Fast Lifelong Adaptive Inverse Reinforcement Learning from Crowdsourced Demonstrations

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learning from Demonstration (LfD) approaches empower end-users to teach robots novel tasks via demonstrations of the desired behaviors, democratizing access to robotics. However, current LfD frameworks are not capable of fast adaptation to heterogeneous human demonstrations nor the large-scale deployment in ubiquitous robotics applications. In this paper, we propose a novel LfD framework, Fast Lifelong Adaptive Inverse Reinforcement learning (FLAIR). Our approach (1) leverages learned strategies to construct policy mixtures for fast adaptation to new demonstrations, allowing for quick end-user personalization; (2) distills common knowledge across demonstrations, achieving accurate task inference; and (3) expands its model only when needed in lifelong deployments, maintaining a concise set of prototypical strategies that can approximate all behaviors via policy mixtures. We empirically validate that FLAIR achieves *adaptability* (i.e., the robot adapts to heterogeneous, user-specific task preferences), *efficiency* (i.e., the robot achieves sample-efficient adaptation), and *scalability* (i.e., the model grows sublinearly with the number of demonstrations while maintaining high performance). FLAIR surpasses benchmarks across three continuous control tasks with an average 57% improvement in policy returns and an average 78% fewer episodes required for demonstration modeling using policy mixtures. Finally, we demonstrate the success of FLAIR in a real-robot table tennis task.

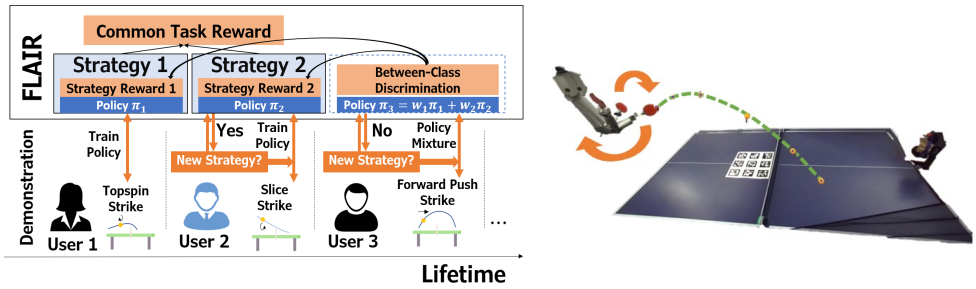
Keywords: Personalized Learning, Learning from Heterogeneous Demonstration, Inverse Reinforcement Learning

1 Introduction

Robots are becoming increasingly ubiquitous with recent advancements in Artificial Intelligence (AI), largely due to the success of Deep Reinforcement Learning (DRL) techniques in generating high-performance continuous control behaviors [1, 2, 3]. However, DRL's success heavily relies on sophisticated reward functions designed for each task. These hand-crafted reward functions typically require iterations of fine-tuning and consultation with domain experts to be effective [4]. Instead, Learning from Demonstration (LfD) approaches democratize access to robotics by having users demonstrate the desired behavior to the robot [5], removing the need for per-task reward engineering. While LfD research strives to empower end-users with the ability to program novel behaviors onto robots, we must consider that end-users may adopt varying preferences and strategies in how they complete the same task [6]. An LfD framework that assumes homogeneity across the set of provided demonstrations could cause the robot to fail to infer the accurate intention, resulting in unwanted or even unsafe behavior [7, 8]. On the other hand, embracing individual preferences can help robots achieve better performance and long-term acceptance from humans [9].

While personalization is important for accurate recovery of the specific demonstrator's behavior, personalization can also prove inefficient if each individual policy must be inferred separately. To avoid this, prior work, MSRD [10], decomposed shared and individual-specific reward information

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.



(a) (b)

Figure 1: (a) This figure shows an illustration of the lifelong learning process with our proposed method, FLAIR. As each demonstrator performs their strike, FLAIR determines whether the demonstration is novel. If a demonstration can be explained by a *policy mixture* of previously learned strategies, FLAIR accepts the policy mixture without training a new strategy. If the mixture explanation is not close to the demonstration, FLAIR creates a new strategy and a prototype policy for the demonstration. Common knowledge between strategy rewards is distilled into a common task reward [10]. (b) This figure illustrates a topspin and slice mixture policy on a WAM robot in a table tennis task. The mix of topspin and slice results in a flat push-like behavior at the contact point.

across heterogeneous demonstrations (i.e., demonstrations seeking to accomplish the same task with different styles). While MSRD makes significant improvements on the accuracy and efficiency in personalized policy modeling, the framework must be trained all-at-once and is unable to handle *incremental/lifelong learning*, a more realistic paradigm for LfD real-world applications.

In this work, we develop FLAIR: Fast Lifelong Adaptive Inverse Reinforcement learning. As a running example, consider a series of humans teaching a robot how to play table tennis, a compelling robot learning platform utilized in prior work [11, 12, 13]. Users of the robot may have their own preferences for table tennis strikes. As shown in Figure 1(a), the first user demonstrates a topspin strike, while the second user demonstrates a slice strike. Instead of repeating a strategy that has been seen, the third user demonstrates a forward flat strike, which could be explained by a mixture (i.e., a composition of known behaviors) of the previously seen topspin and slice prototypical behaviors.

Unlike prior LfD algorithms, FLAIR is capable of continually learning and refining a set of prototypical strategies either to (1) efficiently model new table tennis strike demonstrations as mixtures of the acquired prototypes (e.g., the third user in our example) or (2) incorporate a new strategy as a prototype if the strategy is sufficiently unique (e.g., the second user). In a lifelong learning scenario where demonstrations arrive in sequence, FLAIR autonomously identifies prototypical strategies, distills common knowledge across strategies, and precisely models each demonstration as prototypical strategies or mixtures. We show FLAIR accomplishes *adaptivity*, *efficiency*, and *scalability* in LfD tasks in simulated and real robot experiments:

1. **Adaptive Learning**: We display the *adaptivity* of FLAIR by successfully personalizing to heterogeneous demonstrations on three simulated continuous control tasks. FLAIR models demonstrations better than best benchmarks and achieves an average of 57% higher returns on the task.
2. **Efficient Adaptation**: We demonstrate better sample *efficiency* of FLAIR by showing its mixture optimization needs an average of 78% fewer samples to model demonstrations compared with training a new policy, which prior frameworks require.
3. **Lifelong Scalability**: We showcase the *scalability* of FLAIR in a simulated experiment obtaining 100 demonstrations sequentially. FLAIR identifies on average eleven strategies and utilizes *policy mixtures* to achieve a precise representation of each demonstration, providing empirical evidence for FLAIR's ability to learn a compact set of prototypical strategies in lifelong learning.
4. **Robot Demonstration**: We demonstrate FLAIR's ability to construct successful policy mixtures achieving personalization in a real-world table tennis robot experiment.

2 Related Work

Two common approaches in LfD are to either directly learn a policy, i.e., Imitation Learning (IL), or infer a reward to train a policy, i.e., Inverse Reinforcement Learning (IRL) [14]. IL learns a direct mapping from states to the actions demonstrated [15, 16]. Although a straightforward approach, IL suffers from correspondence matching issues and is not robust to changes in environment dynamics due to its mimicry of the demonstrated behaviors [17, 18]. IRL, on the other hand, infers the demonstrator's latent intent in a more robust and transferable form of a reward function [19].

Although traditional IRL approaches often overlook heterogeneity within demonstrations, there has been recent work that models heterogeneous demonstrations [20, 21, 22, 23, 24]. One intuitive way is to classify demonstrations into homogeneous clusters before applying IRL [6]. The Expectation Maximization (EM) algorithm also operates on a similar idea and iterates between E-step and M-step, where E-step clusters demonstrations and M-step solves the IRL problem on each cluster [25, 26]. When the number of strategies is unknown, a Dirichlet Process prior [27, 28, 29] or non-parametric methods [30] could be used. In these approaches, each reward function only learns from a portion of the demonstrations, making them prone to the issue of reward ambiguity [10]. Furthermore, these methods assume access to all demonstrations beforehand, which is not realistic for LfD algorithm deployment. We instead consider the more realistic setting of lifelong learning [31], where an agent adapts to new demos through its lifetime and continually builds its knowledge base.

Despite the abundance of previous approaches, few consider the relationship between the policies learned to represent each demonstration. Our method, FLAIR, exploits these relationships to not only model heterogeneous demonstrations (*adaptability*), but do so by creating expressive policy mixtures from previously extracted strategies (*efficiency*), and can scale to model large number of demonstrations utilizing a compact set of strategies (*scalability*).

3 Preliminaries

In this section, we introduce preliminaries on Markov Decision Processes (MDP), Inverse Reinforcement Learning (IRL), and Multi-Strategy Reward Distillation (MSRD).

Markov Decision Process – A MDP, M, is a 6-tuple, $\langle \mathbb{S}, \mathbb{A}, R, T, \gamma, \rho_0 \rangle$. $\mathbb{S}$ and $\mathbb{A}$ are the state and action space, respectively. R is the reward function, meaning the environment provides $R(s)$ reward when the agent is in state s. $T(s'|s,a)$ is the probability of transitioning into state s' after taking action a in state s. $\gamma \in (0,1)$ is the temporal discount factor. ρ_0 denotes the initial state probability. A policy, $\pi(a|s)$, represents the probability of choosing an action given the state and is trained to maximize the expected cumulative reward, $\pi^* = \arg\max_\pi \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R(s_t)\right]$, where $\tau = \{s_1, a_1, s_2, a_2, \cdots\}$ is a trajectory.

Inverse Reinforcement Learning – IRL considers an MDP sans reward function (MDP\R) and aims to infer the reward function R based on a set of demonstration trajectories $\mathcal{U} = \{\tau_1, \tau_2, \cdots, \tau_N\}$, where N is the number of demonstrations. Our method is based on Adversarial Inverse Reinforcement Learning (AIRL) [18], which solves the IRL problem with a generative-adversarial setup. The discriminator, D_θ, predicts whether the transition, (s_t, s_{t+1}), belongs to a demonstrator vs. the generator, $\pi_\phi(a|s)$. π_ϕ is trained to maximize the pseudo-reward given by the discriminator via the policy gradient.

Multi-Strategy Reward Distillation – MSRD [10] assumes access to the strategy label, $c_{\tau_i} \in \{1, 2, \cdots, M\}$, for each demonstration, τ_i (M is the number of strategies), and decomposes the per-strategy reward, R_i, for strategy i as a linear combination of a common task reward, R_{Task}, and a strategy-only reward, $R_{\text{S-}i}$. MSRD parameterizes the task reward by θ_{Task} and strategy-only reward by $\theta_{\text{S}-i}$. MSRD takes AIRL as its backbone IRL algorithm, and adds a regularization loss which distills common knowledge into θ_{Task} and only keeps personalized information in $\theta_{\text{S}-i}$. The MSRD loss for the discriminator (the reward) is shown in Equation 1.

$$\begin{aligned} L_D = &- \mathbb{E}_{(\tau,c_\tau)\sim\mathcal{U}}\left[\log D_{\theta_{\text{Task}},\theta_{\text{S-}c_\tau}}(s_t, s_{t+1})\right] - \mathbb{E}_{(\tau,c_\tau)\sim\pi_\phi}\left[\log\left(1 - D_{\theta_{\text{Task}},\theta_{\text{S-}c_\tau}}(s_t, s_{t+1})\right)\right] \\ &+ \alpha \mathbb{E}_{(\tau,c_\tau)\sim\pi_\phi}\left[||R_{\text{S-}c_\tau}(s_t)||_2\right] \end{aligned} \tag{1}$$

4 Method

In this section, we start by introducing the problem setup and notations. We then provide an overview of FLAIR, and its key components: *policy mixture* and *between-class discrimination*.

In our problem setup, We consider a lifelong learning from heterogeneous demonstration process where demonstrations arrive in sequence, as illustrated in Figure 1(a). We denote the i-th arrived demonstration as τ_i. Unlike prior work, FLAIR does not assume access to the ground-truth strategy label, c_{τ_i}. Similar to MSRD, FLAIR learns a shared task reward $R_{\theta_{\text{Task}}}$, strategy rewards $R_{\theta_{\text{S-}j}}$, and policies corresponding to each strategy π_{ϕ_j}. We define the number of prototype strategies created by FLAIR till demonstration τ_i as M_i, and $\eta_R(\tau) = \sum_{t=1}^{\infty} \gamma^{t-1} R_\theta(s_t)$ as trajectory τ's discounted cumulative reward with the inferred reward function R_θ. The goal of the problem is to accurately model each demonstration sequentially with as few environment samples as possible.

4.1 Fast Lifelong Adaptive Inverse Reinforcement Learning (FLAIR)

In our lifelong learning from heterogeneous demonstration problem setup, when a new demonstration τ_i becomes available, we seek to accomplish two key goals: a) design a policy that solves the task while personalizing to the demonstration (i.e., the standard objective in personalized LfD), and b) incorporate knowledge from the demonstration to facilitate precise, efficient, and scalable adaptation to future demonstrations (i.e., the characteristics required for a lifelong LfD framework). We present our method, FLAIR, in pseudocode in Algorithm 1.

Algorithm 1: FLAIR

Input : Demonstration modeling quality threshold ϵ

1 $M_0 = 0$, MixtureWeights=[], m=[]
2 **while** *lifetime learning from heterogeneous demonstration* **do**
3 Obtain demonstration τ_i
4 $\vec{w}_i \leftarrow$`PolicyMixtureOptimization`$(\tau_i, \{\pi_{\phi_j}\}_{j=1}^{M_i})$
5 $D_{\text{KL}}^{\text{mix}} \leftarrow \mathbb{E}_{\tau \sim \pi_{\vec{w}_i}} D_{\text{KL}}(\tau_i, \tau)$
6 **if** $D_{KL}^{mix} < \epsilon$ **then**
7 MixtureWeights[i]$\leftarrow \vec{w}_i$, $M_{i+1} \leftarrow M_i$
8 **else**
9 $\pi_{\text{new}}, R_{\theta_{\text{S-}(M_i+1)}} \leftarrow$`AIRL`$(\tau_i)$
10 $D_{\text{KL}}^{\text{new}} \leftarrow \mathbb{E}_{\tau \sim \pi_{\text{new}}} D_{\text{KL}}(\tau_i, \tau)$
11 **if** $D_{KL}^{mix} < D_{KL}^{new}$ **then**
12 MixtureWeights[i]$\leftarrow \vec{w}_i$, $M_{i+1} \leftarrow M_i$
13 **else**
14 $M_{i+1} \leftarrow M_i + 1$
15 $m_{M_{i+1}} \leftarrow i$
16 MixtureWeights[i]$\leftarrow [\underbrace{0, 0, \cdots, 0}_{M_i \text{ zeros}}, 1]$
17 Update $R_{\theta_{\text{Task}}}, R_{\theta_{\text{S-}j}}, \pi_{\phi_j}$ by `Between-Class Discrimination` and `MSRD`

To accomplish these goals, FLAIR decides whether to explain a new demonstration with previously learned policies (a highly efficient approach), or create a new strategy from scratch (a fallback technique). In the first case, FLAIR attempts to explain the new demonstration, τ_i, by constructing *policy mixtures* (Line 4) with previously learned strategies. FLAIR evaluates the optimized mixture according to the demonstration recovery objective (line 5). If the trajectory generated by the mixture is close to the demonstration (evidenced by the KL-divergence between the *policy mixture* trajectory and the demonstration state distributions falling under a threshold, ϵ), FLAIR adopts the mixture without considering creating a new strategy (line 6). Since the *policy mixture* optimization (details in Section 4.2) is more sample efficient than the AIRL training-from-scratch, FLAIR can bypass

the computationally expensive new-strategy training (line 9) if the mixture provides a high-quality recovery of the demonstrated behavior. This procedure results in an *efficient* policy inference.

If the mixture does not meet the quality threshold, ϵ, FLAIR trains a new strategy by AIRL with the new demonstration, and compares the quality of the new policy to the *policy mixture* (Lines 9-11). If the mixture performs better, we accept the mixture weights to represent τ_i (line 12). If the new strategy performs better, we accept the new strategy as an additional prototype and update our reward and policy model (accordingly, in Line 14, we increment the number of strategies by one). Further, we call the demonstration, τ_i, the "pure" demonstration for strategy M_{i+1}, meaning strategy M_{i+1} purely represents demonstration τ_i (line 15). As such, the mixture weight for τ_i is a one-hot vector on strategy M_{i+1} (lines 16).

To effectively maintain a knowledge base, we propose a novel training signal named *Between-Class Discrimination* (BCD). BCD trains each strategy reward to capture the fact that each demonstration has a certain percentage of the strategy. In the table tennis example (Figure 1(a)), the third user's behavior is a mixture of the topspin and the slice, indicating topspin and slice strategy rewards should partially recognize the third demonstration. BCD encourages the two strategy rewards to give partial rewards to the third demonstration. In addition to BCD, FLAIR also optimizes MSRD loss (Equation 1) for all strategies with their corresponding pure demonstrations, and updates the generator policies based on the learned reward.

4.2 Policy Mixture Optimization

To achieve efficient personalization for a new demonstration τ_i (Line 4 of Algorithm 1), we construct a *policy mixture* with a linear geometric combination of existing policies $\pi_1, \pi_2, \cdots, \pi_{M_i}$ (Equation 2), where $w_{i,j} \geq 0$ are learned weights such that: $\sum_{j=1}^{M_i} w_{i,j} = 1$.

$$\pi_{\vec{w}_i}(s) = \sum_{j=1}^{M_i} w_{i,j} a_j, \quad a_j \sim \pi_j(s) \tag{2}$$

As the ultimate goal of demonstration modeling is to recover the demonstrated behavior, we optimize the linear weights, $\vec{w}_i$, to minimize divergence between the trajectory induced by the mixture policy and the demonstration, illustrated in Equation 3.

$$\underset{\vec{w}_i}{\text{minimize}}\, \mathbb{E}_{\tau \sim \pi_{\vec{w}_i}} \left[\text{Div}(\tau_i, \tau)\right] \tag{3}$$

Specifically, we choose Kullback-Leibler divergence (KL-divergence) [32] on the state marginal distributions of trajectories in our implementation. We estimate the state distribution within a trajectory by the kernel density estimator [33]. More details can be found in supplementary.

Since the trajectory generation process is non-differentiable, we seek a non-gradient-based optimizer to solve the optimization problem. Specifically, FLAIR utilizes a naïve, random optimization method; it generates random weight vectors $\vec{w}_i$, evaluates Equation 3, and chooses the weight that achieves the minimization. Empirically, we find random optimization outperforms various other optimization methods (see the supplementary for a detailed comparison) for FLAIR.

4.3 Between-Class Discrimination

Although MSRD distills the task reward from various demonstrated strategies, MSRD does not encourage the strategy rewards to encode distinct strategic preferences. MSRD also requires access to ground-truth strategy labels for all demonstrations, which limits scalability. In order to increase the strategy reward's discriminability between different strategies, we propose a novel learning objective named *Between-Class Discrimination* (BCD). BCD enforces the strategy reward to correctly discriminate mixture demonstrations from the pure demonstration: If demonstration τ_i has weight $w_{i,j}$ on strategy j (as identified in *Policy Mixture*), we could view the probability that τ_i happens under the strategy reward, $R_{\text{S-}i}$, should be $w_{i,j}$ proportion of the probability of the pure demonstration, τ_{m_j}. This property can be exploited to enforce a structure on the reward given to the

Table 1: This table shows learned policy metrics between AIRL, MSRD, and FLAIR. The higher environment returns / lower estimated KL divergence, the better.

Domains	Inverted Pendulum			Lunar Lander			Bipedal Walker		
Methods	AIRL	MSRD	FLAIR	AIRL	MSRD	FLAIR	AIRL	MSRD	FLAIR
Environment Returns	-172.7	-166.4	$\mathbf{-38.5}^{**}$	-7418.1	-9895.3	$\mathbf{-6346.6}^{*}$	-30637.2	-74166.0	$\mathbf{-6144.8}^{**}$
Estimated KL Divergence	4.08	7.67	$\mathbf{4.01}^{**}$	72.0	70.9	$\mathbf{67.2}^{**}$	13.0	32.6	$\mathbf{12.5}^{**}$

* Significance of $p < 0.05$
** Significance of $p < 0.01$

pure-demonstration, τ_{m_j}, and mixture-demonstration τ_i, as per Lemma 1. A proof is provided in the supplementary.

Lemma 1. *Under the maximum entropy principal,*

$$w_{i,j} = \frac{P(\tau_i; S\text{-}j)}{P(\tau_{m_j}; S\text{-}j)} = \frac{e^{\eta R_{S\text{-}j}(\tau_i)}}{e^{\eta R_{S\text{-}j}(\tau_{m_j})}}$$

Thus, we enforce the relationship of strategy rewards, S-j, evaluated on pure strategy demonstration, τ_{m_j}, and mixture strategy demonstration, τ_i with mixture weight $w_{i,j}$, as shown in Equation 4.

$$L_{\text{BCD}}(\theta^{\text{S-}j}) = \sum_{i=1}^{n} \left(e^{\eta_{\theta_{\text{S-}j}}(\tau_i)} - w_{i,j} e^{\eta_{\theta_{\text{S-}j}}(\tau_{m_j})} \right)^2 \tag{4}$$

An extreme case of BCD loss is when τ_i is the pure demonstration for another strategy, k (i.e., $m_k = i$). In this case, $w_{i,j} = 0$ (as τ_i is purely on strategy k), and Equation 4 degenerates to encourage the strategy j's reward to give as low as possible reward to τ_i. In turn, strategy rewards gain better discrimination between different strategies, facilitating more robust strategy reward learning, and contributing to the success in lifelong learning.

5 Results

In this section, we show that FLAIR achieves *adaptability*, *efficiency*, and *scalability* in modeling heterogeneous demonstrations. We test FLAIR on three simulated continuous control environments in OpenAI Gym [34]: Inverted Pendulum (IP) [35], Lunar Lander (LL), and Bipedal Walker (BW) [36]. We generate a collection of heterogeneous demonstrations by jointly optimizing an environment and diversity reward with DIAYN [37]. For all experiments excluding the scalability study, we use ten demonstrations. We compare FLAIR with AIRL and MSRD by running three trials of each method. More experiment details and statistical test results are provided in the supplementary.

5.1 Adaptability

Q1: ***Can FLAIR's policy mixtures perform well at the task?*** From ten demonstrations, FLAIR created 6.3 ± 0.5 strategies (average and standard deviation across three trials) in IP, 5.3 ± 1.2 in LL, and 3.3 ± 0.5 in BW. FLAIR's learned policies including the policy mixtures are signficantly more successful at the task (row "Environment Returns" in Table 1), outperforming benchmarks in task performance with 77% higher returns in IP, 14% in LL, and 80% in BW than best baselines.

Q2: ***How closely does the policy recover the strategic preference?*** Qualitatively, we find that FLAIR learns policies and policy mixtures that closely resemble their respective strategies, visualized in policy renderings (videos available in supplementary). We further show that FLAIR is statistically significantly better in estimated KL divergence than AIRL (average 4% better) and MSRD (average 18% better), shown in row "Estimated KL Divergence" in Table 1, where KL divergence is evaluated between policy rollouts and demonstration state distributions.

Q3. ***How well does the task reward model the ground truth environment reward?*** We evaluate the learned task reward functions by calculating the correlation between estimated task rewards with ground-truth environment rewards. We construct a test dataset of 10,000 trajectories with multiple policies obtained during the "DIAYN+env reward" training. FLAIR's task reward achieves $r = 0.953$ in IP (shown in Figure 2), $r = 0.614$ in LP, and $r = 0.582$ in BW, with an average 18% higher correlation than best baselines and statistical significance compared with AIRL and MSRD.

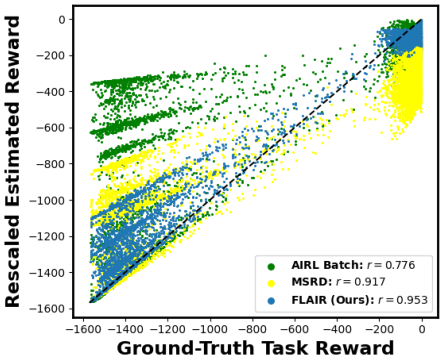


Figure 2: This figure shows the correlation between the estimated task reward with the ground truth task reward for Inverted Pendulum. Each dot is a trajectory. FLAIR achieves a higher task reward correlation than baselines.

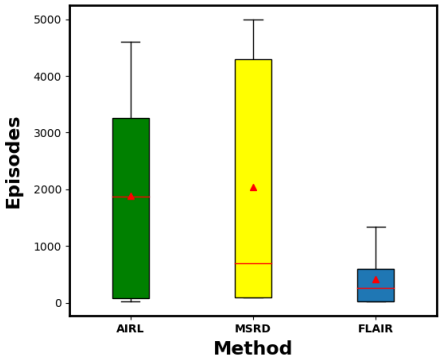


Figure 3: This figure compares the number of episodes needed for AIRL and MSRD to achieve the same Log Likelihood as FLAIR's mixture optimization. The red bar is the median and the red triangle represents the mean.

Q4. ***Can the learned strategy rewards discriminate between different strategies?*** We analyze the learned strategy rewards on all heterogeneous demonstrations, as shown in Figure 4 (right). We find that each strategy reward of FLAIR identifies the corresponding pure demonstration (Demonstrations 0-4,7) alongside the mixtures (Demonstrations 5-6, 8-9). In contrast, the strategy rewards learned without BCD (Figure 4, left) do not clearly distinguish between different strategies. This ablation study finds that FLAIR with BCD achieves 70% better environment returns, 61% better Log Likelihood, and 10% better KL divergence than FLAIR without BCD (additional metrics available in supplementary). The qualitative results in Figure 4 and quantitative results in supplementary together provide empirical evidence that FLAIR with BCD can train the strategy reward to better identify demonstrations with different strategic preferences.

5.2 Efficiency & Scalability

Q5. ***Can FLAIR's mixture optimization model demonstrations more efficiently than learning a new, separate policy?*** We study the number of episodes needed by FLAIR's mixture optimization and AIRL/MSRD policy training to achieve the same modeling performance of demonstrations. The result in Figure 3 demonstrates FLAIR requires 77% fewer episodes to achieve a high log likelihood of the demonstration relative to AIRL and 79% fewer episodes than MSRD. Three (out of ten) of AIRL's learned policies and four of MSRD's learned policies failed to reach the same performance as FLAIR, and those are left out in the bar plot. By being able to reuse learned policies through *policy mixtures*, FLAIR can capture new behaviors rapidly and explain the demonstration adeptly.

Q6. ***Can FLAIR's success continue in a larger-scale LfD problem?*** We generate 95 mixtures with randomized weights from 5 prototypical policies for a total of 100 unique demonstrations to test how well FLAIR scales. We train FLAIR sequentially on the 100 demonstrations and observe FLAIR learns a concise set of 17 strategies in IP, 10 in LL, and 6 in BW that capture the scope of behaviors while also achieving a consistently strong return for each task (Figure 5 and supplementary). We find FLAIR on average is able to maintain or even exceed its 10-demonstration performance when scaling up to 100 demonstrations.

5.3 Sensitivity Analysis

Q7. ***How sensitive is FLAIR's mixture optimization threshold?*** We study the classification skill of the mixture optimization threshold and find it has a strong ability to classify whether a demonstration should be included as a mixture or a new strategy. A Receiver Operating Characteristic (ROC) Analysis suggests FLAIR with thresholding realizes a high Area Under Curve (0.92) in the ROC Curve for IP; the specific choice of the threshold depends on the performance/efficiency tradeoff the user/application demands (ROC Curve and threshold selection methodology in the supplementary).

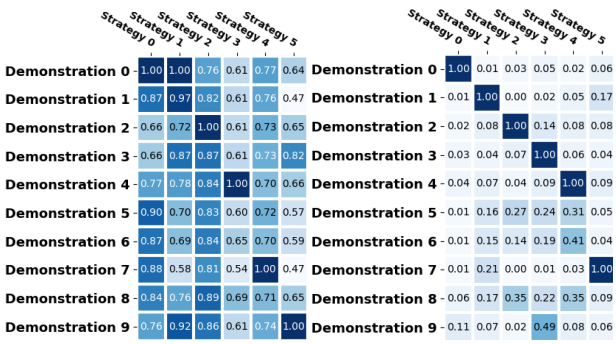


Figure 4: This figure depicts the normalized rewards on demonstrations based on the strategy reward output in Inverted Pendulum for FLAIR without BCD (left) and with BCD (right).

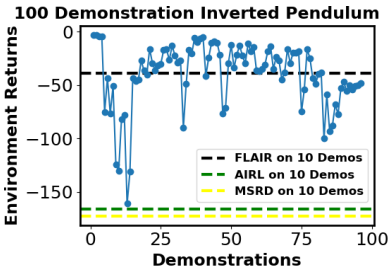


Figure 5: This figure plots the returns of FLAIR policies in the scalability experiment for 100 demonstrations, with AIRL, MSRD, and FLAIR 10-demonstration experiment performance as reference.

5.4 Discussion

We have presented several findings that show our algorithm, FLAIR, sets a new state-of-the-art in personalized LfD. Across several domains, FLAIR is able to achieve better demonstration recovery compared to the baselines of AIRL and MSRD. Not only can FLAIR more accurately infer the task reward (average 18% higher correlation) and associated policy (average 57% higher returns), but FLAIR is also able to perform policy inference with much fewer environmental interactions (average 78% fewer episodes). These characteristics also make FLAIR amenable to lifelong LfD, resulting in one of the first LfD frameworks that can handle sequential demonstrations without requiring retraining the entire model.

6 Real-World Robot Case Study: Table Tennis

We perform a real-world robot table tennis experiment where we utilize FLAIR's *policy mixtures* to model user demonstrations. We first collect demonstrations of four different table tennis strategies, push, slice, topspin, and lob, from human subjects via kinesthetic teaching. After training the four prototypical strategy policies, we assess how FLAIR's policy mixtures can succeed in new user demonstration modeling. A frame-by-frame illustration of an example policy mixture (slice + topspin) is in Figure 1(b). We quantitatively evaluate the fitness of the policy mixtures with respect to user preferences by a questionnaire, where we calculate a strategy score produced via a Likert scale. We compare the strategy scores of the best policy mixture identified by FLAIR, an average strategy, and an adversarially optimized policy mixture (i.e., maximize the KL divergence between the rollout and the demonstration). The results show alignment between strategy score given by the user and the ranking given by FLAIR, demonstrating FLAIR's ability to optimize policy mixtures that fit user's preferences. More details for the results are available in the supplementary.

7 Conclusion, Limitations, & Future Work

In this paper, we present FLAIR, a fast lifelong adaptive LfD framework. In benchmarks against AIRL and MSRD, we demonstrate FLAIR's *adaptability* to novel personal preferences and *efficiency* by utilizing policy mixtures. We also illustrate FLAIR's *scalability* in how it learns a concise set of strategies to solve the problem of modeling a large number of demonstrations.

The limitations of FLAIR include: 1) if the initial demonstrations are not representative diverse strategies, the learned task reward might be biased, and the ability to effectively model a large number of demonstrations (expressivity of the policy mixtures) might be impacted; 2) FLAIR inherits the property from AIRL that the learned rewards are non-stationary, hence it could suffer from catastrophic forgetting. In future work, we will explore remedies for these challenges, such as pre-training FLAIR with representative prototypes before deployment with end-users, and studying how to recover a minimally spanning strategy set that could explain all demonstrations.

References

[1] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[4] L. Matignon, G. J. Laurent, and N. Le Fort-Piat. Reward function and initial values: Better choices for accelerated goal-directed reinforcement learning. In *Artificial Neural Networks – ICANN 2006*, pages 840–849, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[5] S. Schaal. Learning from demonstration. In M. C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1997. URL https://proceedings.neurips.cc/paper/1996/file/68d13cf26c4b4f4f932e3eff990093ba-Paper.pdf.

[6] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 189–196. IEEE, 2015.

[7] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, Dec. 2014. doi:10.1609/aimag.v35i4.2513. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2513.

[8] E. F. Morales and C. Sammut. Learning to fly by combining reinforcement learning with behavioural cloning. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 76, 2004.

[9] I. Leite, C. Martinho, and A. Paiva. Social robots for long-term interaction: A survey. *International Journal of Social Robotics*, 5(2):291–308, Apr 2013. ISSN 1875-4805. doi:10.1007/s12369-013-0178-y. URL https://doi.org/10.1007/s12369-013-0178-y.

[10] L. Chen, R. R. Paleja, M. Ghuy, and M. C. Gombolay. Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*, 2020.

[11] K. Mülling, J. Kober, O. Kroemer, and J. Peters. Learning to select and generalize striking movements in robot table tennis. *Proceedings of the International Journal of Robotics Research (IJRR)*, 32(3):263–279, 2013.

[12] K. Muelling, A. Boularias, B. Mohler, B. Schölkopf, and J. Peters. Learning strategies in table tennis using inverse reinforcement learning. *Biological cybernetics*, 108(5):603–619, 2014.

[13] L. Chen, R. Paleja, and M. Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *Proceedings of Conference on Robot Learning (CoRL)*, 2020.

[14] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent advances in robot learning from demonstration. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 2020.

[15] A. Chella, H. Dindo, and I. Infantino. A cognitive framework for imitation learning. *Robotics and Autonomous Systems*, 54(5):403–408, 2006. ISSN 0921-8890. doi:https://doi.org/10.1016/j.robot.2006.01.008. URL https://www.sciencedirect.com/science/article/pii/S0921889006000200. The Social Mechanisms of Robot Programming from Demonstration.

[16] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[17] P. de Haan, D. Jayaraman, and S. Levine. Causal confusion in imitation learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/947018640bf36a2bb609d3557a285329-Paper.pdf.

[18] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[19] N. D. Daw and P. Dayan. The algorithmic anatomy of model-based evaluation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1655):20130478, 2014.

[20] A. Y. Ng, S. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[21] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. ACM, 2004.

[22] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, page 2586–2591. Morgan Kaufmann Publishers Inc., 2007.

[23] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the National Conference on Artificial intelligence (AAAI)*, pages 1433–1438, 2008.

[24] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, Carnegie Mellon University, 2010.

[25] M. Babes-Vroman, V. Marivate, K. Subramanian, and M. Littman. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 897–904, 01 2011.

[26] G. Ramponi, A. Likmeta, A. M. Metelli, A. Tirinzoni, and M. Restelli. Truly batch model-free inverse reinforcement learning about multiple intentions. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2359–2369. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/ramponi20a.html.

[27] J. Almingol, L. Montesano, and M. Lopes. Learning multiple behaviors from unlabeled demonstrations in a latent controller space. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 136–144, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[28] A. Bighashdel, P. Meletis, P. Jancura, and G. Dubbelman. Deep adaptive multi-intention inverse reinforcement learning. In N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, and J. A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 206–221, Cham, 2021. Springer International Publishing.

[29] J. Choi and K.-e. Kim. Nonparametric bayesian inverse reinforcement learning for multiple reward functions. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/140f6969d5213fd0ece03148e62e461e-Paper.pdf.

[30] S. Rajasekaran, J. Zhang, and J. Fu. Inverse reinforce learning with nonparametric behavior clustering. *arXiv preprint arXiv:1712.05514*, 2017.

[31] J. A. Mendez, S. Shivkumar, and E. Eaton. Lifelong inverse reinforcement learning. In *NeurIPS*, pages 4507–4518, 2018.

[32] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[33] L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Probl. Inf. Transm.*, 23(1-2):95–101, 1987.

[34] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016. URL http://arxiv.org/abs/1606.01540.

[35] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2012.

[36] C. Ericson. *Real-Time Collision Detection*. CRC Press, Inc., USA, 2004.

[37] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.