Toward Robust Policy Summarization

ABSTRACT

AI agents are being developed to help people with high stakes decision-making processes from driving cars to prescribing drugs. It is therefore becoming increasingly important to develop "explainable AI" methods that help people understand the behavior of such agents. Summaries of agent policies can help human users anticipate agent behavior and facilitate more effective collaboration. Prior work has framed agent summarization as a machine teaching problem where examples of agent behavior are chosen to maximize reconstruction quality under the assumption that people do inverse reinforcement learning to construct a mental model of an agent's behavior from demonstrations. We compare summaries generated under this assumption to summaries generated under the assumption that people construct mental models through imitation learning. We show that for some domains, summaries extracted in a way that is optimized for certain reconstruction models are robust to mismatch between summary and reconstruction models, but that this is not the case in every domain. We further show in a human-subject study that people can anticipate the action of an agent using a summary optimized for their reconstruction method, but cannot do so using a summary optimized for another method. These results highlight the importance of assuming correct computational models for how humans extrapolate from a summary, suggesting human-in-the-loop approaches to summary extraction.

1 NOTES AND QUESTIONS

- Ike: Maybe the work of the user study is to provide evidence that the differences in the computational results correspond to differences in peoples' ability to reconstruct in at least 1 domain?

2 INTRODUCTION

AI agents are being developed to help people with high stakes decision-making processes from driving cars to prescribing drugs, but it can be challenging to understand both their limitations and their strengths. Adding human oversight to the development and deployment of AI agents can allow people to identify and correct errors that agent will make in the future, as well as to understand the useful insights the agent has learned. A rich body of research focuses on explaining a specific decision made by an agent to a human user (e.g. [5, 7, 13, 15]), but while this aspect of explaining agent behavior is important, it is not obvious how a user with no knowledge of the agent's behavior should choose which decisions to inspect. Providing a user with a global summary of the agent's behavior can help the user to better understand which of the agent's decisions to interrogate.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 2019, Montreal, Canada

https://doi.org/doi

The problem of policy summarization, or extracting these general policy summaries in the form of a small, informative collections of agent decisions has been approached in two ways. The first approach relies on heuristics for diversity or importance of states shown in the summary [2, 11]. The second approach assumes a computational model of how humans will generalize from the summaries provided, and uses this computational model to generate summaries [12]. The second approach is more principled, since it chooses states according to how well they can be used to generalize agent behavior (the stated goal), but introduces a key challenge: which computational model should we use to approximate how humans will generalize their understanding of the agent's behavior from the decisions show in the summary?

This is an urgent question in light of the cognitive science literature, which shows that users apply different computational models in different situations [10], switching between inverse reinforcement learning (IRL) and imitation learning (IL) depending on their familiarity with the domain [check this point]. This makes it non-obvious how to model the user during policy extraction–something which Huang *et al.* [12] does not account for since it only addresses variations on IRL. In this work, we explore the extent to which the choice of computational model used during the summary extraction process matters. We hypothesize that one of the following scenarios holds: the same set of decisions are informative regardless of the computational model; each model results in sets of decisions that are not informative to other choices of models; or some models will produce sets of decisions that are informative to other models and some will not. Which of these scenarios holds influences how much effort we should expend to ensure that the model used during summary extraction matches how people will use the summary to generalize their understanding of the agent's behavior.

We approach understanding the effect of the model used during summarization in the following two ways. (1) We run computational experiments in a range of domains (Gridworld, PAC-MAN and HIV treatment) exploring how effectively models (IRL-previously used and IL-novel) and sets of hyperparameters (discount factor for IRL and similarity metric for IL) can reconstruct the policy for a different choice of model during summary extraction. The insight is that, while we do not know which of these hypothesized models people will use in a given domain, we can rigorously explore how sensitive each model's ability to generalize the agent's behavior is to the choice of summary extraction model [is this the right way or should this be flipped?]. (2) We conduct a human-subject study assessing the ability of users to predict an agent's action based on summaries extracted using different models in [which domain?]. This allows us to determine whether the variations we found in the computational experiments correspond to real variations in peoples' ability to reconstruct the policy (in at least some cases).

Our results show that [matching the extraction model to the reconstruction model is generally important in both the computational and human-subjects studies (robust to choice of quality metric?)]. [Some additional more detailed result?]. Based on these

results, we argue that how to effectively match computational models used during policy extraction to models used by users to reconstruct agent behavior from the summary is an urgent area for future research. We suggest approaching it through a combination of user studies to determine commonalities in domains where different models are preferred, and human-in-the-loop approaches to choose the parameters of the models [if they matter].

3 RELATED WORK

Explaining Agent Decisions. [Todo: cite Miller somewhere in here] There exists a breadth of work on creating explainable agents. Most of these works [5, 7, 13, 15] focus on explaining a specific decision rather than providing a general understanding of an agent's strategy (as we do). For example, Dodson *et al.* [7] develop a system that generates natural language explanations for an action recommended by an MDP based agent while the user interacts with the policy. For these kinds of local explanations, Broekens *et al.* [5] conducts a user study with hand-coded trees of actions and subgoals to determine which of three explanation types is most useful: explaining an action by the goal it intends to achieve, explaining an action by its enabling condition, and explaining an action by the first action that follows it. They find that explaining an action via its parent goal is best, but that there is an interaction effect between the kind of action, and the best explanation. Closer to our work, Ramakrishnan *et al.* [18] formalizes the problem of detecting "blind spots", that is, situations in which an agent may choose the wrong action because it can't differentiate between two states that require different actions. Our work is more general in that we want to provide the user with a general mental model of the agent, including when it is performing well.

Summarizing Agent Policies via Trajectories. [Ike: maybe more detail on user studies in previous policy summarization work?] Our work falls into the category of policy summarization, in which the goal is to provide the human user with a mental model of how an agent will act across a range of scenarios [3]. To solve this problem, Amir and Amir [2] propose HIGHLIGHTS, an algorithm that generates a summary by extracting a given number of trajectories from simulations of the agent. The trajectories are chosen based on a notion of importance of states, which is defined by the difference in Q-values taking the best and worst action in a specific state. Huang *et al.* [12] formalize the problem of explaining robot behavior to humans as showing a collection of trajectories that allows an IRL agent (a proxy for the human user) to infer the underlying reward function. On a simulated driving task, they find that users perform best if the IRL agent uses an approximate planning algorithm that matches the cognitive limitations of humans. In Huang *et al.* [11], they extend that work to demonstrate that showing diverse, critical states that are helpful for debugging, and for motivating trust. Their human model assumes that states have the same action if they are within a pre-set distance of each other and then select clusters with a state-importance metric similar to Amir and Amir [2]. Our work extends these works in two important ways: First, we consider both IL as well as IRL-based proxies for how humans might generalize from a few trajectories, whereas prior works considered only IRL-based proxies or heuristics. Second, we consider a range of domains, some intuitive—like PAC-MAN—and others less intuitive—like managing HIV. This allows us to demonstrate that while some domains have summaries that are robust to misspecified user models, not all do.

Science of Good Summaries. A parallel literature has examined the science of good summaries from a psychology perspective. Dragan and Srinivasa [8] show that people can learn to anticipate agent actions if they see enough examples, but some agent behaviors cannot be fully anticipated even with many examples. Baker et al. [4] suggest that people use Bayesian inverse planning to infer others' goals based on observations of their actions. Gershman [10] describes how humans rely on a model-based and a model-free system that interact with each other and take control in different situations. Human model-based planning is more accurate when there is less data, with errors dominated by 'computational noise,' i.e. humans have bounded planning ability. As people gain more information about their environment, they switch to model-free systems, where errors are dominated by statistical noise, which comes from being unsure about the transitions, rewards and actions. Pilot et al. [17] discuss how, from a computational computational perspective, IRL is better than IL when long-term planning is required under covariate shift, but otherwise IL may be sufficient.

4 METHODS

Following Huang *et al.* [12], we formalize the problem of extracting a summary of an agent's behavior as an instance of machine teaching [22]. Specifically, given access to a simulator of the agent, we choose a set of state-action pairs $T = \langle\langle s_1, a_1\rangle, ..., \langle s_k, a_k\rangle\rangle$ to maximize the quality of a user's mental model of the agent's behavior derived from these examples. Since we do not know how people construct mental models of policies from examples, we explore two computational models: inverse reinforcement learning, where demonstrations are used to derive a reward function that is then used for planning, and imitation learning where the action is directly predicted at each state. In the following sections, we first review machine teaching in general and then discuss how we apply it to extract both IRL-based and IL-based summaries.

4.1 Learning Mechanism: Machine Teaching

Machine teaching aims to find a set of training examples that induces a known target model to learn a pre-specified source model [22]. In our setting, the agent's policy is the source model that we want to induce, and the target models that we consider are hypotheses for how human users build mental models of the agent's policy from examples of its behavior. We stress that we do not know how humans build mental models of a policy from demonstrations for a given domain [10], which motivates our inclusion of both IRL and IL as plausible hypotheses.

Formally, our problem will be to find the set of examples T of size $|T| = k$ that maximizes the quality measure ρ of the model induced by T under a specific target computational model M.

$$\max_{T \in \mathbb{T}} \rho(M(T), \pi^*)$$
$$\texttt{s.t.} |T| = k$$

where $\pi^*(s) = a$ is the agent's policy and $M(T)$ the target computational model's approximation of that policy. In our case, we define

the quality measure ρ as the accuracy with which the model is able to predict the agent's behavior in unseen states:

$$\rho(M(T), \pi^*) = \frac{1}{|S \setminus T|} \sum_{s \in S \setminus T} \mathbb{1}_{M(s)=\pi^*(s)} \quad (1)$$

We note that this measure differs from the one used by Huang *et al.* [12], where the summary aims to optimize the accuracy of the reward function, and from the one used by Amir and Amir [2], which optimizes for inclusion of "important" states. We argue that accuracy more readily captures ability to anticipate the behavior of an agent in a specific state, but these other metrics might be more beneficial for specific use cases (e.g., if the user's goal is to understand an agent's preferences, approximating the reward structure might be more important).

4.2 Inverse Reinforcement Learning Based Summary Extraction

Given a collection of trajectories, Inverse Reinforcement Learning (IRL) extracts a reward function such that the optimal policy with respect to those rewards matches the demonstrated behavior [16]. We use the Maximum Entropy IRL (Max-Ent) model [23] as a proxy of how people may extract such reward functions given a collection of trajectories. To generate the summaries we use the algorithmic teaching approach presented in [6] to extract a set of trajectories that optimally reconstruct the true policy. IRL-based extraction of summaries was also explored in [12].

Model of Human Extrapolation: Maximum Entropy IRL. The Max-Ent model in Ziebart *et al.* [23] is a model-based approach to IRL that formulates the problem of learning a policy from observed trajectories as optimizing a linear function mapping the features of each state to a reward value. Its goal is to match the *feature expectations* of the learned policy to those of the observed trajectories. These expectations are defined as

$$\mu_\pi^{(s,a)} = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \phi(s_t) | \pi, s_0 = s, a_0 = a] \quad (2)$$

There may be many reward functions resulting in feature expectations that match the observed trajectories; Max-Ent chooses one based on the maximum entropy principle. We note that Max-Ent is mathematically equivalent to the probabilistic reward-based model used for IRL-based summary extraction in Huang et al. [12]. Max-Ent assumes possible noise in expert demonstration, while the model from Huang et al. assumes noise in the reconstruction, resulting in the same computations.

The algorithm's objective is to find a reward function that maximizes the likelihood of the trajectories in the summary. It does this by using gradient descent to update the reward function based on the difference between the feature expectations of the observed trajectories and the feature expectations under the optimal policy of an MDP with the current rewards. See [23] for details.

Summary Extraction Method: Machine Teaching. Given our model for human extrapolation, we can now consider the task of optimizing which trajectories to include in the summary such that the user's reward model—as recovered via performing Max-Ent IRL—will be as accurate as possible. For this, we turn to [6], a machine teaching approach for IRL. The authors propose the SCOT algorithm, which selects the minimal set of demonstrations that allows the learner to obtain a reward function *behaviorally equivalent* to the optimal policy π^*. The behavioral equivalence class (BEC) of π^* is defined as the set of reward functions under which the policy is optimal. The BEC of π^* can be expressed by the intersection of halfspaces given by the following constraints:

$$w^T(\mu_\pi^{(s,a)} - \mu_\pi^{(s,b)}) \geq 0, \forall a \in \arg\max_{a' \in A} Q^*(s, a'), b \in A, s \in S \quad (3)$$

where $w \in \mathbb{R}^k$ are the reward weights, and $\mu_\pi^{(s,a)}$ are the expected feature counts as described above. The BEC of a demonstration containing only part of the state-action pairs from the optimal policy, is defined by the intersection of halfspaces for the demonstrated states and actions. The machine teaching algorithm seeks to find the smallest set of trajectories with halfspace constraints covering the constraints defined by π^*. It starts by computing the BEC of the optimal policy, creating a set of halfspace constraints. This set contains many non-binding and thus redundant constraints, which are removed using linear programming. Candidate trajectories of length L are extracted from the original policy, and halfspace constraints are computed for each trajectory.

In our case, we provide the algorithm with a budget k for the number of state-action pairs in the summary, and specify L (the length of each trajectory), such that the number of extracted trajectories is $\frac{k}{L}$. Since the algorithm greedily selects trajectories, we terminate it once the budget limit is reached. In cases where the non-redundant constraint set is covered with fewer than the budget specified for trajectories, the remaining ones are chosen randomly from the set of candidate trajectories.

4.3 Imitation Learning Based Summary Extraction

An alternative model for human extrapolation, rather than identifying rewards and planning against them, tries to directly mimic the agent. Imitation learning captures the notion that people may build mental models of agent behaviors that predict how the agent will act based on the actions it has taken in similar states, with no concept of reward or goal. We learn a function mapping from states directly to actions: $f : s \rightarrow a$. In our experiments, we use the Gaussian random field (GRF) model and the active learning algorithm described in Zhu *et al.* [21] to find the set of size k that maximizes accuracy on states not included in the summary. To our knowledge, IL-based summary extraction has not been previously studied.

Model of Human Extrapolation: Gaussian Random Field. The GRF model in Zhu *et al.* [21] represents data points—in our case, states—as vertices in a graph connected by edges weighted by their similarity, and makes predictions by propagating labels—in our case, actions—through the graph. The model is defined by an energy function $E(y)$ that weights the squared difference between the labels between each pair of nodes (i, j) according to their similarity, w_{ij}, where w is some kernel (in our experiments we consider RBF and polynomial kernels). The probability of the labels can be written as follows, where β is a tunable inverse temperature parameter (our

experiments use $\beta = 1$).

$$E(y) = \frac{1}{2}\sum_{i,j} w_{ij}(y(i) - y(j))^2$$

$$p(y) = \frac{1}{Z_\beta} exp(-\beta E(y))$$

Predictions are then made as follows:

$$f_u = -L_{uu}^{-1} L_{ul} f_l \tag{4}$$

where f are the labels, and $L = D - W$ is the combinatorial Laplacian matrix with W denoting the similarity between pairs of data points and $D = diag(\sum_j w_{ij})$. The indices u and l refer to unlabeled and labeled data points respectively, and sub-select the corresponding rows or columns (in that order) of the matrix. Points where $f_u(i) > 0.5$ are then labeled as positive, and the rest are labeled as negative. We extend this to the multiclass setting by doing one-vs-rest classification as suggested in Zhu *et al.* [20]. Conceptually, the GRF-based model of human extrapolation can be thought of as guessing an action for a new state based on the distribution of actions taken in similar states.

Summary Extraction Method: Active Learning. As with the IRL approach, given a model of human computation, we need to define a procedure for extracting a set of trajectories such that a human using a GRF-type computation will be able to recover the true policy. To do so, we use the active learning algorithm in Zhu *et al.* [21]. The algorithm implements the expected error reduction strategy, greedily choosing at each step to include the node label that minimizes the zero 0/1 loss on all unseen states:

$$x^*_{0/1} = \arg\min_x \sum_{k=1}^{K} P_\theta(y_k|x)\left(\sum_{u=1}^{U} 1 - P_{\theta^{+x,y_k}}(\hat{y}|x^{(u)})\right)$$

$$x^*_{0/1} = \arg\min_x \sum_{u=1}^{U} \mathbb{1}_{y^{(u)}=\arg\max_k P_{\theta^{+x,y}}(y_k|x^{(u)})}$$

where θ^{+x,y_k} is the model that has been retrained with the new tuple added into the training set [19]. In general, this scales with the number of classes, but since we can plug the labels in directly—given the policy, we can plug in y_k rather than $P_\theta(y_k|x)$—we can avoid the sum above and make the computation tractable. The algorithm operates by finding the held-out (unlabeled) data point that maximizes the accuracy on the remaining held-out points after retraining the model with the original training set plus that point. This is efficient because the GRF allows for efficient re-training.

5 COMPUTATIONAL EXPERIMENTS

Given that there are many plausible computational models of human extrapolation, our first goal was to investigate whether this multiplicity of models matters: For example, if we generate a summary assuming that the user will attempt to reconstruct the agent's policy using IRL, will the user's ability to reconstruct suffer if they happen to use IL? An answer to the negative would suggest that all of these models are robust to misspecification, whereas an answer to the positive would suggest that it will be important to identify the human's computational model in each particular situation.

We investigate this question by generating summaries of agent behavior in 3 domains: a random gridworld [6], PAC-MAN[1], and an HIV simulator [1], using a variety of hyper-parameter configurations for both IRL-based and IL-based models of human computation. We then compute how well the model is able to reconstruct the agent's policy given that summary. We refer to the computational model used to generate the summary as the "extraction model" and the computational model used to reconstruct the policy from the summary as the "reconstruction model". The extraction model represents our assumed model of human computation, and the reconstruction model allows us to simulate what would happen if people generalized using that model.

5.1 Empirical Methodology

Domains. We consider the following three domains. The domains are chosen to represent a variety of environments: gridworld is a static, navigational environment, whereas PAC-MAN is a dynamic, navigational environment (due to the moving ghost, ability to eat food). The HIV simulator represents a non-navigational, signal-based environment. While other works (e.g. [2, 12]) have focused on navigational domains, one of our goals was to see if there are differences across different kinds of domains.

- **Random Gridworld**: We use the 9x9 random grid world defined in [6]. The feature vector is an 8-dimensional indicator that determines the reward at each state. The agent's behavior is computed using value iteration with $\gamma = 0.95$.
- **PAC-MAN**: We use a 6x7 PAC-MAN grid with a single piece food pellet in the middle and a wall surrounding it on 3 sides [2]. A single ghost moves towards PAC-MAN deterministically, and PAC-MAN moves in the direction of the nearest food that does not result in a collision with the ghost. We derive 2 sets of features: one tailored for IL that includes the direction of the nearest food and an indicator for each action that specifies whether it will result in a collision with a wall or with the ghost; and one tailored for IRL that includes the distance to food, a binary indicator for whether PAC-MAN consumes food, and a binary indicator for whether PAC-MAN is consumed by a ghost.
- **HIV Simulator**: We use the HIV simulator described in [1] which includes 6 features corresponding to six cells in the blood, and 4 actions corresponding to activating or not activating two types of drugs. We generate a policy using fitted Q iteration as in [9] with a 0.05 initial state perturbation.

Variations of IL and IRL. Both extraction methods (active learning for IL and machine teaching for IRL) include hyperparameters. We tune several of these to explore the effect they have on the extracted summaries. For IL, we consider the kernel [RBF, polynomial], the length scale [0.01, 0.1, 1., 10., 100.] (some choices of length scale resulted in numerical issues for some domains; we did not include these in our results), and the degree for polynomial kernels [1, 2, 3]. For IRL, we consider the trajectory length [2, 3, 4] (we keep the summary size fixed for a given domain; this parameter affects the number of trajectories extracted). In the PAC-MAN domain, we

[1] We use the implementation from the Berkeley Intro to AI course: http://ai.berkeley.edu/project_overview.html

[2] http://ai.berkeley.edu/project_overview.html

also explore the choices of state representation using the IL and the IRL-specific feature sets described above.

Method Details. Each extraction method also had a number of other parameters that were held fixed and not tuned. The most important was the number of state-action pairs to include in the summary. For each domain, we chose a summary size (k) so that at least one variation of both IL and IRL had reasonable reconstruction accuracy for a summary extracted with that variation, and where increasing the summary size does not change the accuracy of the variations we consider markedly. For the random gridworld, we set $k = 48$, for PAC-MAN and HIV, we set $k = 24$.

IRL had several additional hyperparameters that needed to be specified for its planner: the discount factor γ, the rollout horizon, the learning rate and the number of iterations. We set $\gamma = 0.95$ for gridworld and $\gamma = 0.98$ for HIV to match the learned policy, and rollout horizon to 10 for random gridworld and 25 for HIV as a long time horizon; for PAC-MAN, we set $\gamma = 0.5$ and the rollout horizon to 3 to account for the more reactive behavior of the agent in avoiding ghosts. We used a learning rate of 1 for the random gridworld, 0.1 for PAC-MAN and 0.01 for the HIV simulator and ran 100 iterations, stopping if the rewards changed by less than 1e-5 between two consecutive iterations.

Finally, design choices were required to apply our methods to the HIV domain, which has a continuous state space. We extracted a batch of 5 episodes of 200 steps from the simulator to run the summary extraction for the IRL and IL methods[3]. Since the IRL reconstruction method requires an explicit transition function, we discretized the domain by running K-Means clustering on each state in the trajectory and using the the cluster centers as the state representations. We chose $n = 100$ clusters which resulted in accuracy of above 0.95 for predicting the action at each state as the most common action in each cluster.

5.2 Results

Figure 1 shows the accuracy for the complete set of reconstruction methods across all the different hyperparameters. Many methods were completely dominated by other methods. For ease of interpretation, we present the best performing methods in figure 2, showing the mean reconstruction accuracy for each combination of extraction model and reconstruction model over 20 random restarts.

Across all datasets, all models reconstruct the policy most accurately when the summary is extracted using the same model. In Figure 2, the accuracy for each column (corresponding to the reconstruction model) is highest for the row (corresponding to the extraction model), when the reconstruction model and extraction model match. Note that for IRL, there are several summaries extracted with different trajectory lengths that all correspond to the same IRL reconstruction method. This suggest that both the IL and IRL methods extract summaries that allow the model assumed during summary extraction to reconstruct the policy accurately.

In some cases, using any model of human computation other than the correct one during extraction leads to poor reconstruction. In the random gridworld domain, which involves tiles with different

[3]There is little variance between episodes so 5 is sufficient to capture variation.

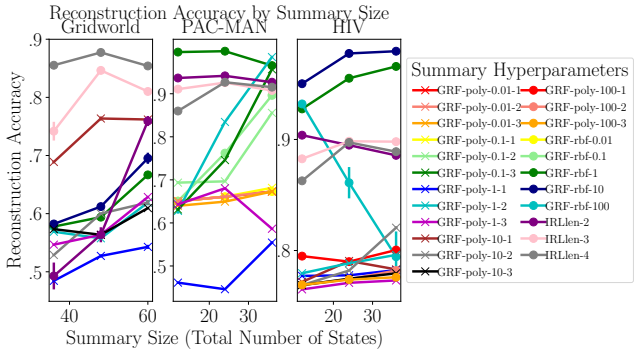
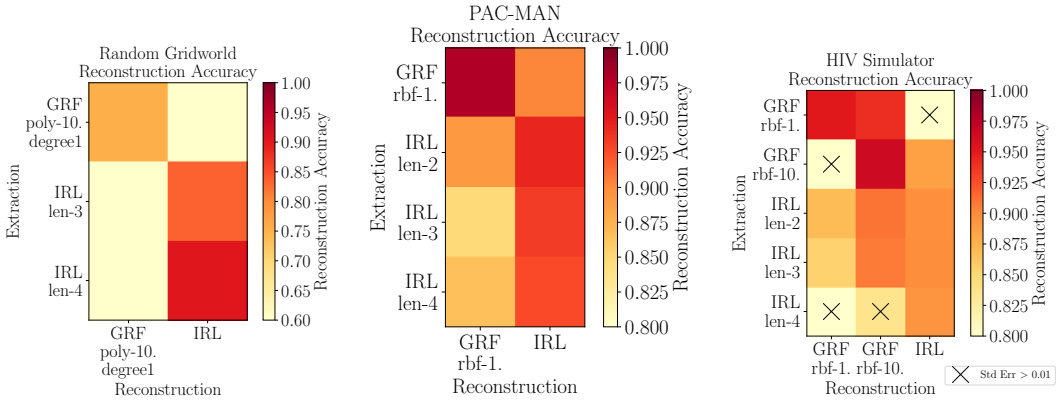


Figure 1: Reconstruction accuracy across a variety of reconstruction methods and hyper-parameter settings (assuming the same extraction method), by summary size, averaged over 5 random restarts. We include standard error bars but they are generally too small to see. Some methods completely dominate others; we further investigate the robustness of best-performing methods in Figure 2.

rewards, the IRL model reconstructs the policy with higher accuracy than the IL model. This makes sense, as the agent's behavior in this domain was determined using value iteration where the reward function was defined as a linear function of the state features, so IRL had a clear advantage over IL. Despite this, the IL model in Figure 2(a) (top row) can still produce summaries that the model can reconstruct with accuracy of 0.75 ± 0.002.

More interestingly, in this domain, the IRL model cannot reconstruct the policy well with this summary, and the IL model cannot reconstruct the model well with the summary extracted with the IRL models. This suggests that in some settings, none of the models we use to extract summaries are robust to a mismatch between the model used during summary extraction and the model used for reconstruction; in such settings we must make sure that we have an accurate model of human computation for the reconstruction.

In other settings, some summaries are more robust to a mismatch between summary extraction and reconstruction models than others. In PAC-MAN (Figure 2(b)), the summary extracted with IRL and trajectory length 2 (second row) allows the GRF model with length scale 1 (first column) to reconstruct the policy with higher accuracy than the summaries generated with other choices of trajectory length (rows 3 and 4). In the HIV simulator (Figure 2(c)), the summary extracted with the GRF model with RBF kernel and length scale 10 (second row) allows the IRL model to reconstruct the policy more accurately than the GRF model with RBF kernel and length scale 1 (row 1). The summaries extracted with the IRL model with trajectory lengths 2 and 3 (third and fourth rows) allow both GRF models (first and second columns) to reconstruct the policy more accurately than the summary extracted with the IRL model and trajectory length 4 (fifth row). This suggests that under certain settings, there may exist models of human computation that will produce summaries that are robust across a variety of true models, and this property can be checked simply via computation (that is, without user studies).



(a) Random Grid World **(b) Pacman** **(c) HIV Simulator**

Figure 2: Accuracy averaged over 20 random restarts of every reconstruction model used for summary extraction with summaries extracted with each model. The rows correspond to extraction models and the columns correspond to reconstruction models. The number following the kernel type indicates the length scale. All reconstruction models reconstruct the policy most accurately with the matched summary extraction model. In the PAC-MAN and HIV domains, some summary extraction models are more robust to misspecification of the reconstruction model than others; in the random gridworld domain, no summary extraction models are robust to misspecification.

Related, we see that hyperparameter choices also affect the robustness of the reconstruction. Specifically, in the HIV domain, we see that one hyperparameter setting of IL model was more robust to misspecified hyperparameters than the other: The summary extracted with the IL model with RBF kernel and gamma 1 (row 1) allows the other IL model (row 2) to reconstruct the summary accurately, while the summary extracted with the IL model with RBF kernel and gamma 10 (row 2) does not. Whether it is important to match both the model class and the hyperparameter settings to human mental models, or whether it is enough to get the model class right is an area for future work.

[Ike: I want to rethink this part, but I don't know whether to remove it or to do it differently. Maybe depends on the user study? For now, I think we can ust cut it!]

A mismatch in features may result in poor reconstruction. In the PAC-MAN domain, we experimented with giving the agent two different ways to represent its state. As can be seen in Table 1, if we extract a summary with any of the methods used in figure 2(b), neither the GRF with the features derived for IRL nor the IRL with the features derived for IL is able to reconstruct the policy with any summary with reasonable accuracy. While this is not surprising, it suggests that even if we extract summaries using a model that matches how humans generalize, we may not produce good summaries if we do not also use the same set of features that the human is using to characterize states.

6 HUMAN-SUBJECT STUDY

Our computational experiments showed that a mismatch between summary extraction methods and the methods used to reconstruct a policy can lead to wrong generalizations. We conducted a preliminary user study to assess people's reconstruction strategies, and in particular, their ability to extrapolate from summaries optimized for IL or IRL reconstruction methods.

Extraction Model	Reconstruction Model	Reconstruction Accuracy ± Std Err
GRF-rbf, 1.0	GRF-rbf,1.0-IRL feat.	0.349 ± 0.002
GRF-rbf,1.0	IRL-IL feat.	0.567 ± 0.000
IRL-trajlen2	GRF-rbf,1.0-IRL feat.	0.321 ± 0.002
IRL-trajlen2	IRL-IL feat.	0.567 ± 0.000
IRL-trajlen3	GRF-rbf,1.0-IRL feat.	0.333 ± 0.003
IRL-trajlen3	IRL-IL feat.	0.553 ± 0.000
IRL-trajlen4	GRF-rbf,1.0-IRL feat.	0.353 ± 0.002
IRL-trajlen4	IRL-IL feat.	0.581 ± 0.001

Table 1: For each summary in Figure 2(b), we compute the reconstruction accuracy with every model with features derived for the other model (i.e. we use the features derived for IRL to do IL and vice versa). No reconstruction model achieves accuracy higher than 0.6.

6.1 Empirical Methodology

Participants. 71 participants were recruited through Amazon Mechanical Turk (28 female, Mean age = 39.69, STD = 12.27). Participants received a base payment of $1.5, and could earn a bonus of up to $1 with respect to the percentage of correct answers.

Task. In the experiment, participants were shown summaries of agents using different policies to navigate 2-D maps. The maps consist of colored tiles on a 2-D grid, where colors correspond to state features. The summary shows how the agent acts in some parts of the map by displaying arrows corresponding to actions in some tiles. Each summary was generated to allow accurate reconstruction of the policy with either IRL or IL. Because the IRL-based approach extracts trajectories of states, in those summaries we used arrows of different colors to indicate separate agent trajectories.

Participants were asked to predict which action the agent would take in states shown on a different 2-D grid, if it employs the same strategy presented in the summary. These states were shown next to the summary, such that participants could inspect the summary while making their predictions. Figure 3 shows example action

prediction tasks for IRL-based and IL-based summaries. Participants were also asked to report their confidence in their prediction on a 7-point Likert scale (1 - not at all confident to 7 - very confident).

The summaries were displayed on a 4x4 grid, with 4 binary features per cell representing different colors. The test grids used for the prediction task were of size 2x3 and were created for each policy and summary method. In all grids, states correspond to the grid cells, where for each state only one of the binary features was active. The possible actions were moving in four directions (up, down, left and right), where the agent could not take actions causing it to leave the grid borders. For each policy, We created two state-feature mappings: one for IL, mapping state features to actions, and the other for IRL, mapping state features to rewards.

We used a summary budget of 12, i.e., each summary showed 12 state-action pairs. IRL-based summaries were extracted using machine teaching, with trajectories of length 3 (for a total of 4 trajectories). Summaries were extracted for IL by choosing a state of each color and then filling with random states to get a summary of the same size; in this small domain, active learning was not needed. We use an SVM with a linear kernel to measure reconstruction since this was sufficient to capture this simple policy.

The feature representation of states, test grids and the policies to be summarized were handcrafted such that each summary obtained reconstruction accuracy of 1.0 if the extraction model matched the reconstruction model, and obtained an accuracy of less than 0.5 if the extraction model did not match the reconstruction model. (Reconstruction accuracy was computed for the policy states not shown in the summary, and for all states in the test grid.) We chose this setup to see whether participants would employ the correct reconstruction method, or whether they would use the same reconstruction methods for different summaries, which we expect to result in low accuracy.

Procedure. Participants were first shown a tutorial explaining the grid world properties and the structure of the summaries. To avoid biasing participants toward a particular reconstruction method, they were only told that the agent's behavior is in some way related to the colors of tiles, but they were not told whether colors map to actions (IL) or to rewards (IRL). Then, they had to pass a quiz ensuring they read and understood the instructions. Next, they were asked to complete the task described above for summaries of four different agent policies, two extracted using IRL and two using IL. For each summary, after making all predictions, participants were asked to provide a brief text description of the agent's behavior.

The first two tasks were considered training tasks (though this was not mentioned to the participants), intended to familiarize them with the domain and interface. In the training phase, participants were shown one IRL and one IL summary, and asked to predict the action only in two tiles of the test grid. Next, they were shown two more summaries, one IRL and one IL, and were asked to predict the action taken by the agent in each of the tiles in the test grid (6 in total). The order in which the IRL and IL summaries were shown was randomized to avoid bias due to learning effects. The assignment of summary extraction method to agent policies was also randomized to avoid potential effects of different generalization difficulty levels for different policies. For each grid, participants were asked to predict actions for each empty tile one at a time. We used a within-subject study design, such that all participants evaluated both summary methods.

6.2 Results

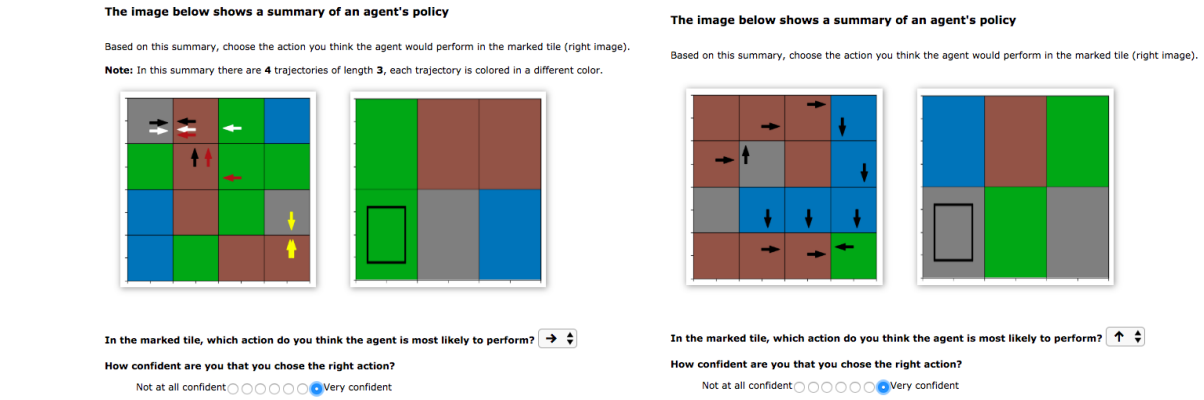
As shown in Figure 4(a), participants were able to predict the agent's actions much more accurately when given summaries extracted by the IL method than for those extracted with IRL (Mean correct answers: IL = 0.89, IRL = 0.39). Moreover, participants took twice as long to predict each action when shown the IRL-based summaries compared to IL-based summaries (Mean time in seconds: IL = 8.96, IRL = 16.16), see figure 4(b). The differences in accuracy and time were statistically significant ($p < 0.001$ in both cases, using Wilcoxon signed-ranked test). We conducted another study on a small group of participants ($N = 18$) where we showed only summaries extracted using IRL, to test whether participants are fixated by the IL summaries (although IL and IRL summaries were shown in an alternating order). The mean percent of correct answers increased from 0.39 to 0.6, but remained significantly lower than for the IL summaries (0.89).

Looking at the explanations provided for each summary, we observed that for the IL summaries most participants based their predictions on a mapping of colors to actions, naturally using IL for reconstruction. E.g. "Blue moves left, grey moves down, brown moves right, green moves up" or "Direction was based on color". This behavior implies that participants were imitating the agent behavior they observed in the summary. For the IRL summaries, the explanations indicate that participants had difficulty understanding the agent behavior. Many of them still tried mimicking the actions presented in the summary, e.g "he was all over the place but I tried to mimic the direction on the color". Some participants also based their explanations on directional behavior regardless of color, e.g. "The agent seemed to always go down if it could". This emphasizes the effect of prior assumptions people make regarding a domain on their comprehension—even though participants were explicitly told (and answered a quiz question) that the behavior was only affected in some way by colors, some of them still assumed the agent was trying to get to a certain location or move in a certain pattern. There were only few participants that noticed the different colors have different values to the agent, and that the actions are taken accordingly to obtain the highest value, e.g. "Preference of color (move cost). High to low: green, blue, brown, grey".

Confidence ratings for IL summaries were higher when participants were correct in their predictions, compared to when they were wrong (mean confidence: correct = 5.72, incorrect = 5.0,$p = 0.0003$). As opposed to that, for the IRL summaries the differences in confidence between correct and incorrect answers was insignificant (mean confidence: correct = 4.16, incorrect = 4.01, $p = 0.26$). Also, the mean confidence for IRL summaries was significantly lower than reported for IL (IRL = 4.07, IL = 5.65, $p < 0.001$). These results are in line with the descriptive explanations provided by the participants, showing they were more confident in their predictions based on IL summaries.

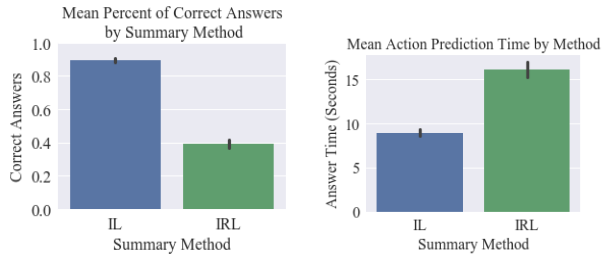
7 DISCUSSION & FUTURE WORK

Summaries of agent policies have potential to improve human users' understanding of agents' behavior. In this paper, we investigated



(a) Inverse Reinforcement Learning **(b) Imitation Learning**

Figure 3: Experimental interface. (a) an IRL-based summary and action prediction task; (b) and IL-based summary and action prediction task. Arrows indicate the agent's actions in different states in the grid; the IRL approach extracts summaries that visit the same state multiple times–denoted with offset arrows, while the IL approach does not. Participants are asked to predict the action for tiles (shown in a square) on a different grid.



(a) **(b)**

Figure 4: Participants were more accurate when predicting actions, and responded faster, when shown IL-based summaries. (a) Mean accuracy of participants using IRL-based and IL-based summaries; (b) Mean response times for action prediction when shown IRL-based and IL-based summaries.

different approaches for extracting policy summaries, which differ in their assumptions about the methods that might be used by people to extrapolate from these summaries and reconstruct an agent's policy. We conducted computational experiments to investigate the effect of using different computational models both for summary extraction and for policy reconstruction.

Specifically, we explored, both computationally and through user studies, differences between summaries that presume that the human will use a future-value oriented way of processing a summary (IRL) or use a mimic-based approach (IL). Across different types of domains, we found that different models of computation might be preferred. We also found that for any given reconstruction model used, the highest policy reconstruction accuracy was achieved when the summary was extracted using the same model. While in some domains there were summaries that were fairly robust to different reconstruction models, in other domains it was crucial that the extraction and reconstruction models match. Moreover, we found that matching between the feature representation used for summary extraction and policy reconstruction can also be important, with mismatch resulting in low reconstruction accuracy.

While we explored several variants of each IL and IRL, there are many design choices. For example, we defined the optimization goal as predicting the agent's behavior in unseen states; prior work has considered accurately modeling the reward function [12], or including "important" states, as measured by Q-value differences [2, 11]. For the optimization procedures, our machine teaching approach for IRL stopped at a fixed budget, even if perfect recovery was not achieved; our active learning for IL was chosen because it was computationally-efficient (methods with guarantees do exist, e.g. [14], but do not account for sets of fixed size, or the requirement to only include examples that exist in a dataset). Finally, we used a Gaussian random field model for IL and the Max-Ent approach for IRL; we could have chosen other classifiers and IRL approaches.

That said, with the variation we capture, our results emphasize the importance of selecting summaries that align with users' policy reconstruction model and feature representation. The results are further supported through a preliminary human-subject study, which showed that people failed to predict an agent's policy when their reconstruction strategy did not match the strategy for which summaries were optimized. In our experiment, people tended to make imitation learning-based inferences, causing them to fail to extrapolate from summaries optimized for reconstruction with IRL.

Predicting which reconstruction strategies people will deploy can be difficult, since their strategies can vary depending on the context and their a priori knowledge and assumptions about the domain and the agent's behavior; importantly, different users might deploy different reconstruction strategies. In cases where some extraction model is robust across different reconstruction models, summaries based on the model might be beneficial even if people use different reconstruction approaches. However, since our results show that in some domains there might not be a robust extraction

model, and that feature representation alignment between extraction and reconstruction also substantially impacts the ability to recover a policy, finding robust summaries might not be feasible.

To address this problem, one approach for future work could be a human-in-the-loop summary extraction process that first elicits users' reconstruction method and feature representation and then extracts summaries that align with the user's model. Another option would be to use user-studies to identify in which types of situations people generally choose one reconstruction approach over the other, as well as learn ways to prime users to use the reconstruction approach which could result in better generalization for a given domain. Given the variability that we see in reconstruction across different types of domains, these kinds of studies will be essential for endowing users with accurate mental models of agents; our work demonstrates that simply showing trajectories, even intelligently chosen to be robust, will not be sufficient.

REFERENCES

[1] BM Adams, HT Banks, M Davidian, Hee-Dae Kwon, HT Tran, SN Wynne, and ES Rosenberg. Hiv dynamics: modeling, data analysis, and optimal treatment protocols. *Journal of Computational and Applied Mathematics*, 184(1):10–49, 2005.

[2] Dan Amir and Ofra Amir. Highlights: Summarizing agent behavior to people. In *Proc. of the 17th International conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2018.

[3] Ofra Amir, Finale Doshi-Velez, and David Sarne. Agent strategy summarization. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1203–1207. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[4] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[5] Joost Broekens, Maaike Harbers, Koen Hindriks, Karel van den Bosch, Catholijn Jonker, and John-Jules Meyer. Do you get it? user-evaluated explainable bdi agents. In Jürgen Dix and Cees Witteveen, editors, *Multiagent System Technologies*, pages 28–39, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[6] Daniel S. Brown and Scott Niekum. Machine teaching for inverse reinforcement learning: Algorithms and applications. *CoRR*, abs/1805.07687, 2018.

[7] Thomas Dodson, Nicholas Mattei, and Judy Goldsmith. A natural language argumentation interface for explanation generation in markov decision processes. In *International Conference on Algorithmic DecisionTheory*, pages 42–55. Springer, 2011.

[8] Anca Dragan and Siddhartha Srinivasa. Familiarization to robot motion. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 366–373. ACM, 2014.

[9] Damien Ernst, Guy-Bart Stan, Jorge Goncalves, and Louis Wehenkel. Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Decision and Control, 2006 45th IEEE Conference on*, pages 667–672. IEEE, 2006.

[10] Samuel J Gershman. Reinforcement learning and causal models. *The Oxford handbook of causal reasoning*, page 295, 2017.

[11] Sandy H. Huang, Kush Bhatia, Pieter Abbeel, and Anca D. Dragan. Leveraging critical states to develop trust.

[12] Sandy H. Huang, David Held, Pieter Abbeel, and Anca D. Dragan. Enabling robots to communicate their objectives. *CoRR*, abs/1702.03465, 2017.

[13] Omar Zia Khan, Pascal Poupart, and James P Black. Minimal sufficient explanations for factored markov decision processes. In *ICAPS*, 2009.

[14] Ji Liu and Xiaojin Zhu. The teahing dimension of linear learners. *Journal of Machine Learning Research*, 17(162):1–25, 2016.

[15] Meghann Lomas, Robert Chevalier, Ernest Vincent Cross II, Robert Christopher Garrett, John Hoare, and Michael Kopack. Explaining robot actions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 187–188. ACM, 2012.

[16] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[17] Bilal Piot, Matthieu Geist, and Olivier Pietquin. Learning from demonstrations: Is it worth estimating a reward function? In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 17–32. Springer, 2013.

[18] Ramya Ramakrishnan, Ece Kamar, Debadeepta Dey, Julie A. Shah, and Eric Horvitz. Discovering blind spots in reinforcement learning. *CoRR*, abs/1805.08966, 2018.

[19] Burr Settles. Active learning literature survey. Technical report, 2010.

[20] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 912–919. AAAI Press, 2003.

[21] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. *ICML 2003 Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, pages 58–65, 2003.

[22] Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pages 4083–4087, 2015.

[23] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.