2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

A-EXP4: Online Social Policy Learning for Adaptive Robot-Pedestrian Interaction

Pengju Jin, Eshed Ohn-Bar, Kris Kitani, and Chieko Asakawa[1]

***Abstract*— We study self-supervised adaptation of a robot's policy for social interaction, *i.e.,* a policy for active communication with surrounding pedestrians through audio or visual signals. Inspired by the observation that humans continually adapt their behavior when interacting under varying social context, we propose Adaptive EXP4 (A-EXP4), a novel online learning algorithm for adapting the robot-pedestrian interaction policy. To address limitations of bandit algorithms in adaptation to unseen and highly dynamic scenarios, we employ a mixture model over the policy parameter space. Specifically, a Dirichlet Process Gaussian Mixture Model (DPMM) is used to cluster the parameters of sampled policies and maintain a mixture model over the clusters, hence effectively discovering policies that are suitable to the current environmental context in an unsupervised manner. Our simulated and real-world experiments demonstrate the feasibility of A-EXP4 in accommodating interaction with different types of pedestrians while jointly minimizing social disruption through the adaptation process. While the A-EXP4 formulation is kept general for application in a variety of domains requiring continual adaptation of a robot's policy, we specifically evaluate the performance of our algorithm using a suitcase-inspired assistive robotic platform. In this concrete assistive scenario, the algorithm observes how audio signals produced by the navigational system affect the behavior of pedestrians and adapts accordingly. Consequently, we find A-EXP4 to effectively adapt the interaction policy for gently clearing a navigation path in crowded settings, resulting in significant reduction in empirical regret compared to the EXP4 baseline.**

I. Introduction

Humans are able to *actively change the behavior of other pedestrians* in order to achieve a variety of everyday tasks. For instance, consider the task of safe social navigation, *e.g.,* city driving or maneuvering in a crowded airport environment. In such scenarios it may not be possible to maneuver a desired path due to a variety of reasons, including the complexity of the state space, the physical constraints imposed by the environment, or the presence of an unaware pedestrians ahead. Communicating with verbal or non-verbal cues to surrounding pedestrians, who may be inattentive, becomes a crucial ability for effective and pleasant navigation. As robotic agents are making their way from labs and controlled environments into the real-world and becoming more pervasive, they are more likely to encounter such scenarios in a variety of domains. Inspired by this observation, we explore the possibility of endowing the robotic system with the ability to interact with its surroundings in order to communicate with surrounding pedestrians and clear a path.

[1]Authors are affiliated with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {pengjuj, eohnbar, kkitani, chiekoa}@andrew.cmu.edu. Eshed Ohn-Bar is currently at the Max Planck Institute for Intelligent Systems.

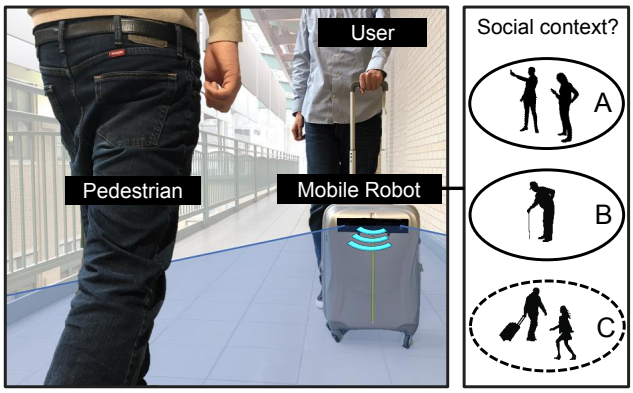

Fig. 1: We develop Adaptive EXP4 (A-EXP4), an online learning algorithm that enables assistive navigation systems to actively communicate with surrounding pedestrians, *e.g.,* through an audio signal, in a more socially acceptable manner. The system learns to automatically identify and adapt to previously unseen contexts, *e.g.,* a novel observed pedestrian type shown under context C in the figure. By continuously adapting to the current situational context in a self-supervised manner, the proposed algorithm enables more effective communication with surrounding pedestrians as well as navigation in very challenging social environments, such as indoor navigation in crowded environments.

However, learning a robot policy for communicating with surrounding pedestrians in real-world environments can be challenging. First, the interaction policy should be socially appropriate and minimally disruptive to avoid uncomfortable behavior. Second, modeling the effect of the interactions is a difficult task, as it is not always clear when and what signal should be broadcast to cause pedestrian behavior to change. One of the key reasons for why the task is challenging is that the reaction of nearby pedestrians is not consistent and can change over time based on the situational context. That is, given the same signal, the behavior of nearby pedestrians can change due to changes in the environment, changes in the configuration of pedestrians, or changes in the unobserved internal state of the pedestrian. This means that the effective state space of the problem is indeed very large and that the distribution over the state space can change over time.

To address these challenges in learning an effective robot-pedestrian interaction policy, we propose a novel online learning algorithm for modulation of the interaction policy, *i.e.,* as observed in human behavior in varying situational and social context [1], [2]. Our proposed reinforcement

978-1-7281-4004-9/19/$31.00 ©2019 IEEE

learning framework continually updates its interaction policy to adapt to the changing nature of the environment and the dynamic state distribution. Based on the well-known EXP4 algorithm [3], [4], our work enables the algorithm to search and generate new and diverse expert policies by utilizing a Dirichlet Process Gaussian Mixture Model (DPMM) over the space of policies. Hence, our approach allows for continuous exploration and learning of new interaction policies under changing and novel situational context.

We study the performance of our algorithm in the context of pedestrian-robot interaction for path clearing during robot navigation. To evaluate the algorithm, we employ two environments, one simulated and one real. The simulated environment allows us to extensively evaluate the algorithm while directly varying pedestrian types and behavior (*i.e.,* social context) over time. Based on insights from the experimental analysis in simulation with diverse pedestrians, we also implement and test the A-EXP4 algorithm in the real-world with real pedestrians. We evaluate the real-world performance of our algorithm using a suitcase-shaped robot as an assistive navigation system, *e.g.,* for a blind person navigating in dynamic and crowded areas (Figure 1). The robot assumes the form of a suitcase because we envision that such a system will be used in airports and travel scenarios. In this assistive scenario, the A-EXP4 algorithm observes how audio signals produced by the navigational system affect the behavior of surrounding pedestrians and adapts accordingly. Consequently, we find A-EXP4 to effectively adapt the interaction policy for gently clearing a navigation path in crowded settings.

II. Related Work

Our work leverages previous research in contextual bandits, online learning, and human-robot interaction to develop an adaptive algorithm for robot-pedestrian interaction, *i.e.,* appropriately modulating the interaction policy under varying pedestrian behavior and context during robot navigation.

Reinforcement and policy learning. In this work, the goal is to learn a control policy from a set of input observations about surrounding pedestrians. We formulate the task of policy learning for pedestrian-robot interaction in a Reinforcement Learning framework. The performance of the policy is based on a reward function that measures collision avoidance and social disturbance. There are many possible methods to optimize for such a policy, including value-based, policy-based [5]–[7], model-based [8], [9], and model-free [10]. We pursue a policy-based approach as it was previously shown to have several benefits, including faster learning and scalability to the dimensions of the observations [10], [11]. A related study to ours employed a multi-armed bandit model for grasp planning [12]. However, a fundamental assumption in standard reinforcement learning approach is that the underlying Markov Decision Process (MDP) remains constant over time, yet this is rarely the case in assistant robots where the system needs to be used in varying scenes. Inspired by model-free policy search learning techniques that utilize sampling-based inference [13]–[15], we propose to a similar mechanism with a DPMM [16], [17] in the context of robot-pedestrian interaction. Our optimization is done in a model-free manner, as it is often difficult to capture an accurate model of different types of pedestrians under different contexts [18]–[25]. Nonetheless, the recent study of Krishnan *et.al.* [26] in unsupervised transition state clustering suggests that Bayesian non-parametrics can be incorporated into model-based approaches for policy learning as well.

Contextual bandits and online learning. We envision a system that continuously learns appropriate policies as part of life-long learning process. To deal with the dynamic state distribution, we leverage ideas from contextual bandits problems under adversarial settings. The primary bandit algorithm for this case is EXP4, a no-regret algorithm proven to perform well under adversarial circumstances [3]. Due to its performance, various follow-up algorithms have been proposed to modify EXP4 and improve its regret bounds [4], [27]–[29]. In contrast to classical bandit problems, we do not assume that the set of 'arms' (policies) is static but instead attempt to learn many policies over time. Our main insight is to maintain a (potentially infinite) number of bandit arms (policies) by utilizing a DPMM over the space of policies. Hence, the resulting algorithm is a novel variation of the EXP4 algorithm that can better adapt to changing environments. We empirically validate that we are able to minimize the regret of the proposed A-EXP4 online learning algorithm even with the DPMM for clustering existing expert policies and finding new expert policies.

Robot-pedestrian interaction. One particular problem tackled in our work is effective pedestrian interaction and avoidance. This problem is relevant to a variety of domains requiring robot-pedestrian interaction [19], [30]. There has been significant interest in the research community for building accurate models to predict the trajectories of pedestrians [18], [20]–[22]. In such settings, prediction models can be applied within the planning loop in order to yield to pedestrians. However, depending on the operation space and the behavior of surrounding pedestrians, this is not always appropriate. Some examples include scenarios where the physical constraints imposed by the environment restrict the path of the robot or the presence of an inattentive pedestrian along the path. Specifically to our application of navigation in crowded areas, humans often interact with surrounding pedestrians verbally through "excuse me" and other forms of communication. Once the robotic system is endowed with the ability to interact and communicate to its surroundings, the problem of social context immediately arises. Hence, we wish to learn a set of appropriate actions that allow the pedestrian to move away from the robot instead of yielding in order to (gently) clear a path when needed.

III. Approach

The goal of this work is to develop an algorithmic framework that allows a navigational robot to naturally signal intent to pass, to a wide range of pedestrians along a navigational path, in order to avoid possible collisions.

We model the interaction of the navigational robot (agent) with nearby pedestrians (environment) as a Markov Decision Process (MDP) and learn the best policy for signaling intent. Formally, we would like to learn a policy $\pi(a|s)$ which maps a state $\mathbf{s}$, *i.e.*, observation from on-board sensors of a nearby pedestrian, to a distribution over a set of actions $a \in A = \{a_1, a_2\}$, *i.e.*, whether or not to initiate a sound, such that total reward obtained by following the policy is maximized. To deal with changes in pedestrian behavior over time, we take an online learning approach that dynamically adjusts the weights over a large set of policies such that the most successful policies have the greatest weight. In the following section we describe our MDP formulation and the proposed online learning algorithm.

A. Robot-Pedestrian Interaction Model

In our MDP, the state space S consists of a set of observations of visible pedestrians and obstacles in the field of view of the navigation system, where each state is defined as

$$s = [p_1, v_1, \gamma_1, \ldots, p_L, v_L, \gamma_L] \tag{1}$$

For each pedestrian l, p_l is a triplet encoding of the 3D position, v_l is a triplet encoding of the 3D velocity, and γ_l is a double encoding the 2D bearing. In our implementation, we set $L = 4$ using the four closest pedestrians to the system.

The reward function $r(s, a)$ is composed of two components

$$r(s, a) = r_{ca}(s, a) + r_{sd}(s, a) \tag{2}$$

The first component r_{ca} is the *collision avoidance* term, which is zero when no pedestrians are within some collision radius (*e.g.,* 1.5 meters) and a very large negative value otherwise. The second component is the *social disruption* term r_{sd} which is zero when the sound is turned off and a small negative value when the sound is turned on. The collision avoidance reward term encourages the robot to alarm pedestrians who are too close to the system and the second social disruption reward term penalizes the robot for being overly disruptive.

B. Adaptive EXP4

Since the reactive behavior of nearby pedestrians can vary greatly over time, the robot-pedestrian interaction policy needs to able to quickly adapt to such changes. To address the dynamic nature of pedestrian behavior, we incrementally learn a large set of robot-pedestrian interaction policies to cover a wide range of pedestrian behavior. To this end, we develop an online algorithm which is able to select the most appropriate policy by maintaining and adapting weights over all polices. In particular, we formulate the temporal adaptation problem as a contextual bandit algorithm.

In contrast to classical bandit problems, we do not assume that the set of 'arms' (policies) is static but instead attempt to learn many new policies over time. In the classical case, each bandit produces some reward from an underlying reward distribution. In the adversarial case, the reward distribution changes over time. EXP4 is a standard algorithm for solving the contextual adversarial bandit problem through the usage of expert advice. The algorithm uses a set of pre-trained experts to map the contextual information to an arm selection. At each time step, the agent receives a set of contextual information about all arms and it is allowed to choose one arm and claim a reward. The goal is to maximize the reward over a time horizon T. With each interaction, a set of weights is maintained for the experts and constantly adjusted depending on the result of the trial. The algorithm has been shown to be no-regret with respect to the best pre-trained expert when the number of arms is known. In our scenario, the number of arms grows over time and we develop an algorithm to select the best arm, as outlined in Algorithm 1.

Algorithm 1 Adaptive EXP4 (A-EXP4)

$\Pi = \{\pi_1, \ldots \pi_N\}$ is the set of all expert policies
Initalize $\mathbf{w} = \{w_i = 1\}$ for $i = 1 \ldots N$
for $t = 1, ..., T$ **do**
  $s_t \leftarrow$ `ObserveState()`
  $W \leftarrow \sum_{i=1}^{N} \mathbf{w}_i$
  $P \leftarrow \{p_j(t) = \mathbf{w}_j / W\}$
  $\pi_i^t \sim$ `Multinomial`$(\Pi; P)$
  explore $\sim$ `Bernoulli`(ϵ)
  if explore **then**
    ${\pi_i}'^t \sim$ `PolicySampler`$(\Pi, \mathbf{w})$
  end if
  $l^t \leftarrow$ `GetLoss()`
  if explore **then**
    $\Pi, \mathbf{w} \leftarrow$ `PolicyUpdate`$({\pi_i}'^t, l^t, \Pi, \mathbf{w})$
  else
    $\mathbf{w}_i^{t+1} \leftarrow \mathbf{w}_i^t e^{-\eta^t l^t}$
  end if
end for

Similarly to EXP4, A-EXP4 maintains a set of expert policies Π and a vector of weights $\mathbf{w}$ over each policy. In our experiments, the policies are represented using a linear policy approximator

$$\pi(s, a; \theta) = e^{\theta^\top \phi(s,a)} \tag{3}$$

where θ is a vector of learned policy parameters and $\phi(s, a)$ is a vector of state features. At each iteration, a policy π is sampled from a multinomial distribution according to the normalized weights. Instead of exclusively applying the policy π as in the classical contextual bandit algorithm, another policy, an *exploration* policy π', is sampled as well (described in the next section). The agent then applies the resulting policy and observes its loss. The loss function we use in our experiments is simply the one step reward described in Equation 2. Specifically, after taking an action based on the current policy π, we observe the reward by measuring the distance of the closest pedestrians to compute a normalized loss, $l = -r$ / `|Min Reward|` where `|Min Reward|` is the magnitude of the smallest possible reward (highest

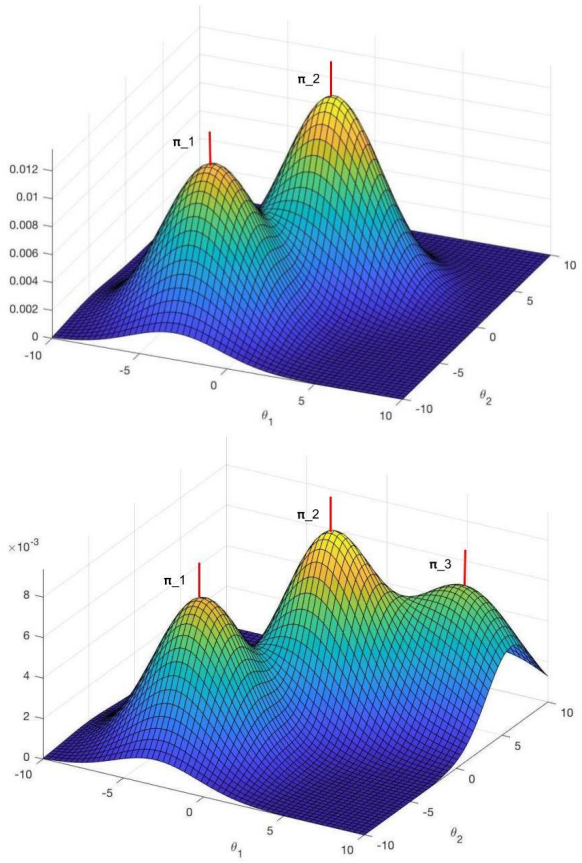

Fig. 2: Demonstration of a sampled expert policy encoded in the GMM. Each component of the GMM represents a single expert policy. In the **top figure**, there are two discovered expert policies. This representation naturally encodes a policy exploration strategy. As the environment changes over time, we will sample policies according to the GMM and the expert policy weights. With enough samples, a new locally optimal policy π_3 is added to the model as shown in the **bottom figure**.

penalty). If the selected policy is an exploration policy, it is passed to an online policy update algorithm described in Algorithm 3. Otherwise, the weights are updated according to the received loss using the traditional exponential gradient update. The policy sampling and learning process will be described in detail next.

C. Policy Sampler: Sampling Exploration Policies

To continually learn new policies, we sample new exploration policies by calling `PolicySampler` in Algorithm 2. The role of `PolicySampler` is to first estimate the distribution over the space of policy parameters Θ induced by the current set of policies Π using a Gaussian Mixture Model (GMM), and then the GMM is used to sample a new policy. The number of Gaussians in the mixture model is equivalent to $|\Pi|$. Example Gaussian mixture distribution induced by a set of two and three policies can be seen in Figure 2. The variance of each Gaussian mixture is set using the weight vector $\mathbf{w}$. For each mixture component i, $\sigma_i = L\mathbf{w}_i$ where L is a tunable scalar kept as a hyper-parameter for the amount of exploration (a high L value encourages more exploration). In our experiments, the value is set to 1.5. In this way, we are able to sample new exploration policies which are close to the highest reward yielding policies.

Algorithm 2 Sample New Exploration Policies

```
function PolicySampler(Π, w)
    G ← GenerateGMM(Π, w)
    π_i'^t ← RandomSampling(G)
    Return π_i'^t
end function
```

Algorithm 3 Incrementally Update a Policy

```
function PolicyUpdate(π(θ), l, Π, w)
    r ← exp(−l)
    Add policy-reward pair: D = D ∪ (θ, r)
    Compute updated policy parameters θ* with Eqn. 4
    if |D| > BufferSize then
        Add policy: Π = Π ∪ π(θ*)
        Π, w ← DirichletProcessMixture(Π, w)
        D = {∅}
    end if
    Return Π, w
end function
```

D. Policy Update: Incremental Learning from Exploration Policies

In order to continually learn new policies, we implement an *incremental* version of the PoWER algorithm [15] which can be used to search for new locally optimal policies using kernel density estimate over a set of sampled parameter-reward pairs S. As shown by Kober *et al.* in [15], the original PoWER algorithm relies on the idea that a way to safely learn new policies is to look at the convex combination of the sampled policies using importance sampling. In its simplest form, a new policy can be estimated using the following update:

$$\theta^* = \theta^* + \frac{\sum_{d=1}^{|\mathcal{D}|} r_d[\theta_d - \theta^*]}{\sum_{d=1}^{|\mathcal{D}|} r(\theta_d)} \tag{4}$$

where θ_d and r_d represent the parameters and reward of a sampled policy π_d, $\mathcal{D}$ is the set of sampled policy-reward pairs and θ^* is the parameters of a mean policy.

As described in our online implementation of PoWER `PolicyUpdate` (Algorithm 3), each newly sampled exploration policy π' and its resulting reward is added to a buffer of recent policies $\mathcal{D}$. The current buffered policy $\pi(\theta^*)$ is updated according Equation 4. Once the buffer reaches a specified size, we add the current buffered policy $\pi(\theta^*)$ into the set of all policies Π and clear the buffer. In this way, our incremental policy learning algorithm is able to constantly add new policies to the master set of policies Π.

As new policies are added to Π, it is possible that Π will contain policies which are very similar. To address this issue,

we estimate a Dirichlet Process mixture model (DPMM) that describes the current set of policies Π. In this way, we are able to effectively reshuffle the policies and indirectly bound the size of Π using the Dirichlet process concentration parameter α_0, similar to [17]. In the Bayesian DPMM, we use a Dirichlet Process as a prior for the mixture model distribution. Specifically, we add a probability distribution F_θ to the model, whose parameters θ are drawn from the DP with a base prior distribution G_0:

$$G \sim DP(\alpha, G_0), \;\; \theta_i \sim G, \;\; x_i \sim F_{\theta_i} \tag{5}$$

The result is an infinite model which can be generated as the limit of a finite process. When a set of n points $\{y_1, y_2, \ldots, y_n\}$ are given and assumed to be distributed according to a DPMM, the posterior probability of any given point y_i belongs to a cluster X_i can be computed using

$$P(X_i = x | X_{-i}, y_i, \theta^*) = \frac{N_{-i,c}}{N - 1 + \alpha} F(y_i, \theta_x^*) \tag{6}$$

$$P(X_i \neq x_j | X_{-i}, y_i, \theta^*) = \frac{\alpha}{N - 1 + \alpha} \int F(y_i, \theta_x^*) dG_0(\theta^*) \tag{7}$$

where x is current existing clusters, X_{-i} are the previous cluster assignments, $N_{-i,c}$ is the cluster's count, and θ^* being the parameter vector associated with the particular cluster.

Estimating the exact posterior of the DPMM requires computing complex integrals over the infinite DP. Approximation algorithms, such as Gibbs sampling [31] and variational inference [32] have been proposed for efficient inference. In our algorithm, we set F to be a Gaussian distribution so that the resulting model is an infinite Gaussian mixture model. The algorithm is implemented efficiently based on [33], and the inference step can be done quickly online.

Note that the algorithm described above continuously inserts new expert policies to the set once they are discovered. However, the algorithm tends to be inherently optimistic with novel experts and this could lead to sub-optimal behavior. Moreover, the iterative discovery process heavily depends on the quality of the initial experts, such that if the first few discovered policies are sub-optimal globally it becomes difficult to sample good policy in the long run. To discourage the algorithm from sampling low performance policies and eliminate policies with low weights (*i.e.,* low returns), we modify the posterior probability in Equations 6 and 7 to be weighted by the weights of the expert policies belonging to that cluster. In practice, we find this to effectively limit the number of expert components because most expert policies will have low weights after sufficient iterations of online evaluation.

IV. Experimental Settings

We analyze the performance of A-EXP4 for pedestrian-robot interaction during robot navigation. For our main experiment, we employ a simulated environment. The environment allows us to directly control the situational context for generating a meaningful comparison between A-EXP4 and the standard EXP4 baseline. We use the simulation to demonstrate that our adaptive algorithm has superior performances and empirically low regret. Particularly, we show that by reshuffling expert policies with DPMM, we can significantly reduce performance variance and lower the regret in adaptive scenarios. Moreover, we implement the system on a real-world mobile platform. While we have no direct control over the underlying pedestrian types and behavior in this case, we use this experiment to illustrate overall pedestrian behavior as a result of employing the audio communication strategy generated by the proposed online learner.

We first constructed a simulation based on the open sourced PedSim package [34], a flexible, light-weight pedestrian simulation engine. In PedSim, the pedestrians are simulated as particles and their movements are computed based on social forces, *e.g.,* the distance from obstacles, other pedestrians. In order to make the simulation more suitable for our problem setup, we add a robot agent that overwrites the social force model and always follows its trajectory without yielding. The robot agent is given the ability to engage an audio signal which forces the pedestrian to yield to the robot. Moreover, we extend the social force dynamic model for pedestrian behavior with additional attributes such as velocity, avoidance radius, awareness radius, and awareness level. These additional attributes influence how the pedestrians behave around the robot while allowing us to create varying pedestrian types and social context. For example, a high level of awareness makes the pedestrian more likely to move away from the robot when it is far away. In contrast, a low level of awareness makes the pedestrian much more likely to collide with other pedestrians and the robot.

The goal of our algorithm is to adapt to variations in pedestrian's reactive behavior. Ideally, our online learner can perform as well on a new pedestrian type in the long run as applying an optimal policy learned offline where the optimal policy is trained with the new pedestrian type beforehand. Therefore, the performance of our algorithm is measured with the notion of regert as it is commonly used in online learning literature. Formally, the regret at time t, R_t, is defined as

$$R_t = \sum_{i=0}^{t} l_t(a_t; \theta_t) - \min_{\theta^*} \sum_{i=0}^{t} l(a_t; \theta^*) \tag{8}$$

which is the difference of accumulated loss (computed over the observed sequence of examples) between the performance of the online algorithm, *e.g.* A-EXP4, and the hindsight optimal model, *i.e.* a policy trained with the new pedestrian type, over time.

During the evaluation, we stochastically generate a new pedestrian behavior type by randomly sampling values for each pedestrian attribute, *e.g.*, pedestrians with medium speed and low values for awareness. To analyze our online algorithm, we trained an optimal policy π^* on the new pedestrian type using the same offline policy gradient algorithm to convergence. For the baseline, we will evaluate against the standard EXP4 algorithm which maintains the expert

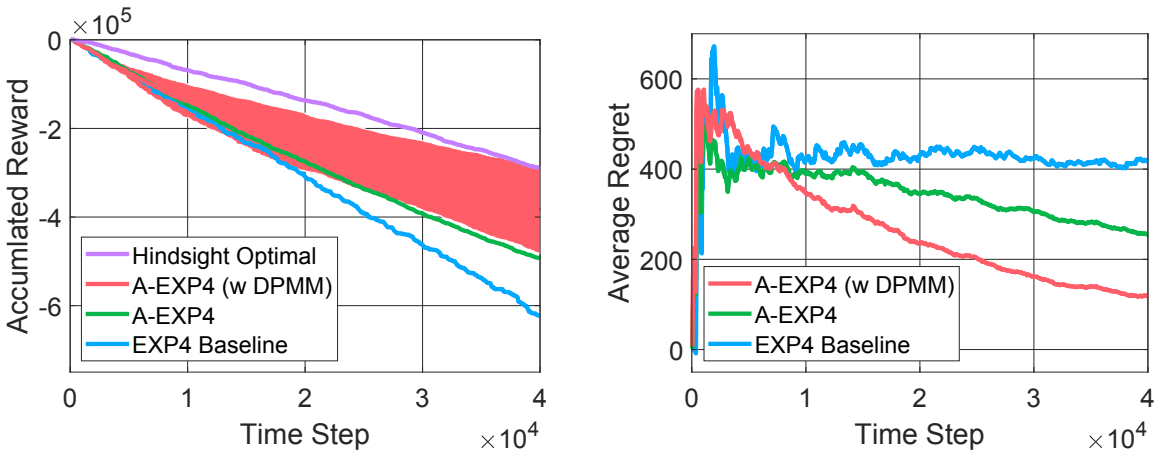

Fig. 3: (a) Average accumulated reward (higher is better) and (b) average empirical regret (lower is better) on a stochastic scene where pedestrian types are randomly generated over time. Results are shown for two A-EXP4 variants, with and without using the Drichlet Process Mixture Model (DPMM) to cluster the experts and remove ones with low weight.

policies statically. Each experiment is repeated 50 trials and the overall average accumulated reward and regret are recorded. For the simulation experiments, we initially trained two expert policies $\{\pi_1, \pi_2\}$ with offline policy gradient for two different sets of pedestrian attributes (slow and high awareness, fast and medium awareness).

Real-world implementation. To better understand the feasibility of using A-EXP4 in real-world robot-pedestrian communication and interaction, we implemented the online learning algorithm onto our suitcase-inspired, non-conspicuous platform. The motivation for such a platform is based on the fact that state-of-the-art assistive navigation systems (*e.g.,* for people with visual impairment [35]–[41]) are often introspective and only give navigational directions to the user. In many cases, they do not consider the ability of the system to actively change the behavior of other pedestrians to help lead a person successfully over a navigation path [42], [43]. Moreover, the design of many systems often assume or leverage the fact that other pedestrians will yield to the navigational system without any prompting. While this may be true for large robotic navigational platforms that are easy to spot (albeit, pedestrians in the environment may still be inattentive), this is does not generally hold for wearable, hand-held, or minimal robotic systems which are much less conspicuous. Hence, enabling assistive navigation in crowded environments, such as airports or malls, is challenging.

The platform (visualized in Figure 1), is equipped with a Microsoft Kinect 2, a small computer, and a speaker for audio interaction. The Kinect is used to extract scene information including obstacles positions, poses, and velocity of incoming pedestrians. We utilized the skeleton tracker and combine it with a standard Kalman filter [44] to estimate the full pedestrian trajectory and state information.

V. Experimental Analysis

Given the experimental setup of generating different pedestrian types over time, we run the baseline EXP4, as well as A-EXP4 with and without the DPMM, *i.e.,* the clustering and removal of policies step. The aggregated results over running the 50 trials (each for 40,000 time steps) are shown in Figure 3.

We can see how the full A-EXP4 algorithm significantly outperforms the EXP4 algorithm in average accumulated reward, by up to 40% towards the end of the experiments. Moreover, several observations can be made about the ability of these online learning algorithms to adapt to new context. For instance, Figure 3(a) demonstrates how A-EXP4 performs similarly to EXP-4 during the onset of the experiment. This is expected, as both algorithms employ the existing expert policies and shift the weights to the expert that performs best. However, as more exploration policies are sampled over time, A-EXP4's performance is shown to significantly improve as new policies are inserted into the expert set. In fact, comparing the three algorithms in Figure 3(a) shows a remarkable adaptation when compared to the hindsight optimal policy. The variance of A-EXP4 with the DPMM depicts how even in diverse settings, the method generally produces a policy as good or better as the A-EXP4 without the DPMM and the EXP4 baseline.

Figure 3(b) shows the average regret of the algorithms, where the average regret of A-EXP4 is shown to decrease steadily towards zero. In contrast, the standard EXP4 algorithm maintains a large average regret because neither pre-trained expert policies were effective against the new pedestrian types introduced by the simulator. When evaluation A-EXP4 without DPMM reshuffling of the expert policies, the performance is worse because the set of expert policies becomes unbounded. This hurts performance because poorly performing policies will never be removed from the set. Therefore, as the expert policies accumulate, it becomes

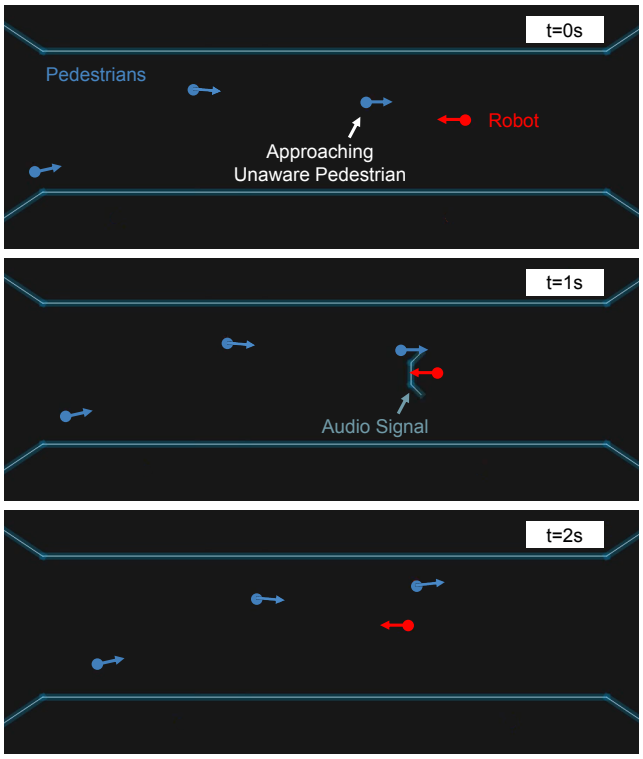

Fig. 4: Example interaction sequence in navigation with low-awareness pedestrians context.

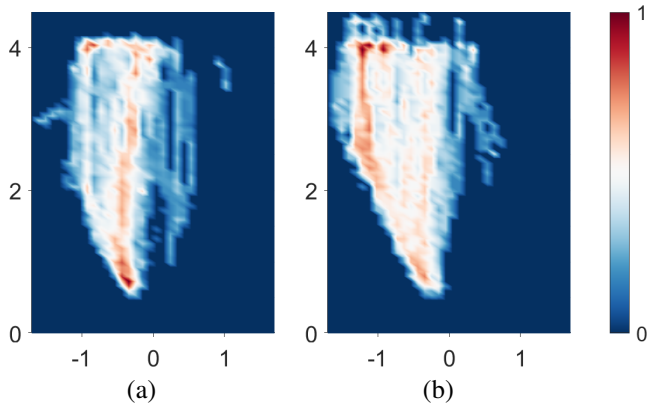

Fig. 5: Top-down view visualization of the normalized spatial distribution (x-axis and y-axis units are meters) of pedestrian trajectories with the real-world implementation of the (a) baseline policy learning algorithm and (b) A-EXP4 for audio signal generation. For meaningful comparison, both policies are tested in the same physical environment. Camera origin is at (0,0).

increasingly more difficult to sample better ones. As a result, this variant of the A-EXP4 algorithm largely depends on the quality of first few sampled policies. On the other hand, in the full A-EXP4, by clustering similar policies and removing ones with low weights we can ensure that the algorithm only samples near policies with high expected reward.

An example simulated interaction sequence is shown in Figure 4. In this scenario, the general context of the environment involves pedestrians with low awareness levels and medium speed. Hence, at times, the robot and a pedestrian may be on a collision or near-collision course. We can see how the A-EXP4 algorithm learns to effectively wait for producing the audio signal (*i.e.,* and incurring a negative reward), until the robot is at close proximity to the pedestrian. Then, triggering the audio signal successfully avoids a near-collision for this context.

Real-world analysis. The simulated environment allows us to directly measure the ability of our system to adapt to clearly defined, varying context over long periods of time. In real-world settings, performing a similar evaluation by defining the current context is more challenging. However, as the A-EXP4 algorithm adapts in a self-supervised manner, we can simply let the algorithm interact with the environment and observe the resulting interaction patterns on pedestrian motion over time.

We verify the analysis in simulation with a similarly designed real-world experiment. We tested the system on a straight forward route through a long hallway to demonstrate the overall effect of our approach on the incoming pedestrians with respect to the navigation task. In the experiment, we generally maintain our forward path without yielding, as in the simulated experiment. We then allowed the learner to iterate over the observed pedestrian trajectories. Overall, 191 pedestrian trajectories were collected for studying the behavior of A-EXP4.

To visualize overall pedestrian behavior using the two policies, we plot the top-down distribution of pedestrian locations in Figure 5. Generally, we observe how our proposed learner was able to generate policies based on pedestrian position and velocity, so that pedestrians avoided the platform much earlier and actively stayed away from its course. In contrast, the baseline policy is shown to be less effective at path clearing. Overall, there were only 111 instances of pedestrians being less than 1 meter from the system, compared to 296 with the baseline. Hence, this experiment shows the feasibility of using the proposed approach to interact with pedestrians in order to clear a path for navigation in crowded environments.

VI. Conclusion and Future Work

Learning a policy for robot-pedestrian interaction in diverse social environments is challenging. Towards this goal, we presented a principled approach for dealing with pedestrian behavior variations using a novel online learning algorithm. The proposed A-EXP4 approach relies on maintaining and adjusting weights over a set of expert policies. We extended a commonly used bandit algorithm to dynamically search new policies online and group them into the expert set using a Bayesian mixture model. Our experimental results show that A-EXP4 has better performance than using a static set of expert policies. Due to the general formulation, A-EXP4 can be applied to a variety of domains requiring adaptive and lifelong learning tasks. Although the position, velocity, and bearing attributes used in this study are essential

to representing pedestrians' state, in the future we would like to study the benefit of additional attributes that can be added in order to further explore how a social policy can be efficiently learned. Another interesting area for improvement would be exploring Thompson sampling [45] or other statistical sampling methods to improve the policy search efficiency in higher dimensions. Given that we developed a real-time adaptive system, an important next step would be further validation in real-world, large-scale robot-pedestrian interaction studies.

VII. ACKNOWLEDGMENTS

We greatly appreciate the assistance of Keita Higuchi and Edward Ahn with evaluation and data collection. This work was sponsored in part by NIDILRR (90DPGE0003), JST CREST (JPMJCR14E1) and NSF NRI (1637927).

REFERENCES

[1] H. C. Triandis, "The self and social behavior in differing cultural contexts," *Psychological review*, vol. 96, no. 3, p. 506, 1989.

[2] N. Mitsunaga, C. Smith, T. Kanda, H. Ishiguro, and N. Hagita, "Adapting robot behavior for human-robot interaction," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 911–916, 2008.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[4] V. Syrgkanis, A. Krishnamurthy, and R. Schapire, "Efficient algorithms for adversarial contextual learning," in *International Conference on Machine Learning*, 2016.

[5] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[6] F. Sehnke, C. Osendorfer, T. Rückstieß, A. Graves, J. Peters, and J. Schmidhuber, "Parameter-exploring policy gradients," *Neural Networks*, vol. 23, no. 4, pp. 551–559, 2010.

[7] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *International Conference on Intelligent Robots and Systems*, 2006.

[8] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning*, 2011.

[9] E. Ohn-Bar, K. Kitani, and C. Asakawa, "Personalized dynamics models for adaptive assistive navigation systems," *Conference on Robot Learning*, 2018.

[10] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[11] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[12] M. Laskey, J. Mahler, Z. McCarthy, F. T. Pokorny, S. Patil, J. Van Den Berg, D. Kragic, P. Abbeel, and K. Goldberg, "Multi-armed bandit models for 2D grasp planning with uncertainty," in *International Conference on Automation Science and Engineering*, 2015.

[13] N. Vlassis, M. Toussaint, G. Kontes, and S. Piperidis, "Learning model-free robot control by a monte carlo EM algorithm," *Autonomous Robots*, vol. 27, no. 2, pp. 123–130, 2009.

[14] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[15] J. Kober and J. R. Peters, "Policy search for motor primitives in robotics," in *Neural Information Processing Systems*, 2009, pp. 849–856.

[16] S. Calinon, A. Pervez, and D. G. Caldwell, "Multi-optima exploration with adaptive gaussian mixture model," in *International Conference on Development and Learning and on Epigenetic Robotics*, 2012.

[17] D. Bruno, S. Calinon, and D. G. Caldwell, "Bayesian nonparametric multi-optima policy search in reinforcement learning." in *AAAI*, 2013.

[18] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Computer Vision and Pattern Recognition*, 2016.

[19] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. Dragan, "Information gathering actions over human internal state," in *IROS*, 2016.

[20] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *IROS*, 2009.

[21] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *European Conference on Computer Vision*, 2012.

[22] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani, "Forecasting interactive dynamics of pedestrians with fictitious play," in *Computer Vision and Pattern Recognition*, 2017.

[23] L. Zeng and G. M. Bone, "Mobile robot collision avoidance in human environments," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 41, 2013.

[24] M. M. Almasri, A. M. Alajlan, and K. M. Elleithy, "Trajectory planning and collision avoidance algorithm for mobile robotics system," *Sensors*, vol. 16, no. 12, pp. 5021–5028, 2016.

[25] S. Hamasaki, Y. Tamura, A. Yamashita, and H. Asama, "Prediction of human's movement for collision avoidance of mobile robot," in *International Conference on Robotics and Biomimetics*, 2011.

[26] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," *International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1595–1618, 2017.

[27] H. B. McMahan and M. J. Streeter, "Tighter bounds for multi-armed bandits with expert advice." in *COLT*, 2009.

[28] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. Schapire, "Contextual bandit algorithms with supervised learning guarantees," in *AISTATS*, 2011.

[29] G. Neu, "Explore no more: Improved high-probability regret bounds for non-stochastic bandits," in *Neural Information Processing Systems*, 2015, pp. 3168–3176.

[30] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, 2016.

[31] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[32] D. M. Blei, M. I. Jordan, *et al.*, "Variational inference for Dirichlet process mixtures," *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[33] D. Steinberg, "An unsupervised approach to modelling visual data," *PhD Thesis*.

[34] Christian Gloor, "PEDSIM: Pedestrian crowd simuulation." [Online]. Available: http://pedsim.silmaril.org/

[35] A. Kulkarni, A. Wang, L. Urbina, A. Steinfeld, and B. Dias, "Robotic assistance in indoor navigation for people who are blind," in *HRI*, 2016.

[36] D. Sato, U. Oh, K. Naito, H. Takagi, K. Kitani, and C. Asakawa, "NavCog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment," in *ASSETS*, 2017.

[37] H. Kacorri, S. Mascetti, A. Gerino, D. Ahmetovic, H. Takagi, and C. Asakawa, "Supporting orientation of people with visual impairment: Analysis of large scale usage data," in *ASSETS*, 2016.

[38] S. Scheggi, M. Aggravi, F. Morbidi, and D. Prattichizzo, "Cooperative human-robot haptic navigation," in *ICRA*, 2014.

[39] D. Ahmetovic, C. Gleason, C. Ruan, K. M. Kitani, H. Takagi, and C. Asakawa, "NavCog: a navigational cognitive assistant for the blind." in *MobileHCI*, 2016.

[40] M. Murata, D. Ahmetovic, D. Sato, H. Takagi, K. M. Kitani, and C. Asakawa, "Smartphone-based indoor localization for blind navigation across building complexes," in *PerCom*, 2018.

[41] C. Ye, S. Hong, X. Qian, and W. Wu, "Co-robotic cane: A new robotic navigation aid for the visually impaired," *IEEE Systems, Man, and Cybernetics Magazine*, 2016.

[42] T.-K. Chuang *et al.*, "Deep trail-following robotic guide dog in pedestrian environments for people who are blind and visually impaired-learning from virtual and real worlds," in *ICRA*, 2018.

[43] J. Guerreiro, E. Ohn-Bar, D. Ahmetovic, K. Kitani, and C. Asakawa, "How context and user behavior affect indoor navigation assistance for blind people," in *W4A*, 2018.

[44] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[45] I. Osband and B. Van Roy, "Bootstrapped Thompson sampling and deep exploration," *arXiv preprint arXiv:1507.00300*, 2015.