\mathrm{nb}} = 0.1$ and the signs of their vertical velocities agree. Training was run for 1M steps for Short and 2M steps for Long, and evaluation was performed every 100k steps. The mentor reached the goal in 81 steps for Short and 241 steps for Long. In this environment, the Hopper moves only in the xz plane and has fewer joints than the Ant, so the dimension of the state space is smaller. However, the Hopper must crouch before it can jump high, whereas the ball mentor simply bounces with a coefficient of restitution of 1; the difference in embodiment is therefore pronounced, which makes this environment difficult in a sense different from AntMaze.

G.3 Analysis of the Number of Steps to the Goal in PointMaze

We supplement the analysis of Q2 reported in the main text, namely whether the observer can reach a trajectory better than the mentor's. Table 5 evaluates each method in PointMaze through the number of steps that the policy after training takes to reach the goal. SAC, HER, and PBRS never reached the goal in any evaluation rollout, and the evaluation success rate of ORCA was almost always zero (Table 1). The episode length is therefore not informative for these methods, and they are omitted from the table. In the following analysis, we call a seed a successful seed when the success rate of the policy after training is nonzero; since each evaluation uses 20 episodes, this means that the goal was reached in at least one of them. Note that, for every method in this experiment, each seed either succeeded in all 20 episodes or failed in all of them. MB-I2L reaches the goal in 326.12 ± 8.03 steps, whereas the mentor requires 720 steps. MB-I2L therefore learns behavior that is more efficient

Table 5: Mean and standard deviation of the episode length of the policy after training in PointMaze, over all seeds and over the successful seeds only. An episode that does not reach the goal terminates at the maximum episode length of 1000 steps and is counted as such in the all-seeds row.

	Ours	CE	CE'	PWIL
Episode length	326.12 ± 8.03	465.00 ± 267.62	732.42 ± 327.77	643.90 ± 302.82
Successful seeds	10/10	8/10	4/10	6/10
Episode length	326.12 ± 8.03	331.24 ± 8.73	331.05 ± 9.23	406.50 ± 109.24

than the mentor trajectory rather than merely reproducing it. CE does not use the mentor trajectory as a target, so the number of steps it takes on its successful seeds provides a reference for what is achievable in this environment without guidance from the mentor trajectory. Its value, 331.24 ± 8.73, is comparable to that of MB-I2L, which indicates that MB-I2L is not merely better than a suboptimal mentor but attains the same level as methods that do not follow the mentor trajectory at all. The more informative comparison is with PWIL, which, like MB-I2L, derives its guidance from the mentor states: restricted to its successful seeds, PWIL reaches the goal in 406.50 ± 109.24 steps, noticeably more than MB-I2L and with a much larger standard deviation, which suggests that PWIL is more strongly drawn toward the mentor trajectory. MB-I2L also succeeded on all 10 seeds, more than any of the compared methods in this table.

G.4 Sensitivity to the Neighborhood Radius η_{nb}

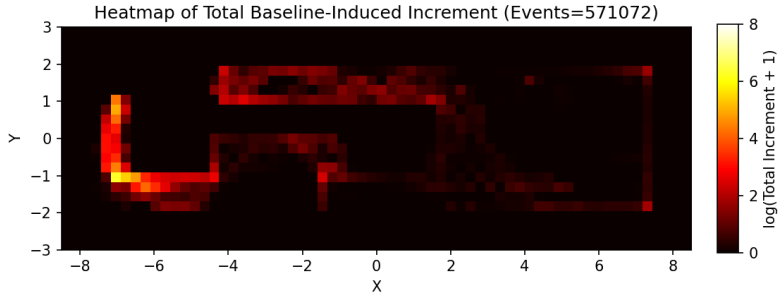
Here we supplement the results for Q3 in the main text, that is, the results obtained when the neighborhood radius is varied. In addition to the normalized AUC (Table 2) and the rate of initially moving left (Figure 4) reported in the main text, we also measured the cumulative baseline-induced increment and visualize it as a heatmap (Figure 6) for further analysis. Of the settings in Table 2, we show the three on which the following argument relies: the default radius $\eta_{\mathrm{nb}} = 0.1$, the radius $\eta_{\mathrm{nb}} = 2$ at which performance degrades, and its wall-aware counterpart $\eta_{\mathrm{nb}} = 2'$. The results show that, with $\eta_{\mathrm{nb}} = 2$, the baseline raises values toward the left, that is, toward the dead end. In contrast, this does not occur with $\eta_{\mathrm{nb}} = 0.1$ or with $\eta_{\mathrm{nb}} = 2'$. Since $\eta_{\mathrm{nb}} = 2'$ keeps the radius at 2 but excludes mentor–observer pairs separated by a wall, this comparison indicates that the value elevation toward the dead end is caused by across-wall correspondences rather than by the radius itself, which is consistent with the recovery of performance under $\eta_{\mathrm{nb}} = 2'$ in Table 2.

G.5 Ablation Studies

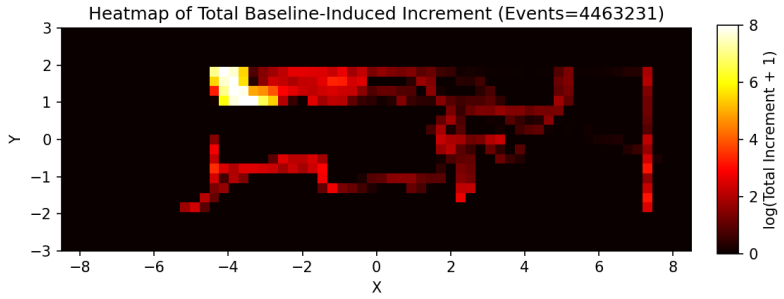
Regarding Q4, we examine the soundness of the mechanisms of MB-I2L by comparing them with modified variants. The results for the bridge, for clipping, and for the predictor E_θ are presented below.

G.5.1 Ablation Study on the Role of the Bridge

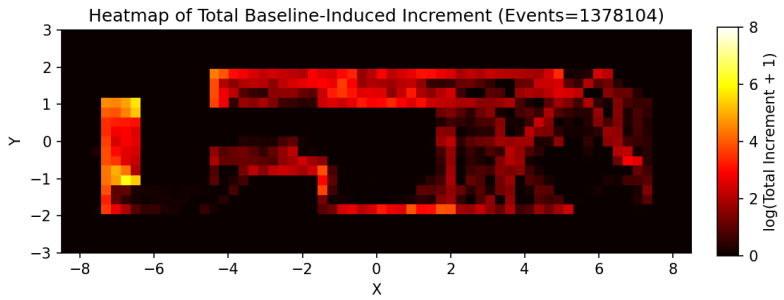
MB-I2L defines bridges and exploits the mentor return-to-go on the basis of them. In this section we compare the default MB-I2L, a variant that also allows backward bridges, and MB-I2L without bridges, in order to assess the effectiveness of the bridge mechanism. In the bidirectional variant, the condition on the progression of the mentor index in Eq. (1) is removed, which relaxes the condition for reaching a bridge endpoint. Under this relaxation the endpoint set also contains the neighborhood of the start point mentor state itself, so a bridge may terminate at the next step. In MB-I2L without bridges, baseline injection is performed when the next state lies in a neighborhood of a mentor state, using $r_t^O + \gamma b_{\mathrm{V}}(s_{t+1}^O)$ as the sampled action baseline; this corresponds to Eq. (9) with the bridge endpoint forced to be the next state. Table 6 shows the results. The variant without bridges performed substantially worse and with a much larger standard deviation, which indicates that propagating the baseline along a bridge, rather than only to the next state, makes the guidance both more effective and more reliable. In contrast, the bidirectional variant performs comparably to MB-I2L, so the forward-index condition has little effect in PointMaze.



(a) $\eta_{\mathrm{nb}} = 0.1$



(b) $\eta_{\mathrm{nb}} = 2$



(c) $\eta_{\mathrm{nb}} = 2'$

Figure 6: Heatmaps of the cumulative baseline-induced increment for different neighborhood radii η_{nb}. In each figure, "Events" denotes the number of times the baseline-induced increment was positive under the corresponding experimental setting.

G.5.2 Sensitivity Analysis for Clipping

We verify the soundness of clipping, the mechanism introduced in this work for stabilization, through experiments. Table 7 shows the normalized AUC obtained when the clipping parameter c_{inc} is varied. In PointMaze, the three settings were indistinguishable. In HopperHurdle-Short, by contrast, performance improved as c_{inc} decreased, and the unclipped setting $c_{\mathrm{inc}} = \infty$ almost never reached the goal. The result of the setting $c_{\mathrm{inc}} = 0.01$ used in our experiments was the best on average.

Table 6: Normalized AUC of evaluation success rate (mean ± SD) in PointMaze. Default uses 10 seeds and the two variants use 5 seeds each.

	Default	Bidirectional	No Bridge
PointMaze	0.77 ± 0.076	0.74 ± 0.072	0.37 ± 0.30

Table 7: Sensitivity analysis for clipping. Normalized AUC of evaluation success rate (mean ± SD). The setting $c_{\text{inc}} = 0.01$ used in our experiments uses 10 seeds, and the other two settings use 5 seeds each.

	$c_{\text{inc}} = 0.01$	$c_{\text{inc}} = 0.1$	$c_{\text{inc}} = \infty$
PointMaze	0.77 ± 0.076	0.75 ± 0.10	0.78 ± 0.061
HopperHurdle-Short	0.63 ± 0.27	0.28 ± 0.39	0.019 ± 0.036

G.6 Prediction Error of the Entropy-to-Go Heads

As described in the main text and in Appendix D, MB-I2L trains the entropy-to-go heads by Eqs. (44)–(45) so that the baseline constructed from the mentor return-to-go is made comparable to the entropy-regularized critic target of SAC. To confirm that this use of a predictor is sound, we measured the prediction MSE during training. Figure 7 shows the result. The prediction MSE decreases over roughly the first 0.3M steps and then remains at about 10^{-8} for the rest of training. A mean squared error of 10^{-8} corresponds to a root-mean-square Bellman residual of about 10^{-4}; even after the factor $1/(1-\gamma) = 100$ by which such a residual can be amplified into an error in the predicted value, the implied error is about 10^{-2}, which is small relative to the scale of the state baseline $b_{\text{V}} \leq 1$. The entropy-to-go heads therefore predict the action-level entropy-to-go with sufficient accuracy for the baseline construction in this experiment.

G.7 Quality of the Mentor Data and Its Effect on Performance

Regarding Q5, this section shows the behavior of MB-I2L for mentor data that differ in path and in maximum speed (four levels). In the PointMaze experiment of the main text, we used a mentor trajectory that takes a detour ('long') in order to make clear the advantage of MB-I2L in not being captured by the mentor trajectory. Here we also examine a mentor trajectory without the detour ('short'). Differences in maximum speed appear as differences in the number of steps of the mentor data. We prepared four levels of maximum speed—faster than the observer, the same as the observer, slower, and even slower—for each of the two paths, giving eight mentor datasets in total. For the fastest data in particular, the maximum speed exceeds what is attainable in the observer's environment, so in the sense that the observer cannot reproduce it, this setting is close to a cross-embodiment setting. The slower mentor data, by contrast, are reproducible by the observer. Note,

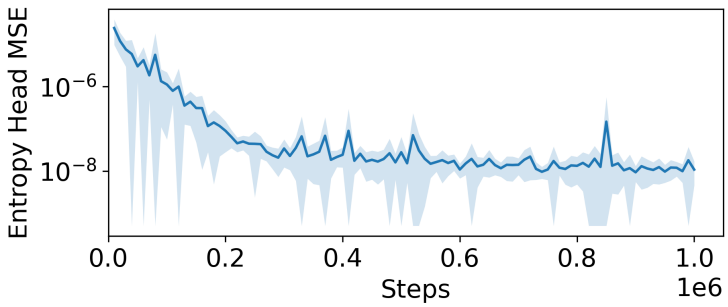
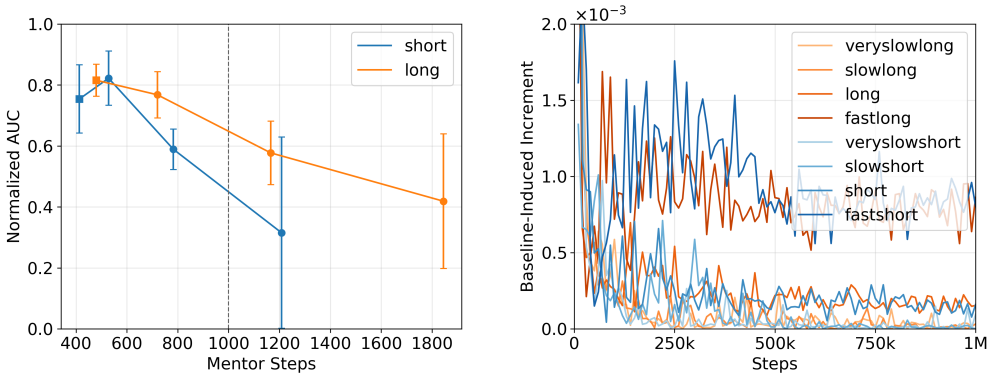


Figure 7: Prediction MSE of the entropy-to-go heads (Eq. (45)) during training in PointMaze. The solid line is the mean over seeds and the shaded region shows the standard deviation across seeds. Note the logarithmic vertical axis.



(a) Normalized AUC when using different mentor trajectories.

(b) Baseline-induced increment vs. step (rolling mean with window size 1000).

Figure 8: Effect of the quality of the mentor data in PointMaze. Both paths are shown in each panel. In (a), square markers indicate the settings whose maximum speed exceeds that attainable by the observer (and which the observer therefore cannot reproduce), and circular markers indicate the others; the dashed line indicates the observer episode length of 1000 steps. Each setting uses 5 seeds, except the long path with the same maximum speed as the observer, which is the setting used in the main text and uses 10 seeds.

however, that an observer episode lasts 1000 steps, so if reproduction would take longer than that, the observer cannot reproduce the entire mentor trajectory. Figure 8a shows the results.

Regardless of the path, performance improves as the number of mentor steps decreases. Comparing the two paths at a similar number of mentor steps, the short path, which is of higher quality in terms of the path, is not better than the long path; at around 700–800 mentor steps the long path is clearly better. The number of mentor steps, rather than the geometric quality of the path, therefore governs performance. An exception to the monotone trend is that, on the short path, the fastest mentor data performed slightly worse than the next fastest, although this difference is within the variability across seeds. The dashed line indicates the observer episode length of 1000 steps; beyond it the observer cannot reproduce the entire mentor trajectory, but performance degrades gradually rather than collapsing.

To analyze which mechanism of MB-I2L produces this behavior, we examined the magnitude of the baseline-induced increment. Figure 8b shows the mean baseline-induced increment over a mini-batch, averaged over seeds and smoothed with a rolling mean of window size 1000. The increment remains large throughout training for the two fastest mentor data, on both paths, whereas it decays for the others, and the slower the mentor the smaller the increment. Since fewer mentor steps means less discounting and hence a larger state baseline b_{V}, a smaller number of mentor steps produces a larger baseline-induced increment, which in turn yields better performance. A possible explanation for why the long path is not worse than the short path at a comparable number of mentor steps is that the longer path passes through a larger part of the maze, so that more observer states fall in a neighborhood and more bridges are formed. On the other hand, the improvement may saturate if the number of mentor steps is reduced too far, which is consistent with the fastest mentor on the short path performing slightly worse than the next fastest.

G.8 Adaptive Disabling of Baseline Injection (MB-I2L′) and Its Threshold Sensitivity

Regarding Q6, in this section we examine MB-I2L′, a variant that adaptively disables baseline injection for individual mentor states during learning in order to suppress the excessive optimism of MB-I2L. As verified in the experiments in the main text (Figure 3), MB-I2L exhibited better performance than the other methods, especially in AntMaze. On the other hand, as analyzed theoretically (Theorem 1) and empirically (Table 2), optimistic baseline injection can lead to suboptimal behavior. In MB-I2L′, once a mentor state s_κ^M satisfies the following three conditions, we disable the baseline

injection that originates from the bridges whose endpoint mentor state is s_κ^M. (i) During the rollout, the number of bridges that are formed with s_κ^M as their start point mentor state and whose endpoint is other than $(s_\perp, s_\perp)$ exceeds a threshold N_1. This indicates that the observer has become able to reach mentor states beyond s_κ^M without termination. (ii) After condition (i) is satisfied, the $\widehat{b}^m_{\mathrm{Q},\alpha}(s,a)$ constructed from the bridges whose endpoint mentor state is s_κ^M is used at least N_2 times in critic training. (iii) The average of the most recent N_2 values of L_t for the action baseline in condition (ii) falls below a threshold L_{th}. Conditions (ii) and (iii) serve to confirm the progress of critic training: once a sufficient number of baseline samples has been used and the baseline-induced increment has become small, we judge that the increase induced by that mentor state has been largely reflected in the critic. Once disabled, the baseline injection for that mentor state is never re-enabled, which keeps the procedure monotone and avoids oscillation between the two regimes. In the experiments we adopted $(N_1, N_2) = (50, 50)$ and $(10, 10)$ and $L_{\mathrm{th}} = 10^{-3}$. Each setting was run with 10 seeds. Smaller thresholds N_1 and N_2 disable baseline injection earlier. MB-I2L is recovered as the limit of infinite thresholds, $(N_1, N_2) = (\infty, \infty)$, which never disables baseline injection and which we denote by Default; its results are those of MB-I2L in Table 1, repeated here for comparison. Table 8

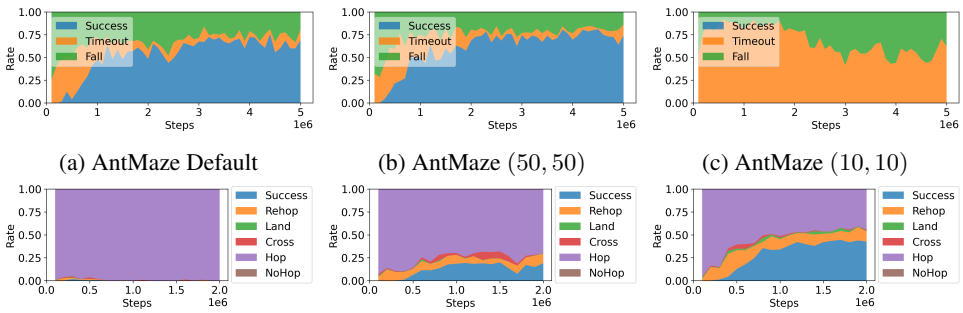
Table 8: Normalized AUC of evaluation success rate of MB-I2L′ (mean ± SD over 10 seeds for every setting) for each threshold setting (N_1, N_2). Default corresponds to $(N_1, N_2) = (\infty, \infty)$, i.e., MB-I2L, and its results are taken from Table 1.

	Default	$(50, 50)$	$(10, 10)$
Point	0.77 ± 0.076	0.76 ± 0.19	0.75 ± 0.097
Ant	0.53 ± 0.054	0.61 ± 0.063	0.00 ± 0.00
Hopper-S	0.63 ± 0.27	0.68 ± 0.28	0.82 ± 0.11
Hopper-L	0.0050 ± 0.016	0.12 ± 0.24	0.29 ± 0.31

shows that a single threshold setting, $(N_1, N_2) = (50, 50)$, performs comparably to or better than MB-I2L in all four environments: the performance is almost unchanged in PointMaze, improves in AntMaze, and changes only slightly in HopperHurdle-Short and HopperHurdle-Long relative to the variability across seeds. The more aggressive setting $(N_1, N_2) = (10, 10)$ yields larger improvements in HopperHurdle-Short and HopperHurdle-Long, but prevents the observer from reaching the goal in AntMaze. These results suggest that excessive optimism can be counteracted during learning, whereas the appropriate degree of disabling differs between environments, reflecting a trade-off between suppressing optimism and retaining the exploration guidance provided by the baseline. The following hypothesis may account for these observations. In AntMaze, the state space is far larger than in HopperHurdle, so similar observer states rarely recur during rollouts. Unless the observer has learned to act along the mentor trajectory from a wide variety of observer states, it is therefore difficult to reproduce the path after baseline injection has been disabled, and the condition for disabling baseline injection has to be strict. Nevertheless, optimistic baseline injection can also occur in AntMaze. For example, a bridge can be completed at an endpoint observer state from which the Ant falls within a few steps; the state baseline at such an endpoint overestimates the observer's value, and baseline injection along the bridge can promote actions leading to such states. We conjecture that disabling baseline injection appropriately lets the value be learned on the basis of the observer's own dynamics, which leads to the improvement with $(N_1, N_2) = (50, 50)$. In HopperHurdle-Short and HopperHurdle-Long, by contrast, the state space is not as large as in AntMaze, so relaxing the condition is less likely to prevent the observer from rejoining the mentor path. Instead, because the shared features do not capture the posture and joint states that determine whether the hurdle can be cleared, baseline injection is more likely to be optimistic, and learning the value on the basis of the observer's own dynamics rather than the baseline has a larger effect on performance. The failure analysis is presented in Appendix G.9.

G.9 Failure Analysis in AntMaze and HopperHurdle-Long

In the experiments in the main text, as Q3, we analyzed the causes of the performance degradation of MB-I2L in PointMaze from the viewpoint of the neighborhood design; in this section, as Q7, we analyze the types of failure of MB-I2L in AntMaze and HopperHurdle-Long. In addition, we



(a) AntMaze Default (b) AntMaze $(50, 50)$ (c) AntMaze $(10, 10)$

(d) HopperHurdle-Long Default (e) HopperHurdle-Long $(50, 50)$ (f) HopperHurdle-Long $(10, 10)$

Figure 9: Failure transitions of MB-I2L and MB-I2L′.

analyze MB-I2L′ in the same way and examine the effect of the thresholds in the condition for disabling baseline injection.

In the AntMaze environment, an episode terminates in one of three ways, namely success, falling over, and timeout, and we measured the proportion of each. In the HopperHurdle-Long environment, by contrast, an episode terminates either in success or in failures such as the Hopper becoming unhealthy or its center of mass entering the hurdle. To classify these failures more transparently, we divided them into five mutually exclusive categories according to how far the observer progressed. Here, a jump refers to the foot leaving the ground, and clearing the hurdle refers to the center of mass passing the hurdle without termination: (i) failure without jumping or clearing the hurdle (NoHop), (ii) failure after jumping but without clearing the hurdle (Hop), (iii) failure after clearing the hurdle but before the foot touches the ground (Cross), (iv) failure after the foot touches the ground following the clearance but without jumping again (Land), which includes the case in which the Hopper clears the hurdle without leaving the ground, and (v) failure after jumping again following such contact (Rehop). We measured the proportion of each. Timeouts are rare in the HopperHurdle environment and did not occur in any of the three settings in HopperHurdle-Long, but in principle a timeout is assigned to one of the above categories according to the stage the observer had reached.

The results are shown in Figure 9. In AntMaze, as training progresses, Success increases and both Timeout and Fall decrease for both MB-I2L (Default) and MB-I2L′ with $(N_1, N_2) = (50, 50)$, and Fall remains the main cause of failure. In contrast, MB-I2L′ with $(N_1, N_2) = (10, 10)$ never reaches the goal. In HopperHurdle-Long, the main cause of failure is Hop, that is, failure after jumping without clearing the hurdle. As Success increases, Hop decreases, whereas the other types of failure show no substantial change.

H Discussion and Limitations

We discuss the main limitations of MB-I2L and the associated design and theoretical trade-offs.

The effectiveness of MB-I2L depends on whether the predefined features and the resulting neighborhood relation provide task-relevant correspondences between the mentor and observer, even when their state spaces differ. Although MB-I2L does not require a large amount of mentor data or mentor actions, its reliance on such designer-specified prior knowledge is a constraint that may limit its applicability. In particular, because this work uses only a single episode of mentor trajectory data, it is generally difficult to learn the mentor–observer correspondence itself from data. Consequently, when the neighborhoods provide appropriate task-level correspondences, MB-I2L can provide a strong exploration signal; however, when the correspondence is too coarse to distinguish states of different values, it can inflate action-value estimates for state-action pairs associated with low-value observer states and thereby hinder exploration, as shown in the failure analysis in the main text. Moreover, in our setting with an extremely limited amount of mentor data, accurately estimating the expected return-to-go from the same or corresponding mentor states is generally difficult when state transitions or rewards are stochastic. MB-I2L uses the empirical return-to-go observed in a single mentor trajectory, which remains available even in such a data-limited setting. This constitutes an important trade-off to consider when applying MB-I2L to stochastic environments. Our results also

suggest that the optimism induced by coarse correspondences can be counteracted during learning. MB-I2L$'$ (Appendix G.8) disables baseline injection for mentor states whose induced increments have already been absorbed by the critic; a single threshold setting, $(N_1, N_2) = (50, 50)$, performed comparably to or better than MB-I2L in all four environments. More aggressive disabling, $(N_1, N_2) = (10, 10)$, yielded larger improvements in HopperHurdle but prevented the observer from reaching the goal in AntMaze, indicating a trade-off between suppressing optimism and retaining the exploration guidance provided by the baseline.

For the theoretical performance bounds, there is a trade-off among interpretability, tightness, and estimability. Theorem 1 upper-bounds the performance loss using a quantity based on the value gap $\Gamma^\star$ between the state baseline b_{V} and the observer's optimal value. This form is interpretable because it directly characterizes how mentor-derived baseline optimism can lead to performance loss; however, $V^{O,\star}$ generally cannot be evaluated directly in an unknown observer MDP. Furthermore, the bound in the theorem is conservative because it counts as a potential source of performance loss even baseline-induced increments that raise the values of actions without changing the original action ordering. Proposition 1 in Appendix C.2 isolates this source of conservativeness by subtracting the cumulative baseline-induced increment attainable by the original optimal policies, yielding a tighter bound. For example, it can reflect the fact that the policy does not degrade when all actions at a given state are increased by the same amount. However, this refinement requires both the baseline-injected action-value function Q_ν^{BI} and the set of optimal policies of the original observer MDP, $\Pi^{O,\star}$, making it even more difficult to evaluate during learning. As a way to relax this dependence on unknown optimal quantities, Corollary 2 in Appendix C.3 shows that a valid bound can also be obtained using a value gap with respect to an arbitrary reference policy $\tilde{\pi}^O$ instead of an optimal policy. However, the resulting state value gap $\Gamma^{\tilde{\pi}^O}$ equals $\Gamma^\star$ plus the suboptimality of the reference policy, so the gain in estimability generally comes at the cost of a more conservative bound. In particular, choosing $\tilde{\pi}^O = \pi_{\mathrm{b}}^O$ makes V^{O,π_{b}^O}, in principle, estimable from observer rollouts without requiring direct knowledge of the optimal value. Thus, our theoretical analysis presents three complementary perspectives: a conservative but interpretable bound that directly relates baseline optimism to performance loss, a tighter refinement that requires unknown optimal quantities, and a reference-policy bound that removes the dependence on the optimal value at the cost of greater conservativeness.