FEEDBACK MANIPULATION REGULARIZATION: ENABLING OFFLINE AGENT ALIGNMENT FOR IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) research has begun to shift focus toward alignment, ensuring agents learn behaviors that adhere to human values. While human demonstrations and feedback have proven crucial for alignment, existing approaches that combine these signals are largely designed for the contextual bandit framing of language generation. Within this framing, recent work has started moving beyond multi-stage pipelines toward fully offline, single-stage pipelines that can reduce computational overhead and remove the need for environment access, broadening alignment's applicability. Yet little work extends such pipelines to sequential decision-making settings, where these complementary inputs can serve as a richer, interconnected signal. We propose Feedback Manipulation Regularization (FMR), an algorithm-agnostic method that harnesses evaluative feedback as a corrective signal to improve the alignment of imitation learning policies. We adapt Safety Gymnasium environments as a principled testbed for alignment evaluation, on which FMR reduces misalignment by up to 98% across a range of imitation learning algorithms while often improving aptitude. FMR remains robust in limited-data regimes, even when demonstrations become predominantly noisy and uninformative.

1 INTRODUCTION

Reinforcement learning (RL) has traditionally faced two major challenges: aptitude, an agent's ability to learn and master a task, and alignment, ensuring it does so in ways that adhere to human values, intentions, and preferences (Ji et al., 2025). While early RL focused on aptitude, alignment has become a growing concern, one expected to intensify as agents become more capable (Leike et al., 2018; Amodei et al., 2016; Dulac-Arnold et al., 2021). Interactive RL (Int-RL) attempts to address both challenges by integrating human input directly into the RL framework (Li et al., 2019; Arzate Cruz & Igarashi, 2020; Najar & Chetouani, 2021). Among these inputs, demonstrations and feedback are particularly intuitive mediums for communication: demonstrations explicitly depict desired behavior, while feedback (e.g., preferences) refines it by indicating what is good, bad, or preferred (Zare et al., 2024; Najar & Chetouani, 2021; Christiano et al., 2017; MacGlashan et al., 2017). Both mediums are now widely used to fine-tune and align large language models through Learning from Preferences (LfP) and Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022). However, these approaches typically rely on sequential, multi-stage pipelines that can require environment access or an additional supervised policy (Ouyang et al., 2022; Liu et al., 2025; Rafailov et al., 2023).

Only recently have works begun combining both mediums in a fully offline, single-stage pipeline (Meng et al., 2024; Hong et al., 2024; Hejna et al., 2024; Hejna & Sadigh, 2023). Despite this progress, most of these methods are designed within the contextual bandit framing and do not readily translate to traditional sequential decision-making settings, with only a few extending to such settings (Hejna et al., 2024; Hejna & Sadigh, 2023). Moreover, in terms of alignment, preference-based feedback carries two notable limitations in sequential settings. First, preferences are inherently relative judgments rather than absolute declarations of alignment. That is, a preferred behavior is not necessarily an aligned one. Second, randomly sampled comparison pairs compound this ambiguity,

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.

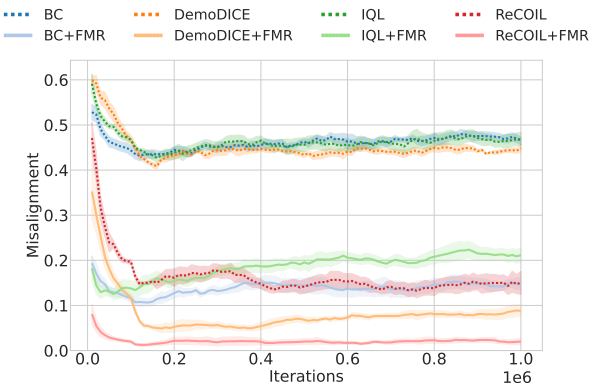


Figure 1: Imitation learning can suffer from high misalignment. Our proposed method, FMR (solid lines), greatly reduces misalignment for each corresponding baseline (dotted lines). Results depict misalignment learning curves for the PathM navigation policy using a 10-50 data ratio. The shaded region represents the standard error over 5 seeds.

making it difficult to target specific behaviors without explicit curation of aligned and misaligned pairs. Therefore, preferences are ill-suited for alignment when behavioral precision is essential.

To address the limitations of preference-based approaches, we turn to evaluative feedback. Popularized by online interactive methods such as TAMER and COACH (Knox & Stone, 2009; MacGlashan et al., 2017), evaluative feedback operates at the state-action level, asking "is this behavior good or bad?" rather than "which of the two is better?". Evaluative feedback removes the relational ambiguity of preferences and enables direct targeting of specific behaviors. Accordingly, we reframe how demonstrations are incorporated through imitation learning, pairing them with evaluative feedback so that aligned sub-trajectories are encouraged and misaligned behaviors are suppressed. This framing can be further extended to Learning from Noisy Demonstrations (LfND), where demonstrations of high alignment quality are treated as expert data and demonstrations containing noisy mixtures of aligned and misaligned behaviors are treated as imperfect data. The result is an offline, single-stage pipeline that combines demonstrations and evaluative feedback for sequential decision-making environments.

Such a pipeline, while challenging, would broaden alignment's applicability through two key benefits: a single-stage process eliminates the computational overhead of multiple training phases, while offline learning removes the need for environment access. Moreover, evaluative feedback offers a less ambiguous alternative to preferences while likewise not requiring knowledge of the correct action. To this end, we introduce Feedback Manipulation Regularization (FMR), an algorithm-agnostic method that combines imitation learning with evaluative feedback in an offline, single-stage pipeline. FMR leverages demonstrations to learn the state space and uses evaluative feedback to directly manipulate the policy toward human-aligned behavior via temperature scaling, a mechanism independent of the underlying imitation learning algorithm. We adapt Safety Gymnasium (Ji et al., 2023) environments to provide a principled testbed for alignment evaluation, using cost signals to measure deviation from aligned behavior. Our experiments demonstrate that FMR improves alignment across a diverse set of imitation learning algorithms and aligned-to-imperfect data ratios (Figure 1), and generally outperforms alternative approaches that treat evaluative feedback as a reward signal or a preference score. Code is available at `https://anonymous.4open.science/r/FMR-E27C`.

2 Related Work

Learning from Noisy Demonstrations Common imitation learning methods include Behavioral Cloning (BC), which maps observations to actions via supervised learning, and IQ-Learn (IQL), which frames imitation as inverse soft-RL without explicit reward models (Garg et al., 2021). Meanwhile, Learning from Noisy Demonstrations (LfND) addresses the realistic mixed-quality demonstration setting by learning from distinct expert and imperfect datasets (Zare et al., 2024). Notable LfND methods include DemoDICE, which matches occupancy to expert data while regularizing against

suboptimal data (Kim et al., 2022), and ReCOIL, which relaxes occupancy matching to enable imitation from arbitrary off-policy data (Sikchi et al., 2024). Despite outperforming standard imitation learning baselines, LfND methods frequently yield misaligned policies even when expert demonstrations consist entirely of aligned behaviors, a limitation that worsens as the expert-to-imperfect demonstration ratio decreases.

Interactive Reinforcement Learning Int-RL integrates human input directly into RL, typically online, in the form of advice (e.g., feedback) or demonstrations (Najar & Chetouani, 2021; Zare et al., 2024). Classical online interactive methods such as TAMER (Knox & Stone, 2009) and COACH (MacGlashan et al., 2017) apply evaluative feedback at the state-action level to label behaviors as "good" or "bad", with credit assignment enabling targeted sub-trajectory labeling. In contrast, Learning from Preferences (LfP) methods use comparative feedback on trajectory pairs to infer a reward function or policy directly (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2023). While works such as (Hejna & Sadigh, 2023) and (Hejna et al., 2024) learn policies directly from offline preferences, this feedback is fundamentally relative rather than absolute. This means a preferred behavior is not necessarily an aligned one. When both behaviors in a comparison pair are misaligned, the less misaligned one may still be preferred. Furthermore, randomly sampled comparison pairs limit the ability to target specific aligned or misaligned behaviors, meaning preference feedback provides only a coarse, relative signal ill-suited for the behavioral precision required for alignment in sequential decision-making settings.

Alignment Evaluation RL evaluation has historically focused on total return, leaving the open question of how to properly quantify agent alignment. Return-based alignment metrics (Liu et al., 2025) fail when misaligned behaviors achieve similar returns, while reward model evaluations do not generalize to methods that forgo modeling reward functions (Malik et al., 2025). Meanwhile, Safe RL has long employed Constrained Markov Decision Processes, where environment-elicited costs quantify constraint violations for evaluating the "safety" of behaviors (García et al., 2015; Ji et al., 2023). Generalizing this idea, we use cost as a natural evaluation metric for alignment, rather than a training objective, to directly detect misaligned behavior.

Temperature Scaling Temperature scaling divides logits by a temperature $\tau > 0$ before softmax, producing class probabilities $P_i = \frac{e^{z_i/\tau}}{\sum_j e^{z_j/\tau}}$, and has seen wide adoption in knowledge distillation (Hinton et al., 2015), calibration (Guo et al., 2017; Xie et al., 2024), and contrastive learning (Wang & Liu, 2021). Two extensions are particularly relevant: adaptive temperature scaling predicts a per-input temperature for flexible adjustment (Xie et al., 2024), while vector temperature scaling assigns a distinct temperature to each class (Guo et al., 2017). FMR takes inspiration from both, using feedback to selectively redistribute probability mass toward more aligned behaviors, but does so in probability space rather than in logit space.

3 PRELIMINARIES

To account for human feedback, we model the environment as a modified Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, h, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability function $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, human feedback function $h : \mathcal{S} \times \mathcal{A} \to \{-1, 0, 1\}$, and discount factor $\gamma \in [0, 1]$. The goal of imitation learning and RL is to learn a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ mapping states to distributions over actions. In RL, this is done by maximizing the return, $G = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$, discounting the future by γ. In imitation learning, r is unknown, which means it is often estimated, explicitly or implicitly. We define $D^U = D^E \cup D^I$ as the union of the expert (i.e., aligned) and imperfect datasets containing tuples without reward $\mathcal{M} \backslash \{r\}$.

Classical approaches to imitation learning such as BC aim to learn a policy π that maps state s to action a via supervised learning by minimizing the negative log-likelihood:

$$\min_{\pi} \mathcal{L}_{\text{BC}}(\pi) = \min_{\pi} -\frac{1}{|D^U|} \sum_{(s,a) \in D^U} \log \pi(a \mid s). \tag{1}$$

Policy learning methods in LfND build upon this core idea to improve robustness against noise in D^I (Kim et al., 2022; Sikchi et al., 2024).

Additionally, many imitation works aim to estimate the value of a state, where $V_\pi : \mathcal{S} \to \mathbb{R}$ denotes the state value function of π. The state value function estimates the expected return $V_\pi(s) = \mathbb{E}_\pi\big[G \mid s_0 = s\big]$ when starting at s and thereafter following π. Likewise, $Q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the state-action value function of π, where $Q_\pi(s, a) = \mathbb{E}_\pi\big[G \mid s_0 = s, a_0 = a\big]$. Although traditional value function estimation relies on rewards r, imitation learning algorithms have developed alternative techniques for estimating value functions without explicit reward signals (Kim et al., 2022; Sikchi et al., 2024; Garg et al., 2021).

4 Feedback Manipulation Regularization

In this section, we introduce our proposed method of feedback manipulation regularization (FMR) for harnessing both demonstrations and feedback as an interconnected source of information. The goal of FMR is to enable human manipulation of an agent's policy through evaluative feedback h on demonstrations. Let D^U denote an imbalanced dataset of demonstrations composed of aligned and imperfect trajectories where typically $|D^E| \ll |D^I|$. We make the following assumption about the learned policy π_θ and latent human-aligned policy π_h.

Assumption 4.1 (Coverage)**.** For any $s \in \mathcal{S}$, if $\pi_h(a \mid s) > 0$ then $\pi_\theta(a \mid s) > 0$.

Since π_h is unknown, we approximate it by reshaping π_θ with feedback, which acts as a corrective signal that redistributes probability mass toward the underlying aligned policy. To achieve this, we take inspiration from temperature scaling for rescaling distributions.

Definition 4.2 (Temperature Scaling for Human-Aligned Policy Approximation)**.** Given $\pi_\theta(\cdot \mid s)$ and a *per-action* temperature $\tau(s, a_j) > 0$ for each $a_j \in \mathcal{A}$ (for simplicity, denoted by τ_j hereafter), define the *approximated human-aligned policy* as

$$\pi_\tau(a_j \mid s) \;:=\; \frac{\pi_\theta(a_j \mid s)\, \tau_j^{-1}}{Z_\theta(s)},$$

where $Z_\theta(s) = \sum_{a_k \in \mathcal{A}} \pi_\theta(a_k \mid s)\tau_k^{-1}$.

By operating directly in probability space, Definition 4.2 requires no assumptions about how π_θ is computed, while preserving the temperature scaling intuition: actions with higher temperatures lose probability mass, and those with lower temperatures gain it.

Definition 4.3 (Feedback and Temperature)**.** Human feedback h on the selected action (i.e., the demonstrated action at s) induces the temperature τ_j via a strictly decreasing map. Hence, positive h implies $\tau_j < 1$ (*encourage*), and negative h implies $\tau_j > 1$ (*discourage*).

Remark 4.4 (Feedback-Adaptive Temperature). We propose a simple exponential form for mapping feedback h to temperature τ_j:

$$\tau_j = \begin{cases} \beta^{-h} & \text{if } h < 0 \text{ and } j = \text{selected action}, \\ \beta^{h} & \text{if } h > 0 \text{ and } j \neq \text{selected action}, \\ 1 & \text{otherwise}, \end{cases} \tag{2}$$

where $\beta > 1$ is a hyperparameter. In a discrete action space, if negative feedback is received, the selected action receives $\tau_j > 1$. If positive feedback is received, all actions other than the selected action receive $\tau_j > 1$, which, since π_τ is normalized, is equivalent to assigning $\tau_j < 1$ to the selected action.[1] When no feedback is received, $\tau_j = 1$, which has no impact on learning. Although each raw feedback signal lies in $\{-1, 0, 1\}$, credit assignment (Knox & Stone, 2009) can map a signal to multiple prior states, where overlapping credited signals are summed. Repeated feedback given in quick succession therefore accumulates to a larger magnitude, yielding a greater redistribution of probability mass.

By systematically discouraging misaligned behaviors through this redistribution of probability mass, aligned demonstrations become more probable even without explicit positive reinforcement. With π_τ defined as a feedback-tempered surrogate for π_h, we formulate policy alignment as a distribution matching problem via reverse KL divergence, a common approach in imitation learning (Kim et al., 2022; Sikchi et al., 2024).

[1] This avoids the negative loss that assigning $\tau_j < 1$ to the selected action would produce due to the logarithm.

Definition 4.5 (Reverse-KL Formulation)**.** We align π_θ to π_τ, the feedback-tempered surrogate for π_h, by minimizing the reverse KL divergence

$$\mathcal{J}(\theta;\tau) \;:=\; \mathbb{E}_{s\sim D^U}\Big[D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s)\,\|\,\pi_\tau(\cdot \mid s)\big)\Big].$$

Proposition 4.6 (Reverse-KL Decomposition)**.** *For any fixed s,*

$$D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s)\,\|\,\pi_\tau(\cdot \mid s)\big) = \mathbb{E}_{a_j\sim\pi_\theta(\cdot|s)}\big[\log \tau(s,a_j)\big] + \log Z_\theta(s).$$

Proof. From Definition 4.2, $\log \pi_\tau(a_j \mid s) = \log \pi_\theta(a_j \mid s) - \log \tau_j - \log Z_\theta(s)$. Substituting into the KL divergence and noting that the $\log \pi_\theta(a_j \mid s)$ terms cancel,

$$\sum_j \pi_\theta(a_j \mid s) \log \frac{\pi_\theta(a_j \mid s)}{\pi_\tau(a_j \mid s)} = \sum_j \pi_\theta(a_j \mid s) \log \tau_j + \log Z_\theta(s). \qquad \square$$

Corollary 4.7 (Generalized KL)**.** *Let the unnormalized target measure be $\tilde{\pi}_\tau(a \mid s) \propto \pi_\theta(a \mid s)\,\tau(s,a)^{-1}$. Then*

$$D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s)\,\|\,\tilde{\pi}_\tau(\cdot \mid s)\big) \;=\; \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\big[\log \tau(s,a)\big] + \mathrm{const}.$$

Remark 4.8 (Connection to Entropy Regularization). Setting $\tau(s,a) = \pi_\theta(a \mid s)$ recovers entropy regularization (Williams & Peng, 1991; Mnih et al., 2016). More generally, since τ is derived from h, feedback modulates the entropy of π_θ, where negative feedback increases entropy while positive feedback decreases it.

Definition 4.9 (Feedback Manipulation Regularization)**.** Given τ induced by feedback via Definition 4.3, the surrogate regularizer is

$$\mathcal{R}_{\mathrm{FMR}}(\theta) := \mathbb{E}_{s\sim D^U}\,\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\big[\log \tau(s,a)\big].$$

Any imitation learning objective can incorporate FMR as

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{IL}}(\theta) \;+\; \alpha\,\mathcal{R}_{\mathrm{FMR}}(\theta), \tag{3}$$

where $\mathcal{L}_{\mathrm{IL}}$ is the imitation learning loss and $\alpha > 0$ controls alignment strength.

Together, Definitions 4.2 and 4.3 specify how feedback systematically reshapes π_θ, while Definition 4.9 provides an algorithm-agnostic regularization term that enforces this alignment under any imitation learning objective.

5 Experiments

This section aims to address the following questions: (1) Can FMR enhance the alignment performance of imitation learning algorithms? (2) How does FMR compare to alternative uses of evaluative feedback, such as treating it as reward or as a preference score? We evaluate all methods across aligned-to-imperfect demonstration data ratios of 10-50 (1:5), 25-50 (1:2), and 50-50 (1:1) to test algorithmic robustness as aligned data D^E becomes increasingly scarce. Additional details regarding the tasks, aligned policies, and data collection, along with visualizations, are provided in Appendix B. Results for the velocity tasks, additional plots for navigation results, and extensive additional experiments are provided in Appendix C.

5.1 Experimental Setup

Environments We adapt environments from Safety Gymnasium (Ji et al., 2023), built on Gymnasium (Towers et al., 2025) and MuJoCo (Todorov et al., 2012), using cost as a principled measure of misalignment exclusively for evaluation. We evaluate on a 3D navigation task and three velocity-restricted locomotion tasks (Hopper, Swimmer, Walker2D), each with a discretized action space suited to human demonstration collection (or human-like proxies). We modify the original navigation task so that an agent (red sphere) randomly spawns in the upper-right region and must reach a fixed goal (green cube) using lidar observations. The navigation task includes two unobservable hazard types for measuring alignment: passable floor hazards and blocking hazard walls, each incurring a cost of 1 per violation. For velocity tasks, a cost of 1 is incurred per step exceeding a task-specific x-velocity threshold, inducing velocity-restricted gaits as aligned policies.

Aligned Policies We define two aligned policies for the navigation task. PathM requires the agent to move diagonally through the center via a narrow hazard-free corridor, while PathBB requires the agent to move along the right boundary, then the bottom boundary, entering the goal from below. Hazards are placed to highlight each aligned path and measure deviation, with hazard walls positioned around the goal to enforce directional entry for PathBB. For each aligned policy, D^E comprises aligned human demonstrations that follow the aligned policy's designated path. None of the demonstrations in D^I reach the goal and instead circle it to obfuscate the objective. Critically, PathM exhibits high overlap between D^E and D^I, as imperfect demonstrations frequently traverse the center corridor, while PathBB exhibits low overlap, as imperfect demonstrations rarely follow the right-and-bottom boundary path. High overlap tests whether aligned sub-trajectories can be identified within otherwise misaligned demonstrations, while low overlap tests generalization to underrepresented aligned behavior, a harder setting since D^I contains fewer relevant sub-trajectories.

For the velocity tasks, we define one aligned policy per environment by imposing an x-velocity threshold, producing a distinctive gait that contrasts with the optimal high-speed policy. SlowHop ($v \leq 0.74$) performs slow forward hops with brief pauses, SlowSwim ($v \leq 0.75$) shortens the peaks of the optimal sinusoidal gait into a slower serpentine motion, and SlowWalk ($v \leq 1.50$) produces a measured bipedal walk in contrast to the optimal running gait. For each velocity task, D^E comprises aligned demonstrations that follow the aligned policy's velocity-restricted gait. Swimmer data has high overlap, as D^E contains a sinusoidal gait similar to the optimal trajectories in D^I, but with reduced amplitude. Hopper, by contrast, has low overlap, as D^I predominantly contains demonstrations exceeding the velocity threshold. Walker2D falls between the two with medium overlap, but its main difficulty lies in its large action space ($|\mathcal{A}| = 3^6$).

Data and Feedback Collection For navigation data, D^E comprises 50 aligned human demonstrations collected per aligned policy, and D^I consists of 50 imperfect human demonstrations shared across policies. For velocity data, human proxy demonstrations are collected, where D^E comprises 50 aligned demonstrations from a cost-augmented SAC-Discrete agent (Haarnoja et al., 2018; Christodoulou, 2019), and D^I comprises 50 demonstrations sampled across standard SAC-Discrete training, yielding variation in quality and dataset overlap. Navigation feedback is collected from a real human evaluator on D^I via a visual replay interface where a 600 ms credit assignment window is used to map feedback to prior states (Knox & Stone, 2009). For velocity data, proxy feedback is generated by converting cost violations into negative feedback only. Swimmer and Walker2D use dense feedback per violation, while Hopper uses sparser feedback given at violation onset and for 50 subsequent steps. Walker2D also receives negative feedback for the 15 steps preceding failure.

Baselines To assess the effectiveness of FMR in enhancing the alignment of baseline imitation learning methods trained on D^U, we evaluate both traditional imitation learning and robust LfND algorithms, namely BC, IQL (Garg et al., 2021), DemoDICE (Kim et al., 2022), and ReCOIL (Sikchi et al., 2024), with and without FMR augmentation. These represent diverse algorithmic paradigms: actor-critic (DemoDICE, ReCOIL), soft Q-learning (IQL), and supervised learning (BC). Notably, BC and IQL are not designed for imperfect demonstrations, making LfND challenging for them, while DemoDICE and ReCOIL are explicitly designed for this setting. To examine alternative approaches to harnessing evaluative feedback, we use Dual Value Learning (DVL) (Sikchi et al., 2024), an offline RL variant of ReCOIL, where we substitute evaluative feedback for reward. Likewise, we employ Contrastive Preference Learning (CPL) (Hejna et al., 2024), an offline, single-stage LfP algorithm, in which evaluative feedback serves as the comparison score for determining preference. This is achieved by calculating the total feedback accumulated per trajectory segment.

Metrics To assess the aptitude performance for the navigation policies, we report *success rate*, defined as the ratio of episodes that reach the goal to total evaluation episodes. For the velocity policies, we report *total normalized return* where return is normalized based on the performance of D^E (Appendix B.2). For all tasks, alignment performance is measured using *misalignment*, defined as the ratio between the number of steps that incur a cost c_t and the total number of steps in the episode T as $\frac{1}{T}\sum_{t=1}^{T} c_t$.

Implementation Details All algorithms are trained for 1M gradient steps with evaluations every 10K steps (50 episodes each). To evaluate the final average performance for each metric, we compute the mean over the last 10 evaluations across all 5 random seeds. When decreasing the ratio between

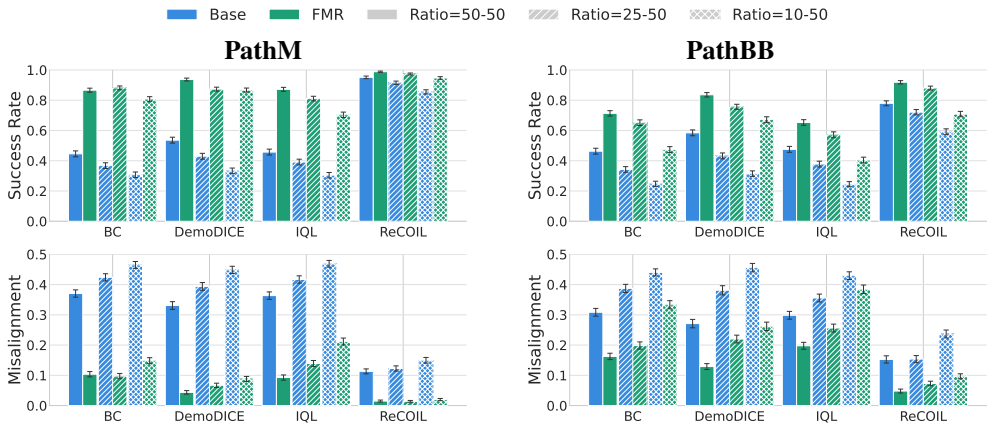


Figure 2: Navigation results across all baseline algorithms and data ratios. Results show mean success rate and misalignment scores with 95% confidence intervals, averaged over the last 10 evaluations across 5 seeds.

D^E and D^I, we find oversampling D^E helps to mitigate loss in performance due to dataset imbalance across all algorithms. For the navigation data, we remove duplicate states occurring at episode initialization. For FMR, we use $\beta = 10$ and $\alpha = 1$ for all algorithms unless stated otherwise. Additionally, we find that DemoDICE+FMR, ReCOIL+FMR, and DVL benefit from cosine learning rate decay when oversampling, while the other algorithms show no noticeable benefit. For additional implementation details, see Appendix A.

5.2 Can FMR Enhance Alignment Performance?

Figure 2 depicts the aptitude (i.e., success rate) and alignment (i.e., misalignment) performance results for the navigation policies PathM and PathBB across all data ratios. PathM exhibits higher baseline performance than PathBB, likely due to higher overlap between D^E and D^I. For both policies, as the data ratio decreases, we observe large drops in success rate and corresponding increases in misalignment across most baselines. The notable exception is ReCOIL, which demonstrates the strongest baseline performance, particularly on PathM, though still at the cost of higher misalignment. FMR yields clear improvements across all baselines, consistently improving success rates and reducing misalignment, even as aligned demonstrations are reduced. ReCOIL+FMR achieves the strongest overall performance, combining ReCOIL's higher success rates with drastically reduced misalignment, achieving up to a 92% reduction on PathM and 67% on PathBB.

Among the velocity policies, SlowSwim (high-overlap data) achieves the strongest baseline performance, while SlowHop and SlowWalk prove considerably more challenging (see Appendix C.2). For SlowSwim, ReCOIL achieves a return close to D^E with low misalignment, while other baselines reach comparable returns but with considerably higher misalignment that grows as the data ratio decreases. In contrast, all FMR variants remain stable across data ratios, keeping returns near D^E while cutting misalignment by up to 98%, with ReCOIL+FMR achieving near-zero misalignment. SlowHop baselines exhibit more varied performance, with BC and IQL substantially exceeding D^E performance and producing larger misalignment scores, while misalignment worsens as the data ratio decreases across all baselines. FMR variants reduce misalignment by up to 75% across all ratios and algorithms, with ReCOIL+FMR maintaining the best alignment.

SlowWalk follows a similar but more pronounced degradation trend, with baseline misalignment remaining persistently high and returns declining at lower data ratios. For this task, adding positive feedback to all state-action pairs in D^E and increasing α to 2 improves FMR performance, which we attribute to the challenges posed by the large action space. FMR variants consistently reduce misalignment across nearly all algorithms and ratios, with ReCOIL+FMR achieving the best trade-off between return and alignment, reducing misalignment by up to 76% overall. DemoDICE+FMR at the 10-50 ratio is a notable outlier showing little reduction, likely due to the large action space and potentially addressable through better hyperparameter tuning of the base algorithm.

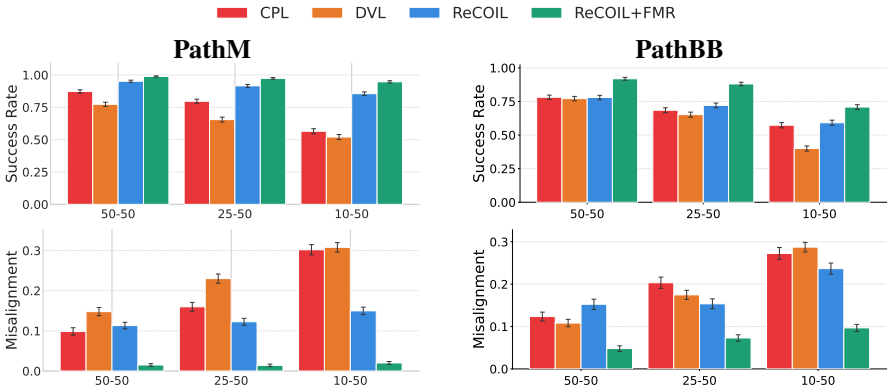


Figure 3: Navigation results comparing ReCOIL+FMR with DVL, CPL, and ReCOIL across data ratios. Results show mean success rate and misalignment scores with 95% confidence intervals, averaged over the last 10 evaluations across 5 seeds.

5.3 How Does FMR Compare to Alternative Uses of Evaluative Feedback?

To evaluate FMR's effectiveness at harnessing evaluative feedback, we adapt two popular alternative methods for comparison. Specifically, we compare FMR against DVL (Sikchi et al., 2024), the offline RL version of ReCOIL adapted by replacing its reward signal with evaluative feedback, and CPL (Hejna et al., 2024), an offline, single-stage LfP algorithm adapted by summing evaluative feedback over trajectory segments to derive preference scores between pairs. For both DVL and CPL, in addition to the evaluative feedback for D^I, we add positive feedback for all state-action pairs in D^E. Without this positive feedback, both algorithms can exhibit performance degradation.

Results for PathM and PathBB are presented in Figure 3. On PathM, ReCOIL+FMR outperforms all alternatives across all data ratios, achieving the highest success rates with substantially lower misalignment. While CPL and DVL achieve moderate success rates at higher data ratios, both degrade considerably as the ratio decreases, with misalignment rising sharply at the 10-50 ratio. PathBB follows a similar trend, with all methods showing degraded performance at lower data ratios. ReCOIL+FMR again achieves the strongest overall performance, maintaining the highest success rates and lowest misalignment across all ratios. CPL performs comparably to ReCOIL at higher ratios but degrades more sharply as the data ratio decreases, while DVL underperforms both, particularly at the 10-50 ratio.

Across the velocity policies, DVL exhibits high misalignment despite achieving returns comparable to D^E, worsening as the data ratio decreases. On SlowWalk, DVL achieves lower misalignment than ReCOIL and CPL, though still higher than ReCOIL+FMR. CPL performs comparably to ReCOIL+FMR on SlowSwim and SlowHop, achieving competitive returns with low misalignment, but degrades substantially on SlowWalk, likely due to the higher-dimensional action space. ReCOIL+FMR demonstrates the most robust performance across all tasks and data ratios, consistently achieving the best trade-off between return and misalignment.

Overall, these results demonstrate that FMR more effectively harnesses evaluative feedback than DVL or CPL. In particular, DVL fails to effectively utilize feedback, consistent with prior work indicating that naively interpreting feedback as reward is ineffective (MacGlashan et al., 2017; Thomaz & Breazeal, 2006). Moreover, CPL falls short even when using evaluative feedback to alleviate the limitations of typical preference feedback.

6 Discussion

Only Negative Feedback Results for the velocity policies show that FMR can learn effectively from negative feedback alone on D^I. Removing positive feedback from D^I (Appendix C.4) yields negligible changes in PathM and PathBB performance, indicating that negative feedback is the primary driver of FMR's advantage. We hypothesize that negative feedback is most valuable at deviation

points, where imperfect trajectories diverge from the aligned policy, as supported by SlowHop, which receives feedback only near such points. Even so, SlowWalk results suggest that positive feedback on fully aligned demonstrations can be valuable, possibly more so in high-dimensional action spaces.

Impact of Imperfect Demonstrations Performance gaps between high- and low-overlap policies reveal a fundamental challenge: when demonstrations have lower overlap with the target behavior, most data becomes irrelevant to learning the aligned policy. While some sub-trajectories may still be beneficial, the limited amount of aligned data constrains what feedback can accomplish. With high overlap, ReCOIL+FMR trained on D^U at a 10-50 data ratio performs similarly to or outperforms BC and IQL trained exclusively on D^E (Appendix C.5), showing that feedback can turn aligned sub-trajectories within otherwise misaligned demonstrations into constructive learning signals. With low overlap, however, training on D^U falls short of expert-only training. Even so, FMR narrows this gap relative to its base algorithm, mitigating, though not eliminating, the cost of irrelevant data.

Feedback Scalability To assess feedback scalability, we randomly remove all feedback from a subset of D^I trajectories under the 50-50 data ratio (Appendix C.6). For navigation, ReCOIL+FMR degrades gradually as feedback coverage decreases, with only slight loss at 50% coverage and performance still marginally above baseline ReCOIL at 20%. Random selection likely amplifies this decay, as dropping feedback on underrepresented or misaligned trajectories risks reinforcing poor behaviors, motivating targeted feedback at deviation points. Among velocity policies, only SlowHop is affected, likely because nearly all D^I trajectories share the same initial path, requiring substantial feedback to override this behavioral prior. A promising direction for future work is generalizing feedback across similar state-action pairs to reduce the required feedback volume (Du et al., 2023).

Generalized Feedback-Adaptive Temperature All evaluated environments use discrete action spaces to support human and human-like demonstration collection. To extend FMR to continuous action spaces, we generalize Eq. (2) to $\tau^{\text{gen}}(s, a) = \beta^{-h}$, which adjusts only the selected action and avoids enumerating the action space. For positive feedback, this yields $\tau^{\text{gen}}(s, a) < 1$ and a negative loss that can reduce interpretability. Preliminary discrete-action experiments show negligible performance differences between the two forms (Appendix C.7). Combined with FMR's probability-space design, which is agnostic to the policy parameterization, this enables straightforward extension to continuous actions.

7 Conclusion

We propose Feedback Manipulation Regularization (FMR), a method that leverages the rich, interconnected information contained in demonstration feedback to improve alignment and, in many cases, aptitude performance in an offline, single-stage, broadening alignment's applicability within sequential decision-making settings.. FMR is compatible with any imitation learning algorithm, functioning analogously to human-controlled entropy regularization. Specifically, feedback modulates the entropy of the policy, where negative feedback increases entropy to discourage undesirable actions and positive feedback decreases entropy to reinforce preferred behaviors. Our experiments show that FMR consistently enhances a variety of imitation learning baselines in limited-data regimes, yielding policies that better align with human intentions.

AI Use Statement

In this work, we used generative AI tools for paper editing (e.g., rewording, grammar, and syntax), code-base documentation, and coding. For example, we used generative AI to help construct minor utility functions and create class templates. We have not used generative AI tools for major tasks such as dataset generation, mathematical theory, or method formulation. We have thoroughly reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Reproducibility Statement

For reproducibility of all experiments, we have included the following. First, we have included the core source code with the anonymized repository link located in Section 1. Second, we have provided descriptions of the tasks and data collection procedures in Section 5.1, with extended details in Appendix B. Third, we have provided the hyperparameters used for all algorithms in Section 5.1 and Appendix A. Finally, we plan to release the full code and data upon publication.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete Problems in AI Safety, July 2016. URL http://arxiv.org/abs/1606.06565.

Christian Arzate Cruz and Takeo Igarashi. A Survey on Interactive Reinforcement Learning: Design Principles and Open Challenges. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, pp. 1195–1209. Association for Computing Machinery, July 2020. ISBN 978-1-4503-6974-9. doi: 10.1145/3357236.3395525.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Petros Christodoulou. Soft Actor-Critic for Discrete Action Settings, 2019. URL http://arxiv.org/abs/1910.07207.

Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior Retrieval: Few-Shot Imitation Learning by Querying Unlabeled Datasets. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation, July 2023. ISBN 978-0-9923747-9-2.

Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, September 2021. ISSN 1573-0565.

Javier García, Fern, and O Fernández. A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. ISSN 1533-7928.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-Learn: Inverse soft-Q Learning for Imitation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4028–4039. Curran Associates, Inc., 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017. ISSN: 2640-3498.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, 10–15 Jul 2018.

Joey Hejna and Dorsa Sadigh. Inverse preference learning: Preference-based rl without a reward function. In *Advances in Neural Information Processing Systems*, volume 36, pp. 18806–18827. Curran Associates, Inc., 2023.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W. Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: Learning from human feedback without reinforcement learning. In *International Conference on Learning Representations*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Conference on Empirical Methods in Natural Language Processing*, pp. 11170–11189. Association for Computational Linguistics, November 2024.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems*, volume 36, pp. 18964–18993. Curran Associates, Inc., 2023.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Jiayi Zhou, Borong Zhang, Donghai Hong, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Lukas Vierling, Zhaowei Zhang, Fanzhi Zeng, Juntao Dai, Xuehai Pan, Hua Xu, Aidan O'Gara, Kwan Ng, Brian Tse, Jie Fu, Stephen Mcaleer, Yanfeng Wang, Mingchuan Yang, Yunhuai Liu, Yizhou Wang, Song-Chun Zhu, Yike Guo, Yaodong Yang, and Wen Gao. AI Alignment: A Contemporary Survey. *ACM Computing Survey*, 58(5):132:1–132:38, November 2025. ISSN 0360-0300. doi: 10.1145/3770749.

Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022.

W. Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: the TAMER framework. In *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16. Association for Computing Machinery, September 2009. ISBN 978-1-60558-658-8. doi: 10.1145/1597735.1597738.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction, 2018. URL https://arxiv.org/abs/1811.07871.

Guangliang Li, Randy Gomez, Keisuke Nakamura, and Bo He. Human-Centered Reinforcement Learning: A Survey. *IEEE Transactions on Human-Machine Systems*, 49(4):337–349, August 2019. ISSN 2168-2305. doi: 10.1109/THMS.2019.2912447.

Xu Liu, Haobo Fu, Stefano V. Albrecht, QIANG FU, and Shuai Li. Online-to-offline rl for agent alignment. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Representation Learning*, volume 2025, pp. 56116–56136, 2025.

James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L. Roberts, Matthew E. Taylor, and Michael L. Littman. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*, volume 70, pp. 2285–2294. JMLR.org, 2017.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation, 2025. URL https://arxiv.org/abs/2506.01937.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, 2024.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.

Anis Najar and Mohamed Chetouani. Reinforcement Learning With Human Advice: A Survey. *Frontiers in Robotics and AI*, 8:74, 2021. ISSN 2296-9144.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023.

Harshit Sikchi, Qinqing Zheng, Amy Zhang, and Scott Niekum. Dual RL: Unification and new methods for reinforcement and imitation learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Andrea L. Thomaz and Cynthia Breazeal. Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st national conference on Artificial intelligence*, pp. 1000–1005. AAAI Press, July 2006. ISBN 978-1-57735-281-5.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, October 2012. doi: 10.1109/IROS.2012.6386109. ISSN: 2153-0866.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments, 2025. URL https://arxiv.org/abs/2407.17032.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2495–2504, 2021.

Ronald J. Williams and Jing Peng. Function Optimization using Connectionist Reinforcement Learning Algorithms. *Connection Science*, 3(3):241–268, January 1991. ISSN 0954-0091. doi: 10.1080/09540099108946587. URL https://doi.org/10.1080/09540099108946587.

Johnathan Xie, Annie S Chen, Yoonho Lee, Eric Mitchell, and Chelsea Finn. Calibrating language models with adaptive temperature scaling. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 18128–18138. Association for Computational Linguistics, November 2024.

Maryam Zare, Parham M. Kebria, Abbas Khosravi, and Saeid Nahavandi. A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges. *IEEE Transactions on Cybernetics*, 54(12):7173–7186, December 2024. ISSN 2168-2275.

A Implementation Details

A.1 Data Preprocessing

For all tasks, we oversample D^E as the number of demonstrations decreases. Oversampling is performed so that the number of samples in D^E is close to, or slightly lower than, that in D^I. Oversampling is done by repeating entire trajectories, and we typically oversample D^E to the size of D^I (i.e., 50 demonstrations). The exceptions are SlowHop and SlowWalk, where the oversample target is 40 demonstrations to prevent D^E from exceeding the number of samples in D^I. Additionally, for the navigation task, we remove duplicate starting states. These duplicates arise from a delay in the human beginning their demonstration, which creates repeated states within the first 120 steps.

A.2 Algorithms

All algorithm networks are implemented in PyTorch using two-layer neural networks with 256 hidden units and ReLU activation. During training, algorithms are trained for 1M gradient steps (i.e., training iterations) using a learning rate of 3×10^{-4} and a batch size of 128. Evaluations are conducted every 10k steps and consist of 50 episodes each. To evaluate the final average performance for each metric, we compute the mean over the last 10 evaluations from training, across all 5 random seeds. Algorithm hyperparameters are selected based on author recommendations or limited searches. Algorithm-specific hyperparameters and details are provided below.

FMR Appendix C.8-C.10 depicts the search over hyperparameters using PathBB ratio 10-50: $\beta \in \{1.5, 10, 100, 1000\}$, $\alpha \in \{0.1, 1, 10, 100\}$, and credit assignment $\in \{300, 600, 1000\}$. For all tasks, unless stated otherwise, we set the hyperparameters as follows: $\beta = 10$ for the feedback-adaptive temperature, $\alpha = 1$ for the regularization strength (for both actor and critic), and the credit assignment window is set to 600 ms. When oversampling we found using cosine learning rate decay to be helpful for DemoDICE+FMR and ReCOIL+FMR. For Walker2D, we found that using positive feedback for all aligned state-action pairs in D^E and a slightly higher $\alpha = 2$ led to better performance due to the large action space. Although feedback is not typically provided for D^E, we observed that positive feedback for Walker2D helped elevate aligned actions. While adding such feedback is trivial, it was generally unnecessary in spaces with smaller action dimensions.

For critic-based algorithms (e.g., IQL, DemoDICE, ReCOIL), FMR can be applied to the critic under the assumption of a discrete action space. This requires transforming the value function into probability space using a function such as softmax. However, the softmax function is highly sensitive to input scale. When value function outputs have large magnitudes, the softmax produces overly peaked distributions that concentrate probability on a single action, likely limiting the ability to modulate entropy. For actor-critic algorithms, we apply FMR to both the actor and the critic.

IQL Following (Garg et al., 2021), our implementation of IQL uses soft Q-learning to test FMR's ability to manipulate a critic-only algorithm. We implement IQL as a critic-only algorithm to verify FMR's ability to work with critic-only algorithms in discrete spaces. A hyperparameter search was performed over the reward loss $\{0.5, 10\}$ using PathM and all ratios. We set the following hyperparameters for all tasks: χ^2 regularization for the reward loss is 10, and the soft value temperature is 1.

DemoDICE Our implementation of DemoDICE follows (Kim et al., 2022). We set the following hyperparameters for all tasks: the discount factor is 0.99, and the gradient penalty coefficients for the discriminator (i.e., cost) and critic are 0.1 and 1×10^{-4}, respectively.

ReCOIL Our implementation of ReCOIL follows (Sikchi et al., 2024) using actor-critic with target networks. We set the following hyperparameters for all tasks: the discount factor is 0.99, actor temperature is 0.1, value temperature is 1, max clip is 7, reward gap is 2, target Q gap is 200, and the target network is updated using a value of 5×10^{-3}.

DVL Our implementation of DVL follows (Sikchi et al., 2024) using actor-critic with target networks. As DVL is the offline RL variant of ReCOIL that additionally accounts for reward, we

adopt largely the same hyperparameters used for ReCOIL across all tasks: a χ^2-based value loss, discount factor of 0.99, actor temperature of 0.1, value temperature of 1, max clip of 5, and a target network update rate of 5×10^{-3}. To improve performance across all tasks, we set positive feedback for all aligned state-action pairs in D^E. When oversampling we found using cosine learning rate decay to be helpful.

CPL Our implementation of CPL follows (Hejna et al., 2024). To select hyperparameters, we performed a grid search over the following values: segment length $\{16, 32, 64\}$, segment count $\{20\text{k}, 40\text{k}\}$, contrastive bias $\{0.25, 0.5, 0.75, 0.85, 0.95\}$, and temperature $\{0.1, 0.3, 0.5, 1\}$. Based on performance across the PathM, PathBB, and Walker2D tasks across various ratios, we selected a segment length of 64, a segment count of 20k, a contrastive bias of 0.85, and a temperature of 0.1. We use a discount factor of 1 and dense preference comparisons as recommended in (Hejna et al., 2024). The preference score is computed using the sum of evaluative feedback per trajectory segment. To improve performance across all tasks, we set positive feedback for all aligned state-action pairs in D^E.

A.3 Computing Resources

All experiments are run in a SLURM cluster environment across five different types of GPU nodes: NVIDIA A40 with an Intel Xeon Gold 6326 CPU, NVIDIA A100 with an Intel Xeon Gold 6326 CPU, NVIDIA A100 with an AMD EPYC 7502 CPU, NVIDIA L40S with an Intel Xeon Gold 6338 CPU or Intel Xeon Platinum 8362 CPU, and NVIDIA H200 NVL with a dual 64-core AMD EPYC 9555 CPU. Each job uses 40 GB of RAM.

B TASK AND DATA COLLECTION DETAILS

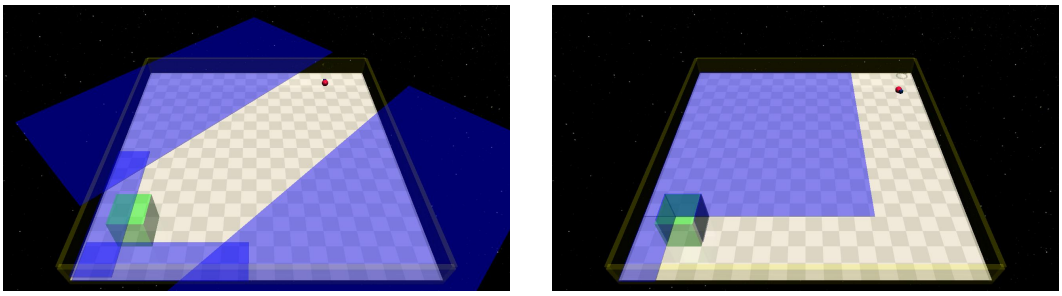
This section provides various visualizations of the different tasks, along with their corresponding trajectories and feedback.

B.1 NAVIGATION POLICIES

Figure 4 depicts the navigation task environment for the aligned PathM and PathBB policies. Blue areas represent hazards, the red sphere with a blue box (indicating the front) represents the agent, and the green cube represents the goal. The yellow walls mark the environment boundaries. These are observable to the agent and do not allow the agent to pass through them. When the agent intersects with a hazard, a red sphere encompasses the agent, indicating it is violating a constraint. This is purely a visual effect for observers.

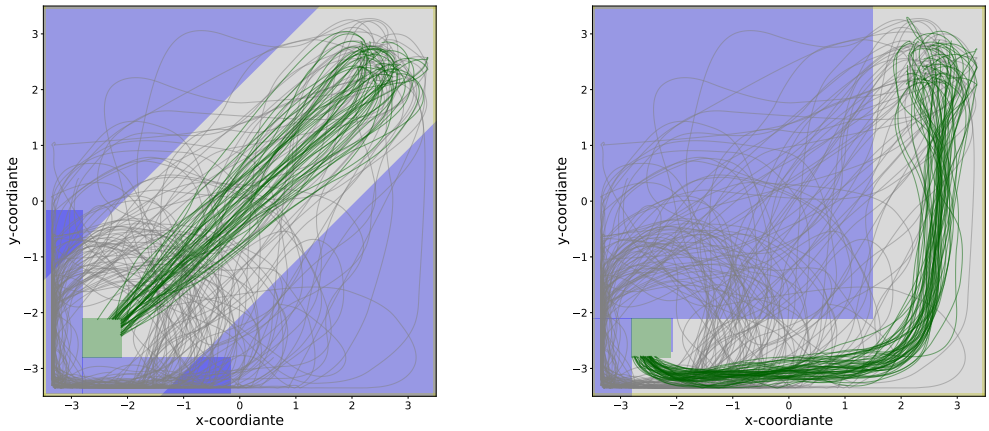
Data Collection For the navigation policies, human demonstrations are collected for both D^E and D^I. For each aligned policy, 50 aligned demonstrations are collected for D^E with zero cost violations and 100% success rate, while 50 imperfect demonstrations are collected for D^I by having humans explore the state space without reaching the goal. Figure 5 visualizes a 2D slice of the environment and trajectories for both aligned navigation policies. The green lines indicate the aligned trajectories for each respective policy. The gray lines (shared across policies) represent the imperfect trajectories. The majority of D^I demonstrations move diagonally through the center (similar to PathM), while a minority follow the top or right boundary before moving toward the goal (similar to PathBB), with all demonstrations circling the goal to obfuscate the objective.

Human feedback is collected by replaying demonstrations through a visual interface, where negative or positive feedback can be provided via keyboard input. When a feedback key is pressed, the raw feedback is assigned to the nearest state. Feedback is collected only for D^I, as it contains demonstrations requiring correction. Policy-specific feedback is collected for PathM and PathBB. The feedback methodology is defined such that the human evaluator provides positive feedback when the agent follows or moves toward the aligned policy, and negative feedback when it diverges or appears to be diverging. Figure 6 displays a heatmap of the human feedback for D^I, provided with respect to each aligned navigation policy. Notice that feedback for Figure 6a encourages the agent to move diagonally toward the goal or back toward the middle of the environment (indicated by green positive feedback). Likewise, Figure 6b encourages the agent to move toward the bottom path, although there are few trajectories that do so.



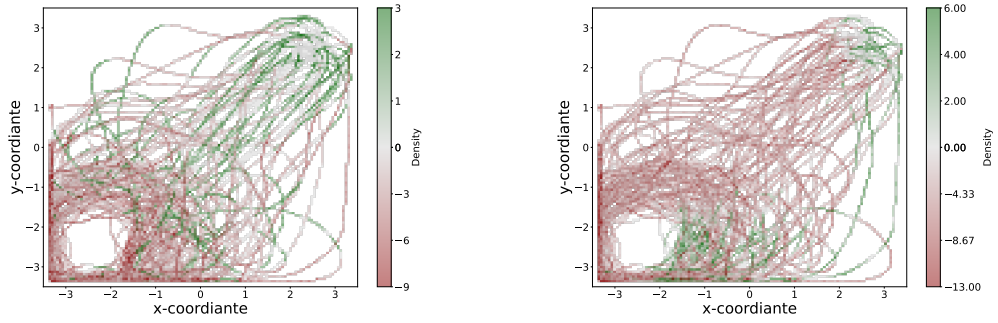
(a) PathM (b) PathBB

Figure 4: Example of navigation task environment setups.



(a) PathM (b) PathBB

Figure 5: Trajectories for the navigation aligned policies depicted as a 2D slice of the environment. The green lines represent aligned trajectories, the gray lines represent imperfect trajectories, and the blue areas represent hazards.



(a) PathM (b) PathBB

Figure 6: Feedback heatmaps of D^I demonstrations for the aligned navigation policies with credit assignment. Color is determined by the feedback density in a particular area in the state space.

B.2 VELOCITY POLICIES

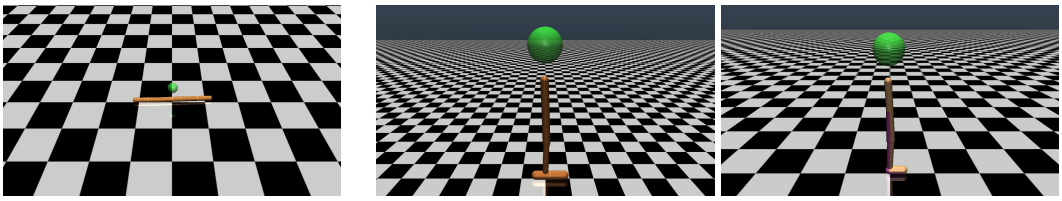
Figure 7 depicts the velocity task environment setups for Swimmer, Hopper, and Walker2D. The sphere above each agent represents whether the agent is exceeding the velocity threshold (this is for visualization purposes only). If the sphere is green, the agent is under the threshold. If the sphere is red, the agent is exceeding the threshold.

Data Collection For the velocity policies, proxy demonstrations are collected for both D^E and D^I using SAC-Discrete. Our implementation of SAC-Discrete follows (Christodoulou, 2019). For all velocity tasks, we used a two-layer neural network with 256 hidden units and ReLU activation for the actor and critic networks. Additionally, we used a learning rate of 3×10^{-4} for the actor and critic and 1×10^{-3} for entropy learning. For Swimmer, we used a discount factor of 0.995 to reach near-optimal performance. The SAC-Discrete agent for Swimmer and Hopper is trained for 5M steps while Walker2D is trained for 10M due to the large action space slowing down learning. For Hopper and Swimmer, samples for D^I are collected every 1M steps while for Walker2D samples are collected at steps 2M, 3M, 4M, 5M, and 8M. The sampling for Walker2D is done to ensure D^I contains samples that have medium overlap with D^E but are of higher quality.

Figure 8 depicts the aligned and the imperfect demonstrations for SlowSwim where the blue dotted line indicates the velocity threshold. Figure 9 depicts the aligned and the imperfect demonstrations for SlowHop where the imperfect demonstrations only cross the threshold. Figure 10 depicts the aligned and the imperfect demonstrations for SlowWalk whose demonstrations act as a middle ground between SlowSwim and SlowHop.

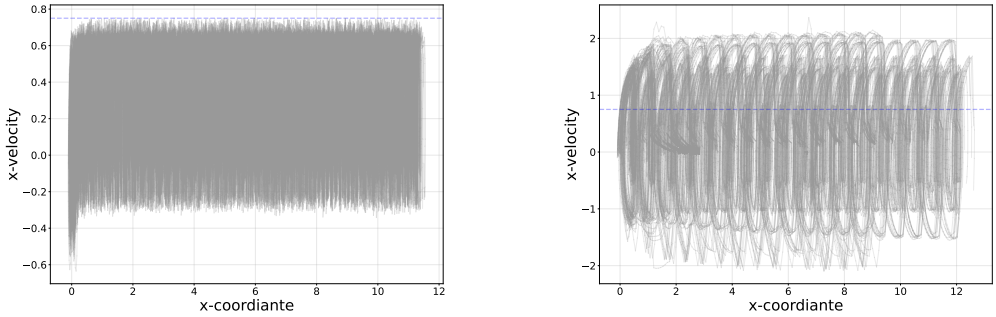
For each velocity aligned policy, demonstrations in D^E contain some cost violations. The performance of the aligned demonstrations D^E for SlowSwim has a return of 283 with a misalignment score of 8×10^{-5}. SlowHop has a return of 1451 with a misalignment score of 0.01. SlowWalk has a return of 2147 with a misalignment score of 8×10^{-3}. Each policy's D^E return is normalized based on these returns.

Feedback is generated using an oracle instead of a human. Oracle feedback is generated by converting the cost into negative feedback, meaning no positive feedback is provided by the oracle. Oracle feedback is used for two reasons: first, to evaluate the viability of synthetic feedback that might be required when human feedback is difficult to provide; second, to evaluate the effectiveness of using only negative feedback.



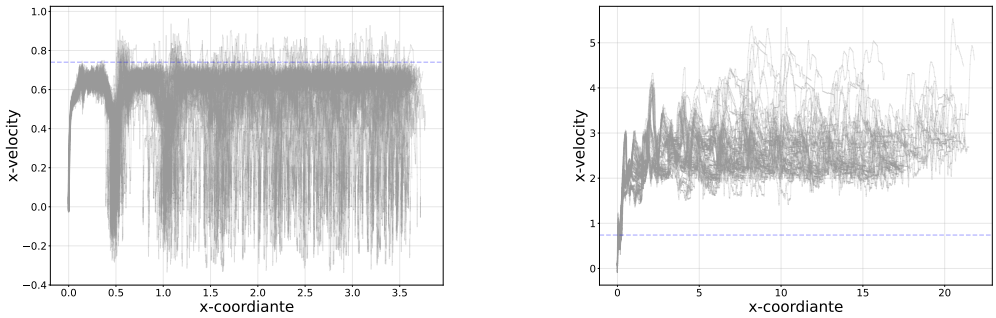
(a) Swimmer (b) Hopper (c) Walker2D

Figure 7: Example of velocity task environment setups.



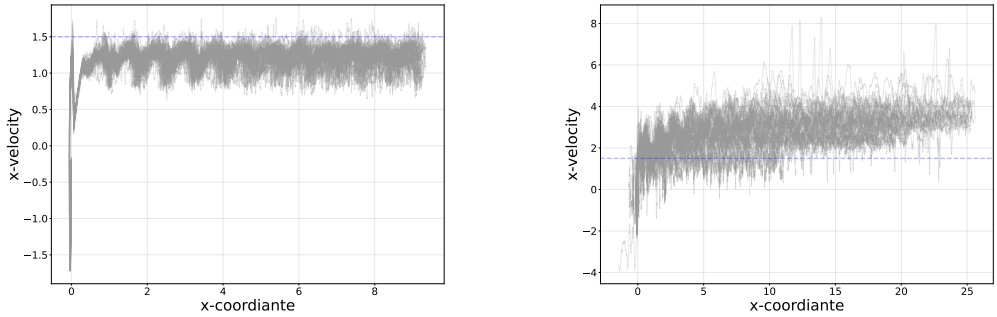
(a) Aligned (b) Imperfect

Figure 8: Trajectories for Swimmer plotted using coordinate and x-velocity. The blue dotted line represents the 0.75 velocity threshold.



(a) Aligned (b) Imperfect

Figure 9: Trajectories for Hopper plotted using coordinate and x-velocity. The blue dotted line represents the 0.74 velocity threshold.



(a) Aligned (b) Imperfect

Figure 10: Trajectories for Walker2D plotted using coordinate and x-velocity. The blue dotted line represents the 1.5 velocity threshold.

C Extended Results

This section provides a collection of additional experimental results mentioned throughout the paper. All learning curveresults are averaged over 5 seeds and smoothed using a moving-average of 10. The shaded region represents the standard error.

C.1 Navigation Task

The following are the results for the navigation tasks.

Table 1: PathM performance comparison across algorithms and data ratios. Results show mean ± std for the last 10 evaluations, over 5 seeds.

		BC		IQL		DemoDICE		ReCOIL	
Variant	**Ratio**	Return	Mis.	Return	Mis.	Return	Mis.	Return	Mis.
Base	50-50	0.45 ± 0.50	0.37 ± 0.31	0.46 ± 0.50	0.36 ± 0.31	0.54 ± 0.50	0.33 ± 0.33	0.95 ± 0.22	0.11 ± 0.21
	25-50	0.37 ± 0.48	0.42 ± 0.31	0.39 ± 0.49	0.42 ± 0.31	0.43 ± 0.50	0.39 ± 0.33	0.92 ± 0.28	0.12 ± 0.22
	10-50	0.31 ± 0.46	0.46 ± 0.29	0.30 ± 0.46	0.47 ± 0.29	0.33 ± 0.47	0.45 ± 0.30	0.86 ± 0.35	0.15 ± 0.24
FMR	50-50	0.87 ± 0.34	0.10 ± 0.23	0.87 ± 0.33	0.09 ± 0.22	0.94 ± 0.24	0.04 ± 0.15	0.99 ± 0.11	0.01 ± 0.08
	25-50	0.88 ± 0.32	0.10 ± 0.22	0.81 ± 0.39	0.14 ± 0.26	0.87 ± 0.33	0.07 ± 0.18	0.97 ± 0.16	0.01 ± 0.08
	10-50	0.81 ± 0.39	0.15 ± 0.25	0.70 ± 0.46	0.21 ± 0.30	0.87 ± 0.34	0.09 ± 0.21	0.95 ± 0.22	0.02 ± 0.09

Table 2: PathBB performance comparison across algorithms and data ratios. Results show mean ± std for the last 10 evaluations, over 5 seeds.

		BC		IQL		DemoDICE		ReCOIL	
Variant	**Ratio**	Return	Mis.	Return	Mis.	Return	Mis.	Return	Mis.
Base	50-50	0.46 ± 0.50	0.31 ± 0.32	0.48 ± 0.50	0.30 ± 0.33	0.58 ± 0.49	0.27 ± 0.35	0.78 ± 0.41	0.15 ± 0.31
	25-50	0.34 ± 0.47	0.39 ± 0.34	0.38 ± 0.48	0.36 ± 0.33	0.43 ± 0.50	0.38 ± 0.39	0.72 ± 0.45	0.15 ± 0.30
	10-50	0.25 ± 0.43	0.44 ± 0.30	0.25 ± 0.43	0.43 ± 0.33	0.31 ± 0.46	0.46 ± 0.36	0.59 ± 0.49	0.24 ± 0.33
FMR	50-50	0.71 ± 0.45	0.16 ± 0.27	0.65 ± 0.48	0.20 ± 0.29	0.84 ± 0.37	0.13 ± 0.25	0.92 ± 0.27	0.05 ± 0.17
	25-50	0.65 ± 0.48	0.20 ± 0.31	0.57 ± 0.49	0.26 ± 0.33	0.76 ± 0.43	0.22 ± 0.33	0.88 ± 0.32	0.07 ± 0.19
	10-50	0.47 ± 0.50	0.33 ± 0.34	0.40 ± 0.49	0.38 ± 0.36	0.67 ± 0.47	0.26 ± 0.35	0.71 ± 0.45	0.10 ± 0.21

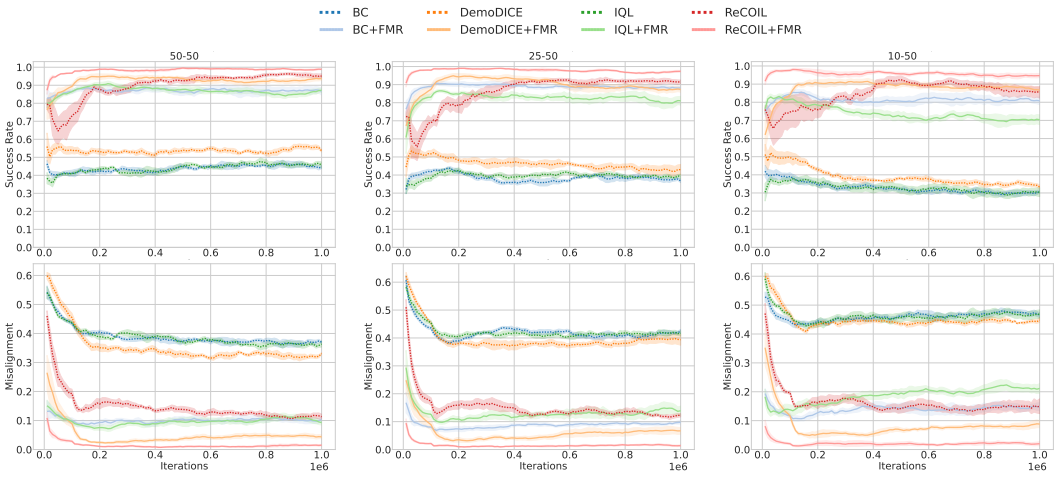


Figure 11: PathM learning curves for baselines and FMR. The shaded region represents the standard error.

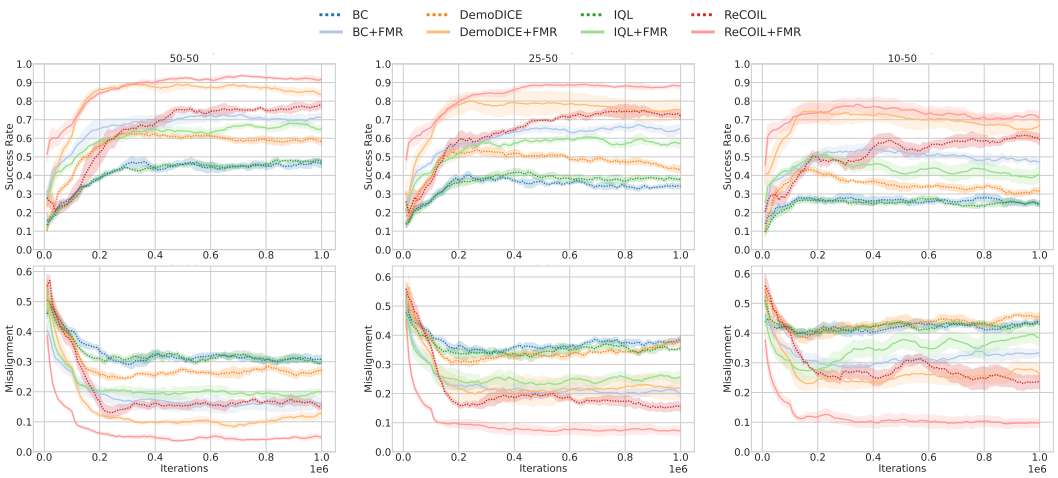


Figure 12: PathBB learning curves for baselines and FMR. The shaded region represents the standard error.

C.2 VELOCITY TASKS

The following are the results for the velocity tasks where return is normalized based on D^E performance.

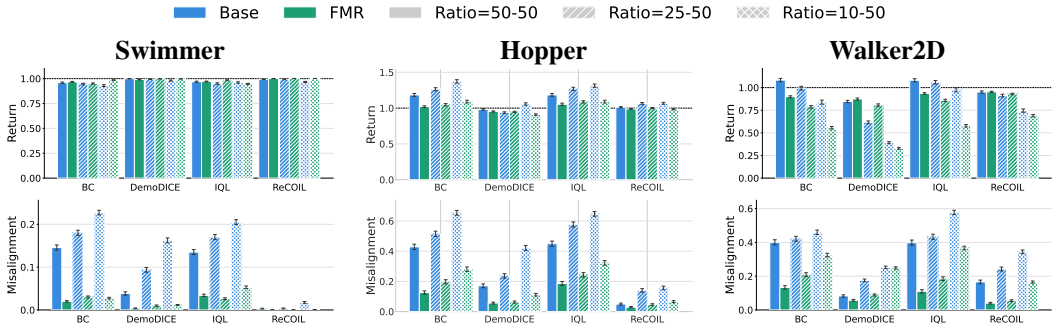


Figure 13: Velocity results across all baseline algorithms and data ratios. Results show mean success rate and misalignment scores with 95% confidence intervals, averaged over the last 10 evaluations across 5 seeds

Table 3: SlowSwim performance comparison across algorithms and data ratios. Results show mean ± std for the last 10 evaluations, over 5 seeds. The * symbol indicates a very small non-zero number.

		BC		IQL		DemoDICE		ReCOIL	
Variant	**Ratio**	Return	Mis.	Return	Mis.	Return	Mis.	Return	Mis.
Base	50-50	0.96 ± 0.18	0.15 ± 0.15	0.97 ± 0.15	0.14 ± 0.15	1.00 ± 0.05	0.04 ± 0.10	0.99 ± 0.07	0.00^* ± 0.02
	25-50	0.95 ± 0.20	0.18 ± 0.15	0.95 ± 0.19	0.17 ± 0.15	1.00 ± 0.08	0.09 ± 0.14	0.99 ± 0.08	0.00^* ± 0.02
	10-50	0.93 ± 0.23	0.23 ± 0.14	0.96 ± 0.17	0.21 ± 0.14	0.98 ± 0.14	0.16 ± 0.15	0.97 ± 0.15	0.02 ± 0.05
FMR	50-50	0.97 ± 0.13	0.02 ± 0.04	0.97 ± 0.09	0.03 ± 0.07	0.99 ± 0.08	0.00^* ± 0.02	1.00 ± 0.01	0.00^* ± 0.00^*
	25-50	0.95 ± 0.14	0.03 ± 0.06	0.99 ± 0.07	0.03 ± 0.05	1.00 ± 0.03	0.01 ± 0.04	1.00 ± 0.01	0.00^* ± 0.00^*
	10-50	0.99 ± 0.07	0.03 ± 0.05	0.95 ± 0.13	0.05 ± 0.08	1.00 ± 0.06	0.01 ± 0.03	1.00 ± 0.01	0.00^* ± 0.00^*

Table 4: SlowHop performance comparison across algorithms and data ratios. Results show mean ± std for the last 10 evaluations, over 5 seeds.

		BC		IQL		DemoDICE		ReCOIL	
Variant	**Ratio**	Return	Mis.	Return	Mis.	Return	Mis.	Return	Mis.
Base	50-50	1.18 ± 0.51	0.43 ± 0.45	1.18 ± 0.49	0.45 ± 0.45	0.98 ± 0.28	0.17 ± 0.32	1.01 ± 0.18	0.05 ± 0.16
	25-50	1.26 ± 0.55	0.51 ± 0.45	1.27 ± 0.55	0.58 ± 0.43	0.94 ± 0.28	0.24 ± 0.37	1.07 ± 0.31	0.14 ± 0.30
	10-50	1.37 ± 0.61	0.65 ± 0.40	1.31 ± 0.59	0.65 ± 0.40	1.06 ± 0.43	0.42 ± 0.41	1.07 ± 0.34	0.16 ± 0.30
FMR	50-50	1.02 ± 0.31	0.13 ± 0.28	1.05 ± 0.38	0.19 ± 0.34	0.95 ± 0.19	0.06 ± 0.16	0.99 ± 0.13	0.03 ± 0.10
	25-50	1.05 ± 0.40	0.20 ± 0.34	1.09 ± 0.41	0.24 ± 0.37	0.95 ± 0.21	0.06 ± 0.17	1.00 ± 0.18	0.05 ± 0.13
	10-50	1.09 ± 0.48	0.28 ± 0.38	1.09 ± 0.48	0.32 ± 0.39	0.91 ± 0.29	0.11 ± 0.22	0.99 ± 0.23	0.07 ± 0.17

Table 5: SlowWalk performance comparison across algorithms and data ratios. Results show mean ± std for the last 10 evaluations, over 5 seeds.

		BC		IQL		DemoDICE		ReCOIL	
Variant	**Ratio**	Return	Mis.	Return	Mis.	Return	Mis.	Return	Mis.
Base	50-50	1.08 ± 0.47	0.40 ± 0.39	1.08 ± 0.45	0.40 ± 0.39	0.85 ± 0.32	0.08 ± 0.17	0.95 ± 0.32	0.17 ± 0.27
	25-50	0.99 ± 0.53	0.42 ± 0.38	1.06 ± 0.49	0.43 ± 0.39	0.62 ± 0.38	0.18 ± 0.20	0.91 ± 0.38	0.24 ± 0.30
	10-50	0.84 ± 0.57	0.46 ± 0.33	0.98 ± 0.57	0.58 ± 0.32	0.39 ± 0.29	0.25 ± 0.18	0.75 ± 0.46	0.34 ± 0.27
FMR	50-50	0.90 ± 0.27	0.13 ± 0.26	0.93 ± 0.23	0.11 ± 0.25	0.87 ± 0.29	0.06 ± 0.13	0.95 ± 0.17	0.04 ± 0.12
	25-50	0.79 ± 0.36	0.21 ± 0.29	0.86 ± 0.30	0.18 ± 0.28	0.81 ± 0.32	0.09 ± 0.15	0.93 ± 0.21	0.06 ± 0.12
	10-50	0.56 ± 0.35	0.33 ± 0.27	0.58 ± 0.36	0.37 ± 0.27	0.33 ± 0.25	0.25 ± 0.17	0.69 ± 0.32	0.17 ± 0.16

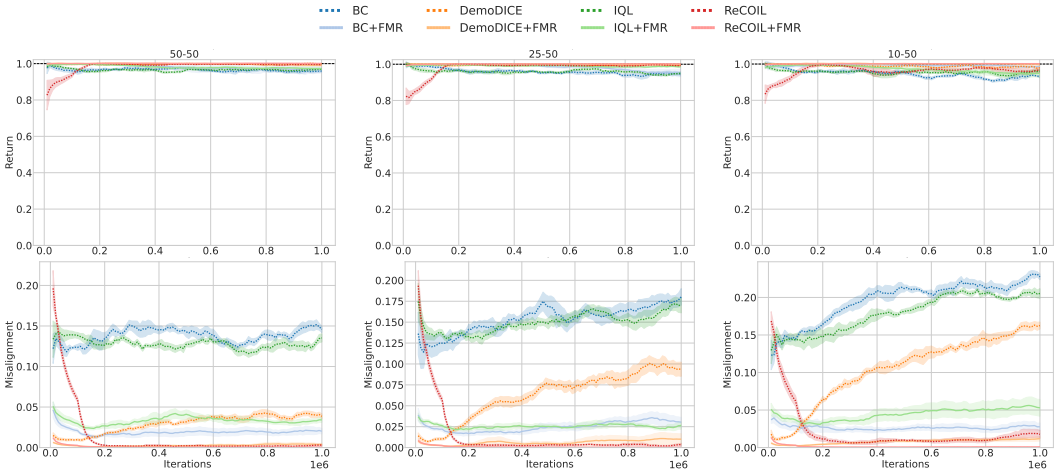


Figure 14: SlowSwim learning curves for baselines and FMR. The shaded region represents the standard error.

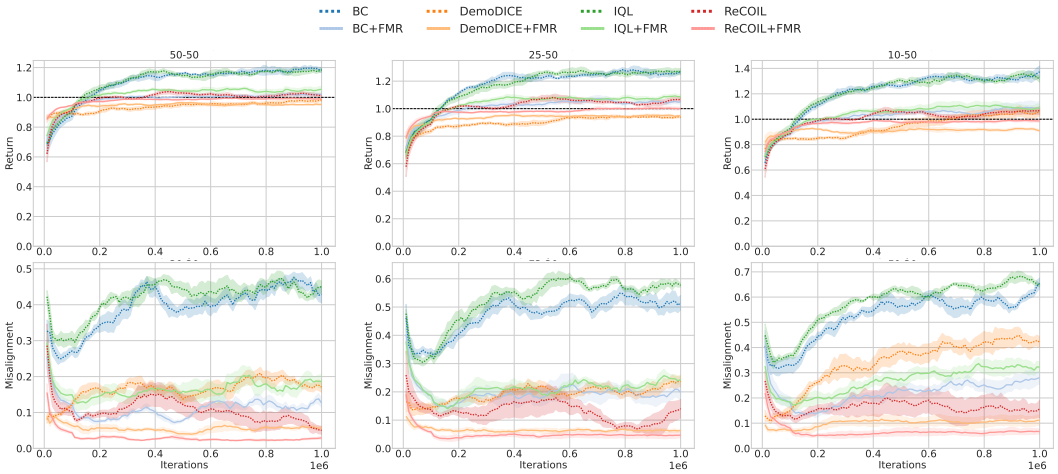


Figure 15: SlowHop learning curves for baselines and FMR. The shaded region represents the standard error.

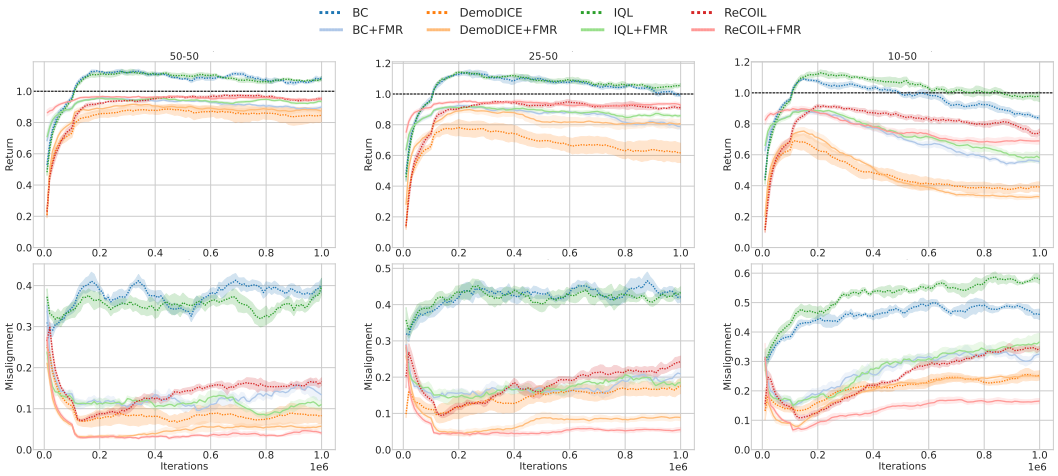


Figure 16: SlowWalk learning curves for baselines and FMR. The shaded region represents the standard error.

C.3 Feedback as Preferences (CPL) and Reward (DVL)

The following results compare FMR with CPL when using the sum feedback to score sub-trajectory preference and DVL when using feedback to replace reward. Results for velocity tasks and additional learning curves are provided.

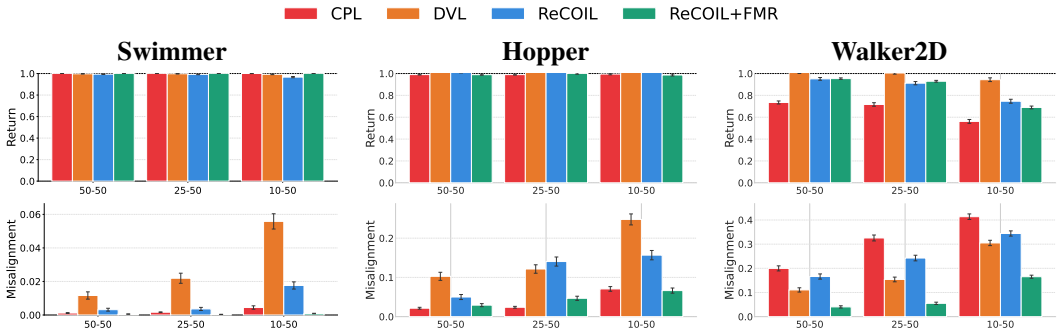


Figure 17: Velocity policies data ratio comparison between FMR, DVL, and CPL with ReCOIL baseline. Results show mean success rate and misalignment scores with 95% confidence intervals, averaged over the last 10 evaluations across 5 seeds.

Table 6: PathM comparison of FMR against alternative methods for utilizing evaluative feedback. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Ratio	Suc.	Mis.
ReCOIL	50-50	0.951 ± 0.215	0.113 ± 0.208
	25-50	0.916 ± 0.278	0.123 ± 0.222
	10-50	0.855 ± 0.352	0.150 ± 0.239
CPL	50-50	0.872 ± 0.334	0.098 ± 0.237
	25-50	0.796 ± 0.403	0.160 ± 0.282
	10-50	0.565 ± 0.496	0.302 ± 0.328
DVL	50-50	0.773 ± 0.419	0.148 ± 0.260
	25-50	0.655 ± 0.476	0.230 ± 0.291
	10-50	0.520 ± 0.500	0.308 ± 0.302
ReCOIL+FMR	50-50	0.988 ± 0.107	0.015 ± 0.084
	25-50	0.974 ± 0.160	0.014 ± 0.076
	10-50	0.947 ± 0.224	0.020 ± 0.092

Table 7: PathBB comparison of FMR against alternative methods for utilizing evaluative feedback. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Ratio	Suc.	Mis.
ReCOIL	10-50	0.592 ± 0.492	0.237 ± 0.334
	25-50	0.721 ± 0.449	0.154 ± 0.296
	50-50	0.780 ± 0.415	0.153 ± 0.305
CPL	10-50	0.573 ± 0.495	0.273 ± 0.359
	25-50	0.685 ± 0.465	0.203 ± 0.341
	50-50	0.781 ± 0.414	0.124 ± 0.267
DVL	10-50	0.400 ± 0.490	0.287 ± 0.286
	25-50	0.652 ± 0.476	0.175 ± 0.274
	50-50	0.771 ± 0.420	0.108 ± 0.222
ReCOIL+FMR	10-50	0.708 ± 0.455	0.097 ± 0.211
	25-50	0.880 ± 0.325	0.073 ± 0.192
	50-50	0.919 ± 0.273	0.048 ± 0.167

Table 8: SlowSwim comparison of FMR against alternative methods for utilizing evaluative feedback. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Ratio	Return	Mis.
ReCOIL	50-50	0.994 ± 0.068	0.003 ± 0.022
	25-50	0.992 ± 0.076	0.004 ± 0.024
	10-50	0.967 ± 0.145	0.018 ± 0.055
CPL	50-50	1.001 ± 0.026	0.001 ± 0.007
	25-50	1.002 ± 0.010	0.002 ± 0.005
	10-50	1.000 ± 0.023	0.004 ± 0.026
DVL	50-50	0.997 ± 0.048	0.012 ± 0.055
	25-50	0.998 ± 0.055	0.022 ± 0.077
	10-50	0.993 ± 0.083	0.056 ± 0.116
ReCOIL+FMR	50-50	1.001 ± 0.008	0.001 ± 0.003
	25-50	1.002 ± 0.008	0.000 ± 0.002
	10-50	1.002 ± 0.008	0.001 ± 0.003

Table 9: SlowHop comparison of FMR against alternative methods for utilizing evaluative feedback. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Ratio	Return	Mis.
CPL	50-50	0.990 ± 0.089	0.022 ± 0.059
	25-50	0.990 ± 0.111	0.024 ± 0.059
	10-50	0.995 ± 0.173	0.071 ± 0.152
DVL	50-50	1.046 ± 0.273	0.103 ± 0.262
	25-50	1.076 ± 0.302	0.121 ± 0.275
	10-50	1.135 ± 0.418	0.248 ± 0.357
ReCOIL	50-50	1.012 ± 0.181	0.050 ± 0.164
	25-50	1.066 ± 0.310	0.140 ± 0.296
	10-50	1.067 ± 0.340	0.157 ± 0.304
ReCOIL+FMR	50-50	0.990 ± 0.127	0.029 ± 0.099
	25-50	1.001 ± 0.179	0.047 ± 0.135
	10-50	0.986 ± 0.225	0.066 ± 0.168

Table 10: SlowWalk comparison of FMR against alternative methods for utilizing evaluative feedback. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Ratio	Return	Mis.
ReCOIL	50-50	0.951 ± 0.317	0.166 ± 0.273
	25-50	0.912 ± 0.379	0.243 ± 0.297
	10-50	0.746 ± 0.460	0.344 ± 0.275
CPL	50-50	0.735 ± 0.352	0.200 ± 0.268
	25-50	0.717 ± 0.411	0.326 ± 0.313
	10-50	0.562 ± 0.449	0.414 ± 0.294
DVL	50-50	1.009 ± 0.221	0.111 ± 0.225
	25-50	1.005 ± 0.267	0.154 ± 0.248
	10-50	0.943 ± 0.436	0.305 ± 0.282
ReCOIL+FMR	50-50	0.954 ± 0.169	0.040 ± 0.119
	25-50	0.929 ± 0.206	0.055 ± 0.122
	10-50	0.690 ± 0.323	0.165 ± 0.162

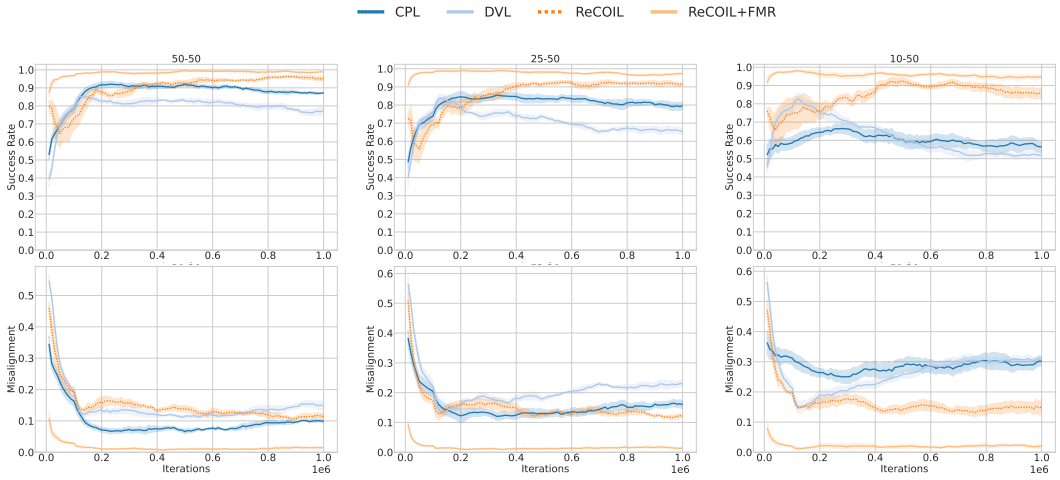


Figure 18: PathM learning curves comparing FMR to alternative methods for utilizing evaluative feedback. The shaded region represents the standard error.

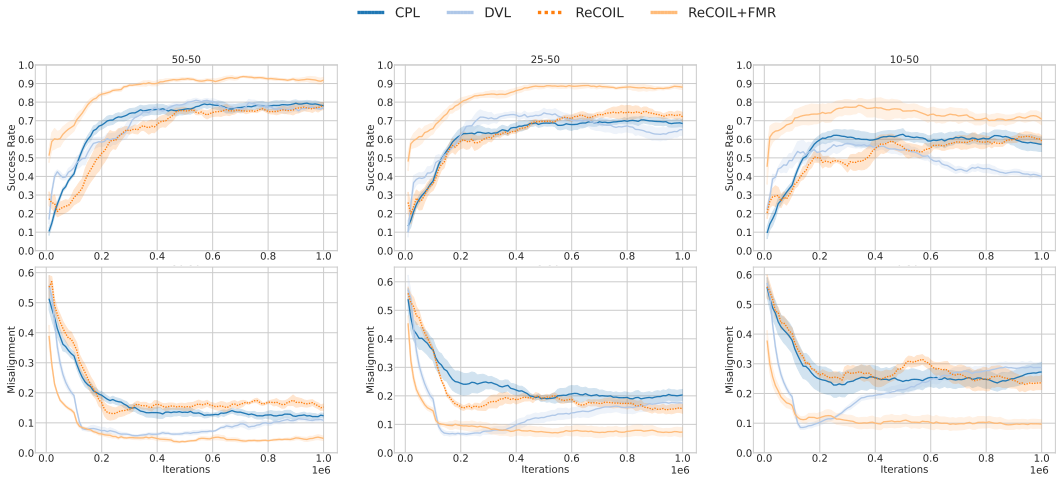


Figure 19: PathBB learning curves comparing FMR to alternative methods for utilizing evaluative feedback. The shaded region represents the standard error.

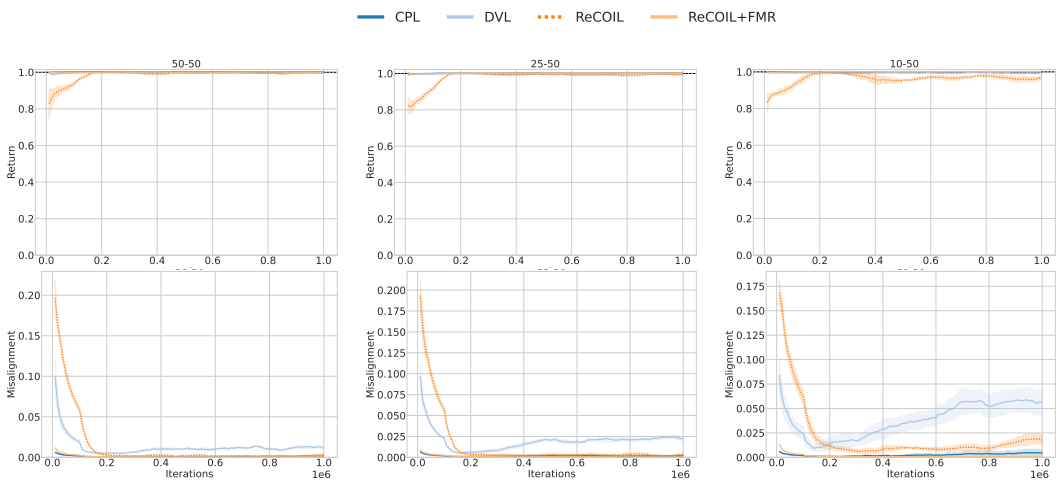


Figure 20: SlowSwim learning curves comparing FMR to alternative methods for utilizing evaluative feedback. The shaded region represents the standard error.

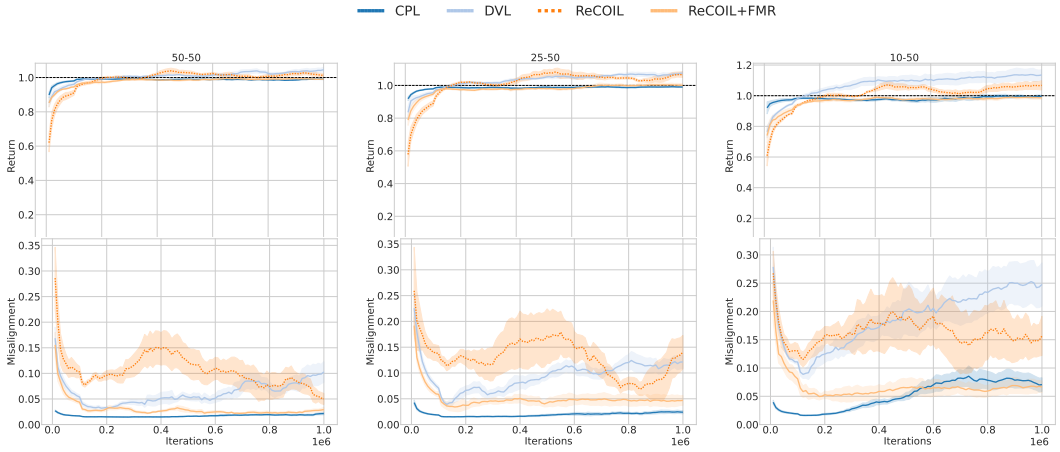


Figure 21: SlowHop learning curves comparing FMR to alternative methods for utilizing evaluative feedback. The shaded region represents the standard error.

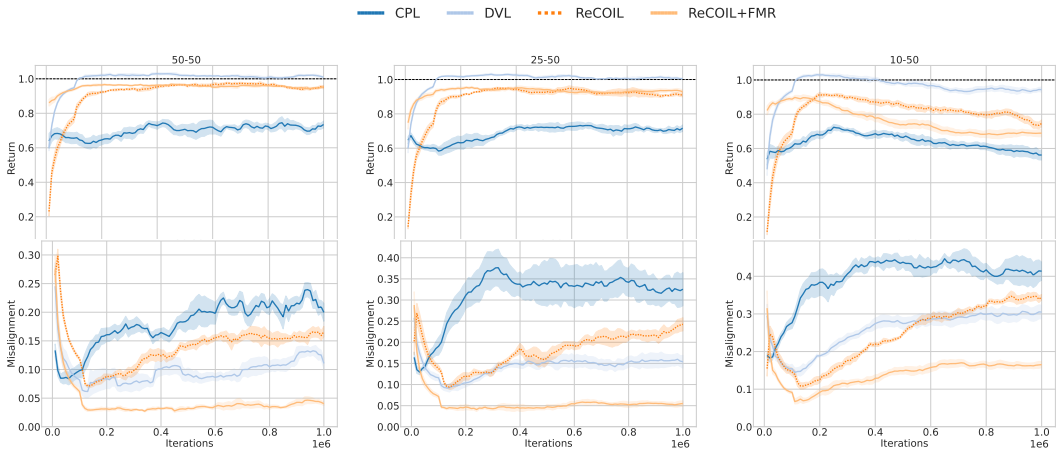


Figure 22: SlowWalk learning curves comparing FMR to alternative methods for utilizing evaluative feedback. The shaded region represents the standard error.

C.4 Only Negative Feedback

Below are the results for the navigation aligned policies, comparing with and without positive feedback for D^I using ReCOIL+FMR.

Table 11: PathM performance comparison for ReCOIL+FMR with and without positive feedback. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Ratio	Suc.	Mis.
w/ pos	50-50	0.988 ± 0.11	0.015 ± 0.08
	25-50	0.974 ± 0.16	0.014 ± 0.08
	10-50	0.947 ± 0.22	0.020 ± 0.09
w/o pos	50-50	0.993 ± 0.08	0.008 ± 0.06
	25-50	0.982 ± 0.13	0.009 ± 0.06
	10-50	0.933 ± 0.25	0.007 ± 0.05

Table 12: PathBB performance comparison for ReCOIL+FMR with and without positive feedback. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Ratio	Suc.	Mis.
w/ pos	50-50	0.919 ± 0.27	0.048 ± 0.17
	25-50	0.880 ± 0.32	0.073 ± 0.19
	10-50	0.708 ± 0.45	0.097 ± 0.21
w/o pos	50-50	0.938 ± 0.24	0.036 ± 0.17
	25-50	0.892 ± 0.31	0.055 ± 0.20
	10-50	0.746 ± 0.44	0.134 ± 0.29

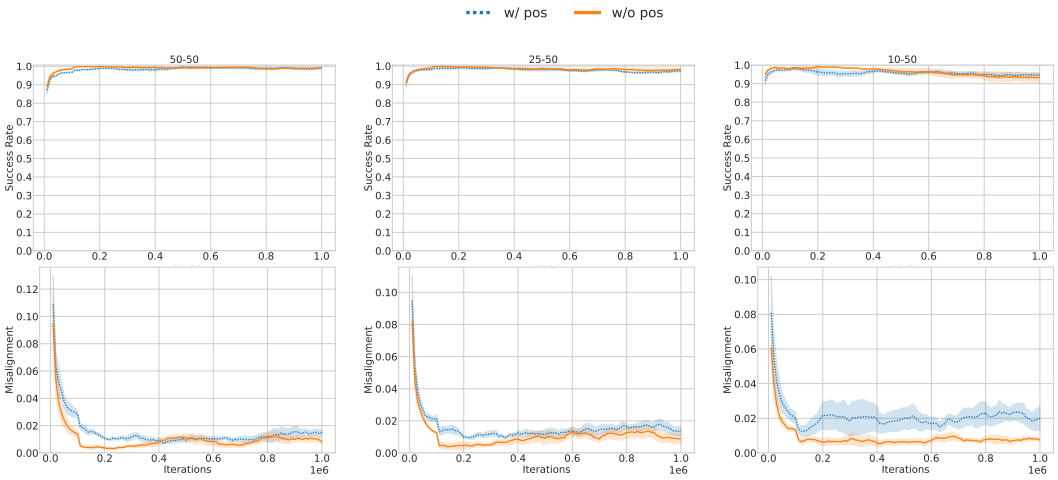


Figure 23: PathM learning curves for ReCOIL+FMR with and without positive feedback. The shaded region represents the standard error.

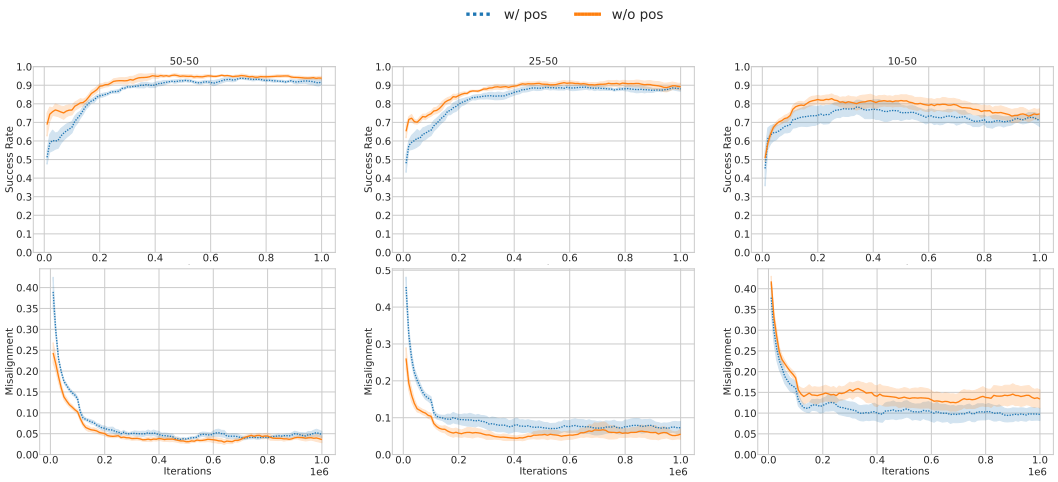


Figure 24: PathBB learning curves for ReCOIL+FMR with and without positive feedback. The shaded region represents the standard error.

C.5 Impact of Imperfect Demonstrations

Subsequent results are presented for testing the usefulness of imperfect demonstrations across all tasks. Results compare ReCOIL+FMR trained using a 10–50 data ratio against BC and IQL when trained on 10 randomly selected aligned demonstrations with oversampling.

Table 13: PathM performance comparison between ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Suc.	Mis.
BC Expert	0.710 ± 0.45	0.167 ± 0.31
IQL Expert	0.801 ± 0.40	0.088 ± 0.24
ReCOIL+FMR	0.947 ± 0.22	0.020 ± 0.09

Table 14: PathBB performance comparison between ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Suc.	Mis.
BC Expert	0.854 ± 0.35	0.068 ± 0.20
IQL Expert	0.883 ± 0.32	0.076 ± 0.22
ReCOIL+FMR	0.708 ± 0.45	0.097 ± 0.21

Table 15: SlowSwim performance comparison between ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Return	Mis.
BC Expert	0.9991 ± 0.034	0.0004 ± 0.001
IQL Expert	1.0017 ± 0.008	0.0003 ± 0.001
ReCOIL+FMR	1.0018 ± 0.008	0.0009 ± 0.003

Table 16: SlowHop performance comparison between ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Return	Mis.
BC Expert	0.974 ± 0.125	0.014 ± 0.006
IQL Expert	0.973 ± 0.130	0.016 ± 0.007
ReCOIL+FMR	0.986 ± 0.225	0.066 ± 0.168

Table 17: SlowWalk performance comparison between ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Algorithm	Return	Mis.
BC Expert	0.944 ± 0.182	0.071 ± 0.086
IQL Expert	0.769 ± 0.315	0.136 ± 0.110
ReCOIL+FMR	0.690 ± 0.323	0.165 ± 0.162

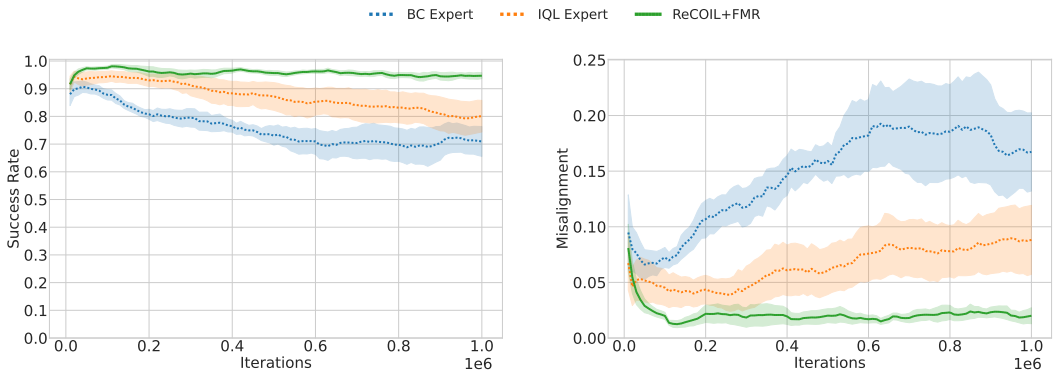


Figure 25: PathM learning curves for ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. The shaded region represents the standard error.

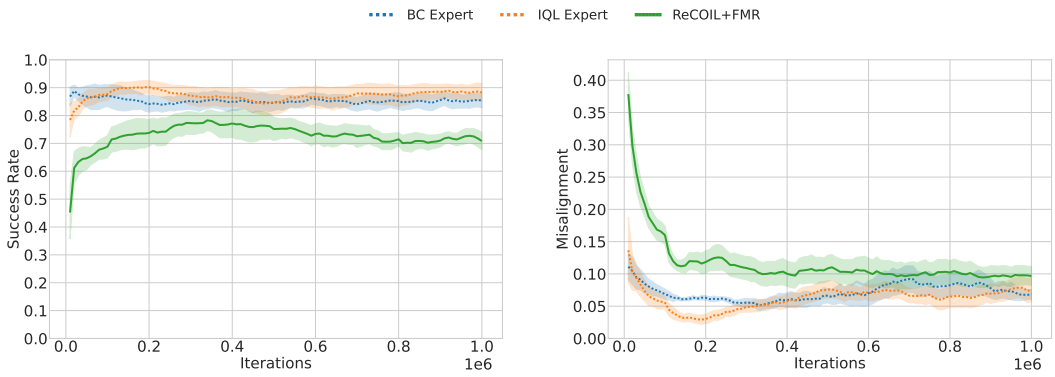


Figure 26: PathBB learning curves for ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. The shaded region represents the standard error.

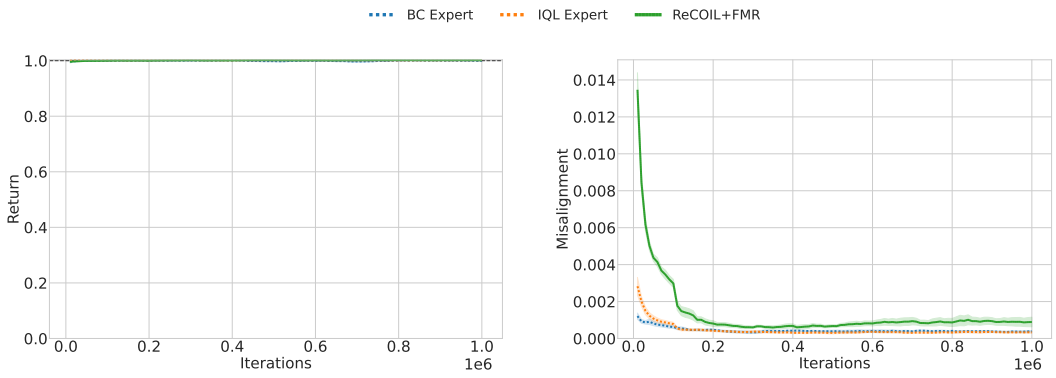


Figure 27: SlowSwim learning curves for ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. The shaded region represents the standard error.

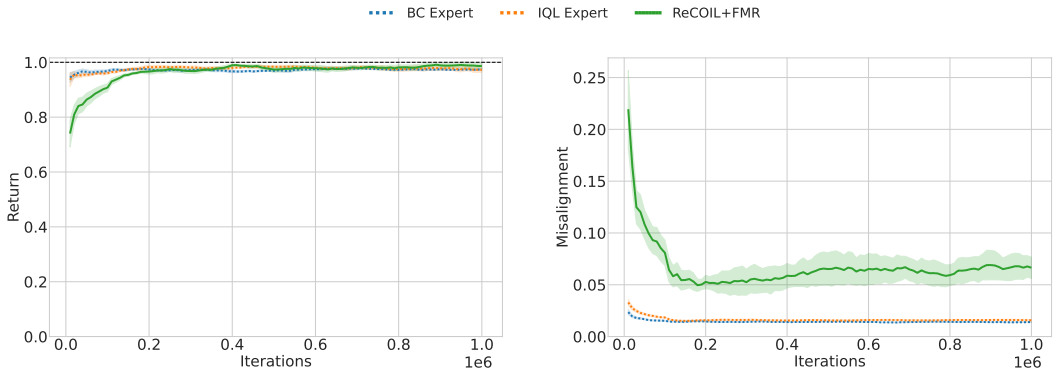


Figure 28: SlowHop learning curves for ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. The shaded region represents the standard error.

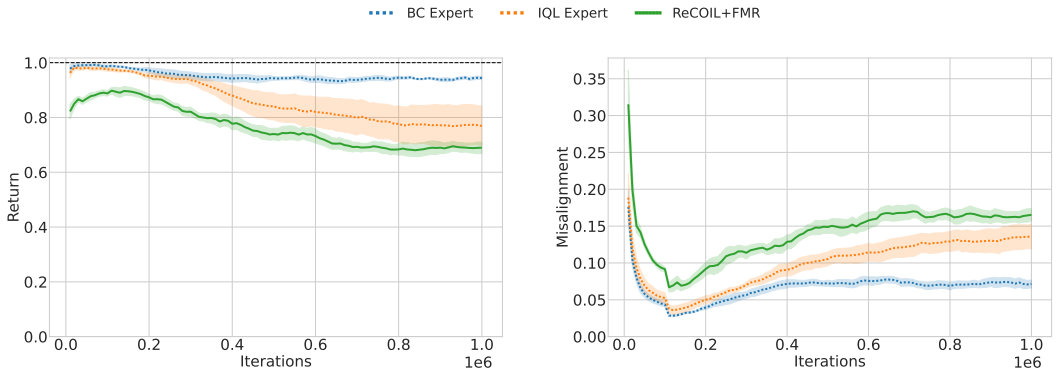


Figure 29: SlowWalk learning curves for ReCOIL+FMR and BC/IQL trained using only aligned demonstrations. The shaded region represents the standard error.

C.6 Feedback Scalability

The following results test the feedback scale using a 50-50 data ratio. Results are reported for ReCOIL+FMR where 0% corresponds to ReCOIL, 20% corresponds to 10 demonstrations with feedback, 50% corresponds to 25 demonstrations with feedback, and 100% corresponds to all demonstrations having feedback. Demonstrations are randomly selected to receive feedback. When a demonstration is selected, feedback for the entire demonstration is used. All remaining demonstrations do not have feedback.

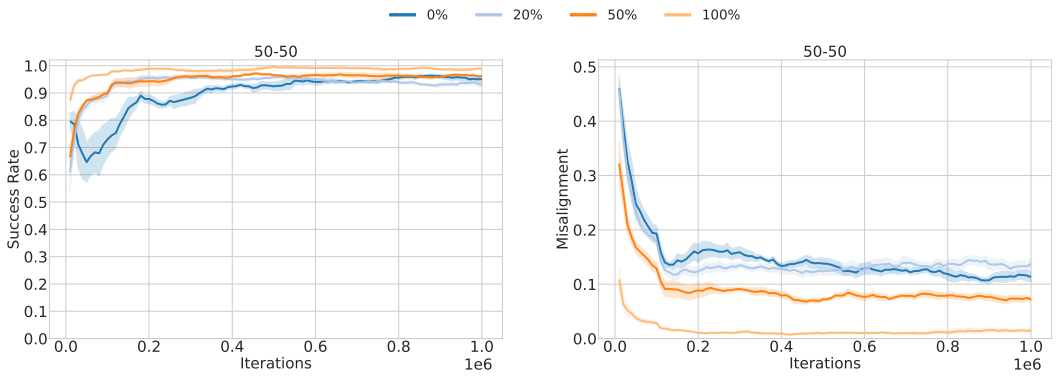


Figure 30: PathM learning curves for ReCOIL+FMR with reduced feedback. The shaded region represents the standard error.

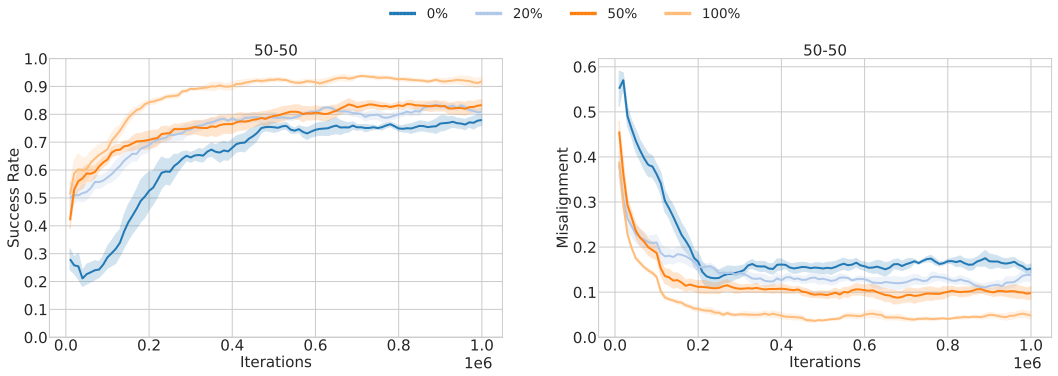


Figure 31: PathBB learning curves for ReCOIL+FMR with reduced feedback.The shaded region represents the standard error.

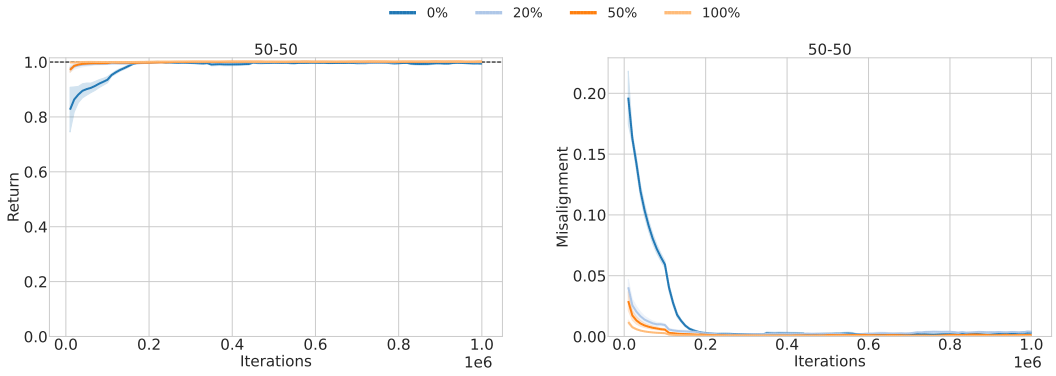


Figure 32: SlowSwim learning curves for ReCOIL+FMR with reduced feedback. The shaded region represents the standard error.

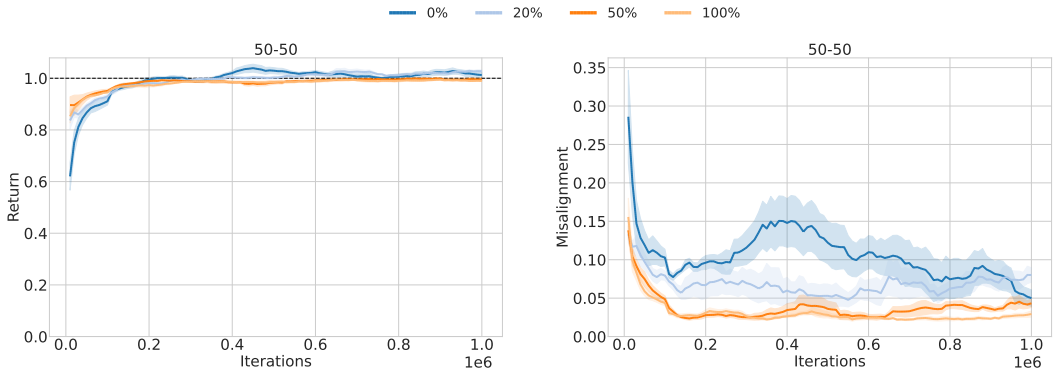


Figure 33: SlowHop learning curves for ReCOIL+FMR with reduced feedback. The shaded region represents the standard error.

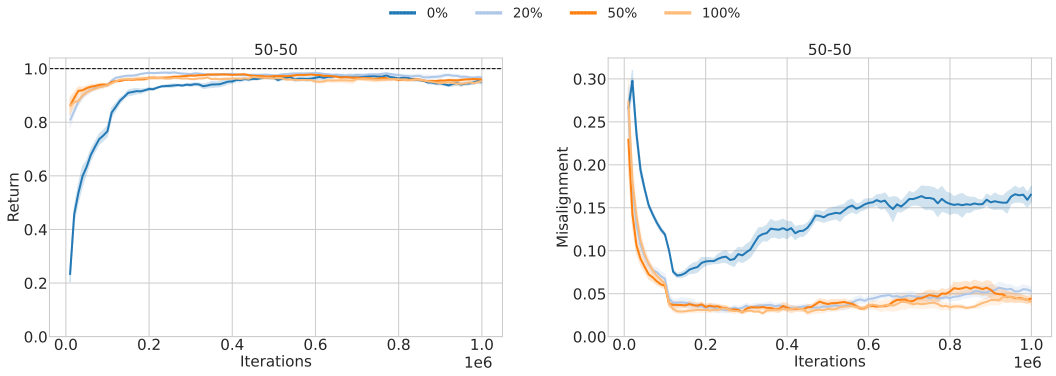


Figure 34: SlowWalk learning curves for ReCOIL+FMR with reduced feedback. The shaded region represents the standard error.

C.7 GENERALIZED FEEDBACK-ADAPTIVE TEMPERATURE

The following are the results when comparing feedback-adaptive temperature τ_j with *generalized feedback-adaptive temperature* τ^{gen}. The comparison is performed across all tasks using ReCOIL+FMR in environments with a discrete action space, demonstrating negligible performance differences. This indicates that either formulation is valid, where τ^{gen} generalizes τ_j beyond discrete action spaces to support continuous action spaces.

Table 18: PathM performance comparison for ReCOIL+FMR when using τ_j and τ^{gen}. Results show mean ± std for the last 10 evaluations, over 5 seeds.

τ Type	Ratio	Suc.	Mis.
τ^{gen}	50-50	0.986 ± 0.12	0.014 ± 0.08
	25-50	0.974 ± 0.16	0.019 ± 0.09
	10-50	0.941 ± 0.24	0.020 ± 0.08
τ_j	50-50	0.988 ± 0.11	0.015 ± 0.08
	25-50	0.974 ± 0.16	0.014 ± 0.08
	10-50	0.947 ± 0.22	0.020 ± 0.09

Table 19: PathBB performance comparison for ReCOIL+FMR when using τ_j and τ^{gen}. Results show mean ± std for the last 10 evaluations, over 5 seeds.

τ Type	Ratio	Suc.	Mis.
τ^{gen}	50-50	0.919 ± 0.27	0.058 ± 0.19
	25-50	0.880 ± 0.33	0.053 ± 0.15
	10-50	0.706 ± 0.46	0.109 ± 0.24
τ_j	50-50	0.919 ± 0.27	0.048 ± 0.17
	25-50	0.880 ± 0.32	0.073 ± 0.19
	10-50	0.708 ± 0.45	0.097 ± 0.21

Table 20: SlowSwim performance comparison for ReCOIL+FMR when using τ_j and τ^{gen}. Results show mean ± std for the last 10 evaluations, over 5 seeds.

τ Type	Ratio	Return	Mis.
τ^{gen}	50-50	1.0014 ± 0.014	0.0007 ± 0.005
	25-50	1.0019 ± 0.008	0.0003 ± 0.001
	10-50	1.0021 ± 0.015	0.0007 ± 0.003
τ_j	50-50	1.0011 ± 0.008	0.0005 ± 0.003
	25-50	1.0017 ± 0.008	0.0004 ± 0.002
	10-50	1.0018 ± 0.008	0.0009 ± 0.003

Table 21: SlowHop performance comparison for ReCOIL+FMR when using τ_j and τ^{gen}. Results show mean ± std for the last 10 evaluations, over 5 seeds.

τ Type	Ratio	Return	Mis.
τ^{gen}	50-50	0.988 ± 0.124	0.033 ± 0.114
	25-50	0.993 ± 0.166	0.042 ± 0.133
	10-50	0.984 ± 0.217	0.061 ± 0.157
τ_j	50-50	0.990 ± 0.127	0.029 ± 0.099
	25-50	1.001 ± 0.179	0.047 ± 0.135
	10-50	0.986 ± 0.225	0.066 ± 0.168

Table 22: SlowWalk performance comparison for ReCOIL+FMR when using τ_j and τ^{gen}. Results show mean ± std for the last 10 evaluations, over 5 seeds.

τ Type	Ratio	Return	Mis.
τ^{gen}	50-50	0.952 ± 0.179	0.035 ± 0.105
	25-50	0.936 ± 0.197	0.046 ± 0.107
	10-50	0.679 ± 0.328	0.152 ± 0.150
τ_j	50-50	0.954 ± 0.169	0.040 ± 0.119
	25-50	0.929 ± 0.206	0.055 ± 0.122
	10-50	0.690 ± 0.323	0.165 ± 0.162

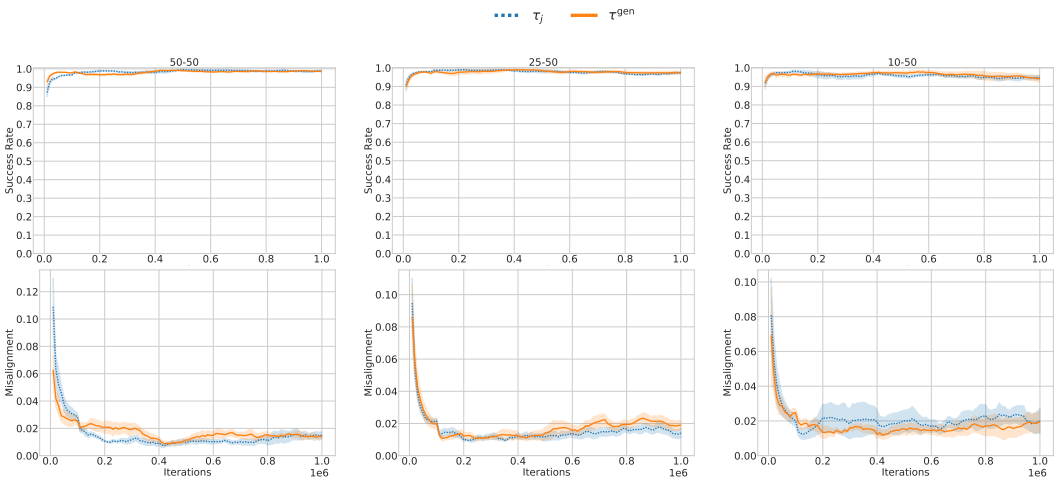


Figure 35: PathM learning curves for ReCOIL+FMR when using τ_j and τ^{gen}. The shaded region represents the standard error.

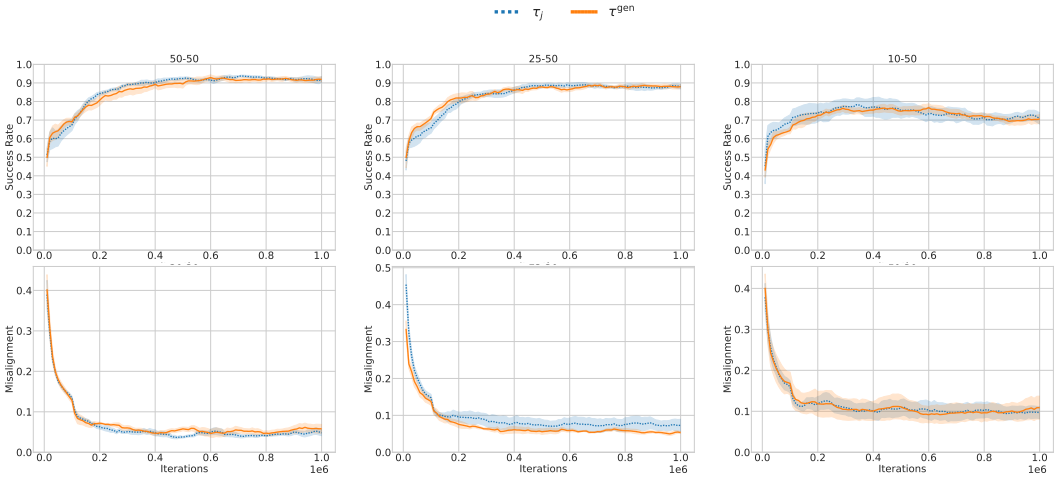


Figure 36: PathBB learning curves for ReCOIL+FMR when using τ_j and τ^{gen}. The shaded region represents the standard error.

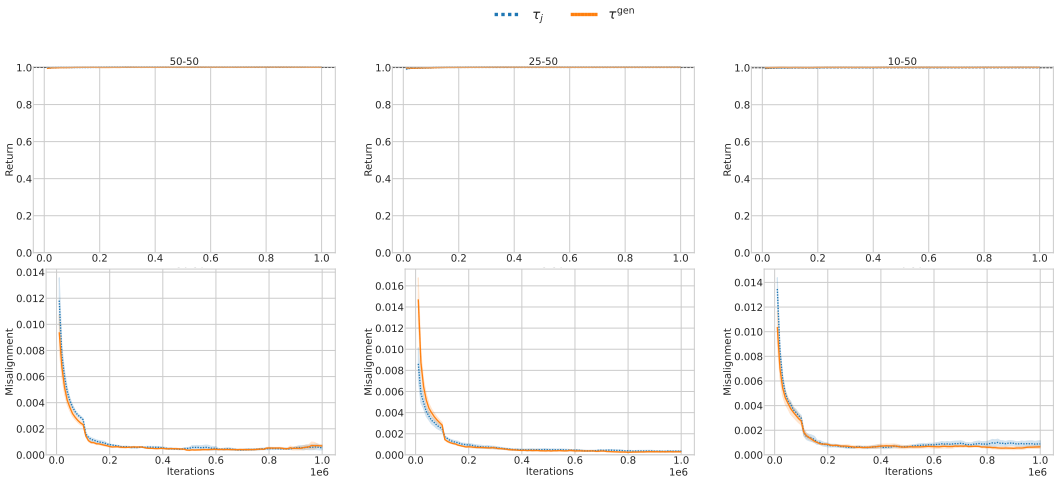


Figure 37: SlowSwim learning curves for ReCOIL+FMR when using τ_j and τ^{gen}. The shaded region represents the standard error.

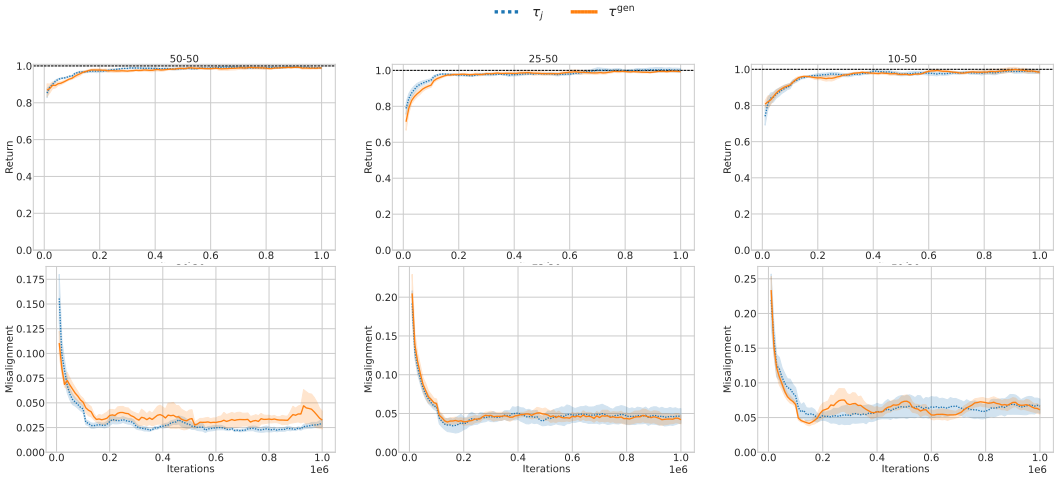


Figure 38: SlowHop learning curves for ReCOIL+FMR when using τ_j and τ^{gen}. The shaded region represents the standard error.

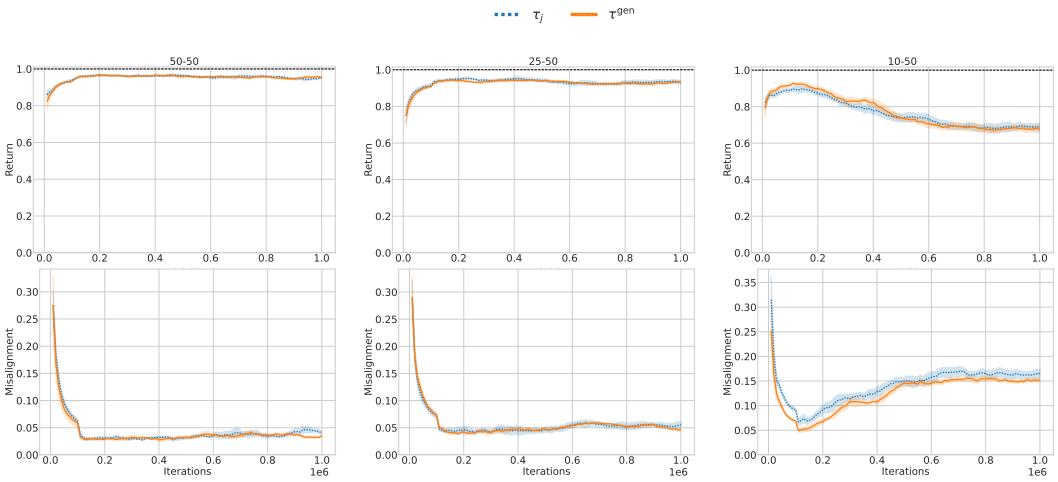


Figure 39: SlowWalk learning curves for ReCOIL+FMR when using τ_j and τ^{gen}. The shaded region represents the standard error.

C.8 β VALUES

Below are the results for various β values when computing the temperature τ_j based on feedback for ReCOIL+FM.

Table 23: PathBB comparison of different β values for ReCOIL+FMR with a 10-50 data ratio. Results show mean ± std for the last 10 evaluations, over 5 seeds.

β	Suc.	Mis.
1.5	0.592 ± 0.49	0.134 ± 0.24
10	0.708 ± 0.45	0.097 ± 0.21
100	0.702 ± 0.46	0.191 ± 0.32
1000	0.709 ± 0.45	0.098 ± 0.24

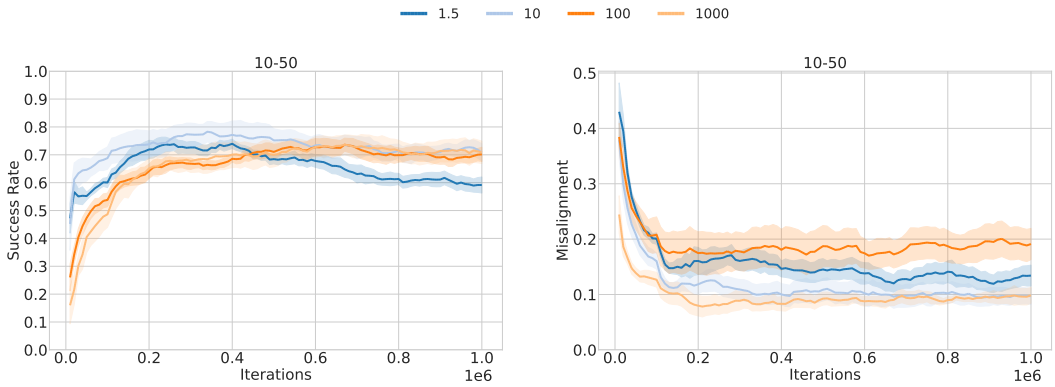


Figure 40: PathBB learning curves of β values for ReCOIL+FMR with a 10-50 data ratio. The shaded region represents the standard error.

C.9 α VALUES

Below are the results for various α values for controlling the regularization or alignment strength of ReCOIL+FM.

Table 24: PathBB performance comparison of different α values for ReCOIL+FMR with a 10-50 data ratio. Results show mean ± std for the last 10 evaluations, over 5 seeds.

α	Suc.	Mis.
0.1	0.506 ± 0.50	0.209 ± 0.31
1	0.708 ± 0.45	0.097 ± 0.21
10	0.672 ± 0.47	0.092 ± 0.21
100	0.510 ± 0.50	0.074 ± 0.17

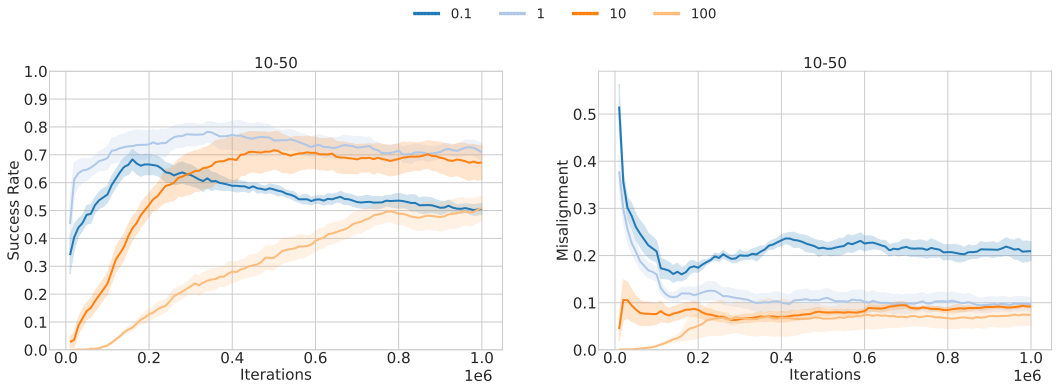


Figure 41: PathBB learning curves of α values for ReCOIL+FMR using PathBB with a 10-50 data ratio. The shaded region represents the standard error.

C.10 Credit Assignment Values

Below are the results for time windows when applying credit assignment to feedback for ReCOIL+FMR.

Table 25: PathBB performance comparison of different credit assignment times for ReCOIL+FMR with a 10-50 data ratio. Results show mean ± std for the last 10 evaluations, over 5 seeds.

Time	Suc.	Mis.
300ms	0.688 ± 0.46	0.124 ± 0.25
600ms	0.708 ± 0.45	0.097 ± 0.21
1000ms	0.758 ± 0.43	0.119 ± 0.30

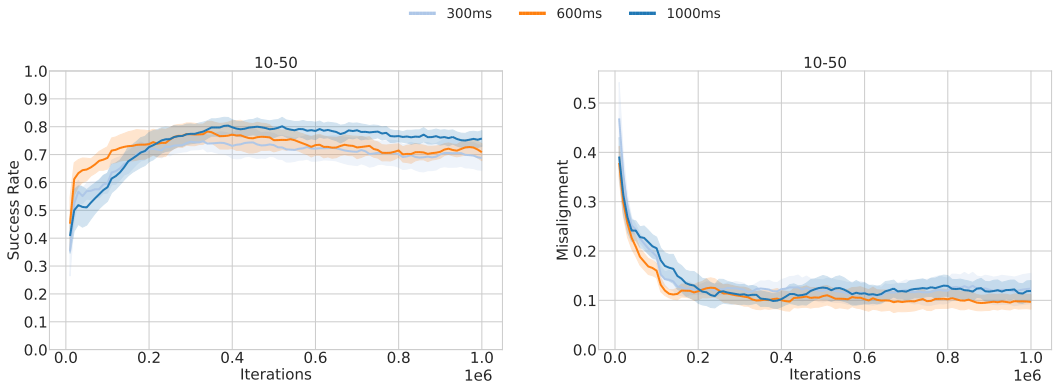
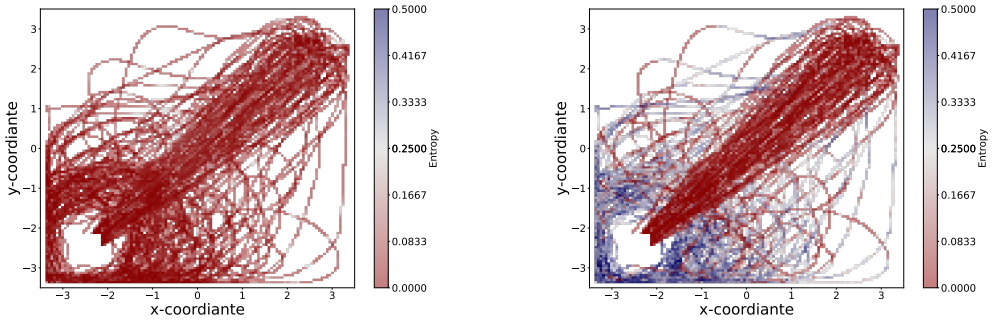


Figure 42: PathBB Learning curves of various credit assignment times for ReCOIL+FMR with a 10-50 data ratio. The shaded region represents the standard error.

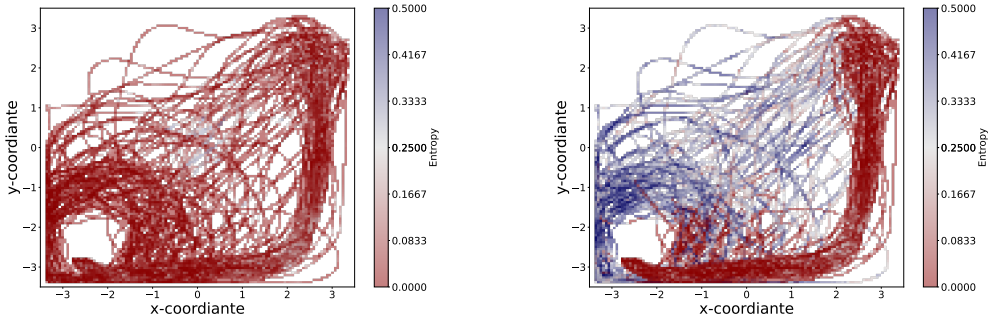
C.11 FMR Induces Feedback-Modulated Entropy

This subsection visualizes state-space entropy for the aligned navigation policies using ReCOIL models. Figure 43 and Figure 44 depict the heatmaps for normalized entropy. To generate these figures, normalized entropy is computed using the final actor model from training, with the expert and imperfect trajectories as input. The average over 5 seeds is then computed. FMR decreases entropy for positive feedback while increasing it for negative feedback. Expert demonstrations receive no feedback and therefore converge to high entropy by default. These entropy heatmaps correspond closely with the original feedback heatmaps given in Figure 6a and Figure 6b.



(a) ReCOIL (b) ReCOIL+FMR

Figure 43: PathM heatmap of normalized entropy for ReCOIL and ReCOIL+FMR.



(a) ReCOIL (b) ReCOIL+FMR

Figure 44: PathBB heatmap of normalized entropy for ReCOIL and ReCOIL+FMR.

D LIMITATIONS AND FUTURE WORK

FMR's primary limitation is scalability, as it relies on state-action-level feedback. However, its superior performance over alternative methods under the same feedback budget suggests more efficient use of feedback, and thus better scalability. Nonetheless, the required feedback volume likely remains dependent on task complexity and the overlap between D^E and D^I. Natural future directions include generalizing feedback across similar state-action pairs to reduce annotation burden (Du et al., 2023) and targeting feedback on D^I at points that deviate from the aligned policy to improve efficacy. Additionally, while this work is limited to discrete action spaces, the τ^{gen} formulation suggests a principled extension to continuous action spaces.