SIRL: Similarity-based Implicit Representation Learning

Andreea Bobu*
abobu@berkeley.edu
University of California, Berkeley

Yi Liu*
yiliu77@berkeley.edu
University of California, Berkeley

Rohin Shah
rohinmshah@deepmind.com
DeepMind Research

Daniel S. Brown
dsbrown@berkeley.edu
University of California, Berkeley

Anca D. Dragan
anca@berkeley.edu
University of California, Berkeley

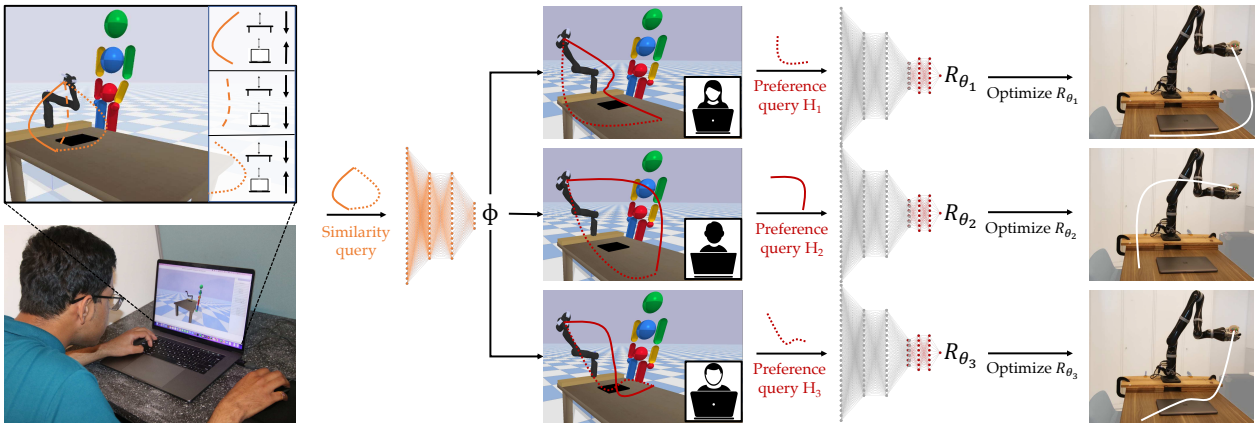


Figure 1: Our goal is to learn representations for robot behavior that capture what is salient to people, and, thus, support generalizable preference learning with low sample complexity. We propose to extract this representation by asking people trajectory similarity queries (left), where they judge which two out of three trajectories are most similar to each other. We then use the representation to learn reward functions corresponding to different people's preferences on different tasks (right).

ABSTRACT

When robots learn reward functions using high capacity models that take raw state directly as input, they need to both learn a *representation* for what matters in the task — the task "features" — as well as how to combine these features into a single objective. If they try to do both at once from input designed to teach the full reward function, it is easy to end up with a representation that contains spurious correlations in the data, which fails to generalize to new settings. Instead, our ultimate goal is to enable robots to identify and isolate the causal features that people actually care about and use when they represent states and behavior. Our idea is that we can tune into this representation by asking users what behaviors they consider *similar*: behaviors will be similar if the features that matter are similar, even if low-level behavior is different; conversely, behaviors will be different if even one of the features that matter differs. This, in turn, is what enables the robot to disambiguate between what needs to go into the representation versus what is spurious, as well as what aspects of behavior can be compressed together versus not. The notion of learning representations based on similarity has a nice parallel in contrastive learning, a self-supervised representation learning technique that maps visually similar data points to similar embeddings, where similarity is defined by a designer through data augmentation heuristics. By contrast, in order to learn the representations that people use, so we can learn *their* preferences and objectives, we use *their* definition of similarity. In simulation as well as in a user study, we show that learning through such similarity queries leads to representations that, while far from perfect, are indeed more generalizable than self-supervised and task-input alternatives.

CCS CONCEPTS

• **Computing methodologies** → **Learning from demonstrations**; **Learning latent representations**; *Cognitive robotics.*

KEYWORDS

robot reward learning, human-aligned representation learning

ACM Reference Format:
Andreea Bobu, Yi Liu, Rohin Shah, Daniel S. Brown, and Anca D. Dragan. 2023. SIRL: Similarity-based Implicit Representation Learning. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (HRI '23), March 13–16, 2023, Stockholm, Sweden.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3568162.3576989

*Both authors contributed equally to this research. This research is supported by the Office of Naval Research (ONR) Young Investigator Award, the Weill Neurohub, and the Apple AI/ML Fellowship. We thank Sid Reddy and Andrea Bajcsy for their feedback.

HRI '23, March 13–16, 2023, Stockholm, Sweden

ACM ISBN 978-1-4503-9964-7/23/03.
https://doi.org/10.1145/3568162.3576989

1 INTRODUCTION

Imagine waking up in the morning and your home robot assistant wants to place a steaming mug of fresh coffee on the table exactly where it knows you will sit. Depending on the context, you will have a different preference for how the robot should be doing its task. Some days it carries your favorite mug close to the table to prevent it from breaking in the case of a slip (so that it will remain your favorite mug); other days the steam from your delicious meal is difficult to handle for the robot's perception, so you'd want it to keep a large clearance from the table to avoid collisions. Similarly, some days you want the robot to keep your mug away from your laptop to avoid spilling on it; other days the mug only has an espresso shot so you want the robot keep it close to the laptop to prevent clutter and leave the rest of the table open for you.

The reward function the robot should optimize changes — whether due to variations in the task, having different users, or, as in the examples above, different contexts that are not always part of the robot state (e.g. holding the user's favorite mug and not just a regular mug). However, the *representation* on top of which the reward is built, i.e the *features* that are important (like the distance from the table, being above the laptop, etc.), are shared. If the robot learns this representation correctly, it can use it to obtain the right reward function, even if the task, user, and context changes. Meta-learning and multi-task learning methods [30, 46, 56] learn the representation from user input meant to teach the full reward, like preference queries or demonstrations. By contrast, we propose that if learning generalizable representations is the goal, then we should ask the user for input that is specifically meant to teach the representation itself, rather than input meant to teach the full reward and hoping to extract a good representation along the way.

But asking people to teach robots representations, rather than tasks, is not so easy. What *are* the features that they care about? While some techniques advocate for people enabling users to teach each feature separately [11], people may not always be able to explicate their representation and break it down into concepts that are individually teachable. In this work, our idea is that we can implicitly tune into the representations people use by asking them to do a proxy task of evaluating *similarity* of behaviors. Behaviors will be similar if the features that matter are similar, even if low-level behavior is different; conversely, behaviors will be different if even one of the features that matter differs. This, in turn, should enable the robot to arrive at the features that matter — we want robots that can disambiguate between what needs to go into the representation versus what is spurious, as well as what aspects of behavior can be compressed together into a feature embedding versus kept separate. We thus introduce a novel type of human input to help the robot extract the person's representation: *trajectory similarity queries*. A trajectory similarity query is a triplet of trajectories that the person answers by picking the two more similar trajectories. In Figure 1 (left), the person chooses the two trajectories that are close to the table and far from the laptop, even though visually they look dissimilar. This results in an (anchor, positive, negative) triplet that can be used for training a feature representation. We call this process Similarity-based Implicit Representation Learning (SIRL).

Our method has a parallel in self-supervised learning work, especially contrastive learning, where the goal is to learn a good visual representation by training from (anchor, positive, negative) triplets generated via data augmentation techniques [19]. However, this notion of similarity is purely visual, driven by manually designed heuristics for data augmentation, and is not necessarily reflective of what users would consider similar. For instance, two images might be labeled as visually different, when in fact their difference is only with respect to some low-level aspects that are not really relevant to the distribution of tasks people care about. This would result in representations that contain too many distractor features that are not present in the human's representation. Our method uses similarity too, but we defer to the user's judgement of similarity, with the goal of reconstructing the *user's representation*.

Of course, our method is not the full answer to learning causally aligned representations. But our experiments suggest that it outperforms methods that are self-supervised, or that learn from input meant to teach the full tasks. In simulation, where we know the causal features, we show that SIRL learns representations better aligned with them, which in turn leads to learning multiple more generalizable reward functions downstream (Figure 1). We also present a user study where we crowdsource similarity queries from different people to learn a shared SIRL representation that better recovers each of their individual preferences. While the study results do show a significant effect, the effect size is much lower than in simulation. This is attributable in part to the interface difficulty of analyzing the robot trajectories, which means more work is needed to determine the best interfaces that enable users to accurately answer similarity queries. Moreover, some users reported struggling to trade off the different features, which means that similarity queries might not be entirely preference-agnostic. Nonetheless, our results underscore that there are gains by explicitly aligning robot and human representations, rather than hoping it will happen as a byproduct of learning rewards from standard queries.

2 RELATED WORK

Learning from Human Input. Human-in-the-loop learning is a well-established paradigm where the robot uses human input to infer a policy or reward function capturing the desired behavior. In imitation learning, the robot learns a policy that essentially copies human demonstrations [47], a strategy that typically doesn't generalize well outside the training regime [40]. Meanwhile, inverse reinforcement learning (IRL) uses the demonstrations to extract a reward function capturing *why* a specific behavior is desirable, thus better generalizing to unseen scenarios [1]. Recent research goes beyond demonstrations, utilizing other types of human input for reward learning, such as corrections [6], comparisons [55], or rankings [14]. Unless explicitly designed for, these methods learn a latent representation implicitly from the respective human input. We seek to instead explicitly learn a preference-agnostic latent space that can be used for downstream tasks like reward learning. We focus on learning models of human reward functions via pairwise preference queries [55], but we believe the latent space we learn can be useful for learning from any of the above types of feedback.

Representation and Similarity Learning. Common representation learning approaches are unsupervised [18, 20, 32] or self-supervised [4, 27, 39, 48], but because they are purposefully designed to bypass human supervision, the learned embedding does

not necessarily correspond to features the person cares about. Prior work leverages task labels [17] or trajectory rankings [13] to learn latent spaces that help identify specific goals or preferences. By contrast, we focus on learning task-agnositic measures of feature similarity that are useful for learning multiple preferences. Some work looks at having people interactively select features from a pre-defined set [15, 16, 42] or teach task-agnostic features sequentially via kinesthetic feature demonstrations [10] or active learning techniques [9, 31, 37]. We instead focus on fully learning a lower-dimensional feature representation all-at-once, rather than one at a time. Furthermore, rather than relying on the human to provide physical demonstrations for learning a good feature space [10, 11], we propose a more accessible and general form of human feedback: showing the user triplets of trajectories and simply asking them to label which two trajectories are the most similar. Triplet losses have been widely used to learn similarity models that capture how humans perceive objects [2, 3, 25, 44, 54]; however, to the best of our knowledge, we are the first to use a triplet loss to learn a general, task-agnostic similarity model of how humans perceive trajectories.

Meta- and Multi-Task Reward Function Learning. To learn multiple models of human reward functions, prior work has proposed clustering demonstrations and learning a different reward function for each cluster [5, 21, 26]; however, these methods require a large number of demonstrations and do not adapt to new reward functions. Meta-learning [29] seeks to learn a reward function initialization that enables fast fine-tuning at test time [33, 51, 57, 58]. Multi-task reward learning approaches pretrain a reward function on multiple human intents and then fine-tune the reward function at test time [30, 46]. This has been shown to be more stable and scalable than meta-learning approaches [43], but still requires curating a large set of training environments. By contrast, we do not assume any knowledge of the test-time task distribution *a priori* and do not require access to a population of different reward functions during training. Rather, we focus on learning a task-agnostic feature representation that can be utilized for down-stream reward learning tasks. In particular, we test our learned representation on the down-stream task of learning models of human reward functions via pairwise preference queries over trajectories [8, 41, 50, 55].

3 METHOD

We present our method for learning preference-agnostic representations from trajectory similarity queries. Our intuition is that if a human judges two behaviors to be similar, then their representations should also be similar. Since directly asking if two trajectories are similar is difficult without an explicit threshold, we instead present the human with a triplet of trajectories and ask them to pick the two most similar (or, equivalently, the most dissimilar one). We use the human's answers to train the representation such that similar trajectories have embeddings that are close and dissimilar trajectories map to embeddings far apart. The robot then uses this latent space as a shared representation for downstream preference learning tasks with multiple people, each with different preferences.

3.1 Preliminaries

We define a trajectory ξ as a sequence of states, and denote the space of all possible trajectories by Ξ. The human's preference over trajectories is given by a reward function $R : \Xi \mapsto \mathbb{R}$ that is unobserved by the robot and must be learned from human interaction. The robot reasons over a parameterized approximation of the reward function R_θ, where θ represents the parameters of a neural network. To learn θ, the robot collects human preference labels over trajectories [22, 55] and seeks to find parameters θ that maximize the likelihood of the human input. The robot can then use the learned reward function to score trajectories during motion planning in order to align its behavior with a particular human's preferences. We focus on explicitly using human input to first learn a good representation and then use that representation for down-stream reward learning, rather than using reward-specific human input (e.g., preferences or demonstrations) to implicitly learn the representation at the same time as the reward function.

3.2 Training the Feature Representation via Trajectory Similarity Queries

We seek to train a latent space that is useful for multiple down-stream preference learning tasks. To do this, we propose learning a preference-agnostic model of human similarity. One way to learn such a model would be to ask users to judge whether two trajectories are similar or not; however, humans are better at giving relative rather than binary or quantitative assessments of similarity [35, 53]. Thus, rather than asking users to use some internal threshold or scoring mechanism to quantitatively measure similarity, we instead focus on qualitative trajectory similarity queries. We present the user with a visualization of three trajectories and ask them to pick the two most similar ones (equivalently the most dissimilar one). The human's queries form a data set $\mathcal{D}_{sim} = \{(\xi^i_{P_1}, \xi^i_{P_2}, \xi^i_N)\}$, where $\xi^i_{P_1}$ and $\xi^i_{P_2}$ are the trajectories that are most similar and ξ^i_N is the trajectory most dissimilar to the other two.

We can interpret similarity (or dissimilarity) as a distance function, so we define the distance between two trajectories as the L_2 feature distance: $d(\xi_1, \xi_2) = \|\phi(\xi_1) - \phi(\xi_2)\|_2^2$. Given a dataset of trajectory similarity queries $\mathcal{D}_{sim}$, we use the triplet loss [7]:

$$\mathcal{L}_{trip}(\xi_A, \xi_P, \xi_N) = \max(d(\xi_A, \xi_P) - d(\xi_A, \xi_N) + \alpha, 0) \ , \quad (1)$$

a form of contrastive learning where ξ_A is the anchor, ξ_P is the positive example, ξ_N is the negative example, and $\alpha \geq 0$ is a margin between positive and negative pairs. However, because our queries do not contain an explicit anchor, our final loss is as follows:

$$\mathcal{L}_{sim}(\phi) = \sum_{i=1}^{|\mathcal{D}_{sim}|} \mathcal{L}_{trip}(\xi^i_{P_1}, \xi^i_{P_2}, \xi^i_N) + \mathcal{L}_{trip}(\xi^i_{P_2}, \xi^i_{P_1}, \xi^i_N) \ . \quad (2)$$

We train a similarity embedding function $\phi : \Xi \mapsto \mathbb{R}^d$ that minimizes the above similarity loss, where d is the representation dimensionality. The intuition is that optimizing this loss should push together the embeddings of similar trajectories and push apart the embeddings of dissimilar trajectories. Before training the representation with the loss in Eq. (2), we may also pre-train it using unsupervised learning [36], which we experiment with in Sec. 4.

3.3 Using SIRL for Reward Learning

Given a learned embedding ϕ, we can use it for learning models of specific user preferences. While we focus on learning from pairwise

preferences, we note that ϕ can in principle be used in downstream tasks that learn from many types of human feedback [34]. When learning a reward function from human preferences, we show the human two trajectories, ξ_A and ξ_B, and then ask which of these two the human prefers. We collect a data set of such preferences $\mathcal{D}_{pref} = \{(\xi_A^i, \xi_B^i, \ell^i)\}$ where $\ell^i = 1$ if ξ_A^i is preferred to ξ_B^i, denoted $\xi_A^i \succ \xi_B^i$, and $\ell^i = 0$ otherwise. We interpret the human's preferences through the lens of the Bradley-Terry preference model [12]:

$$P_\theta(\xi_A \succ \xi_B) = \frac{e^{R_\theta(\phi(\xi_A))}}{e^{R_\theta(\phi(\xi_A))} + e^{R_\theta(\phi(\xi_B))}} \quad . \tag{3}$$

We learn the reward function with a simple cross-entropy loss:

$$\mathcal{L}_{pref}(\theta) = -\sum_{i=0}^{|\mathcal{D}_{pref}|} \ell^i \log P_\theta(\xi_A^i \succ \xi_B^i) + (1-\ell^i) \log P_\theta(\xi_B^i \succ \xi_A^i) \quad . \tag{4}$$

3.4 Adapting to Different User Preferences

We want robots that can adapt to changes to an individual user's preferences depending on the context as well as quickly adapt to new users' preferences. Rather than learn each preference independently by collecting a new set of human data and training a completely new reward function R_θ, we study whether we can leverage the latent space learned by SIRL to perform more accurate and sample-efficient multi-preference learning. When learning a new user's preference model, R_θ the robot can use ϕ to more quickly learn the reward function $R_\theta(\phi(\xi))$. Our main idea is that because this shared latent representation ϕ is trained via preference-agnostic similarity queries, it is more transferable than using a multi-task or meta-learning approach, where the pre-trained network is trained using multiple, specific task objectives. Furthermore, because SIRL uses human input to train ϕ, we hypothesize that the learned feature space will be better suited for learning human reward functions than a latent space learned via unsupervised training.

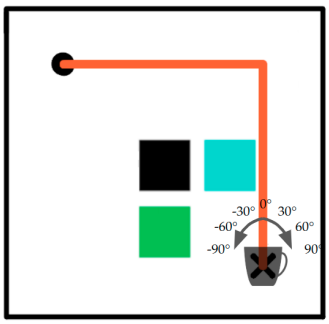
4 EXPERIMENTS IN SIMULATION

We first investigate the quality of SIRL-trained representations and their benefits for preference learning using simulated human input in two environments with ground truth rewards and features.

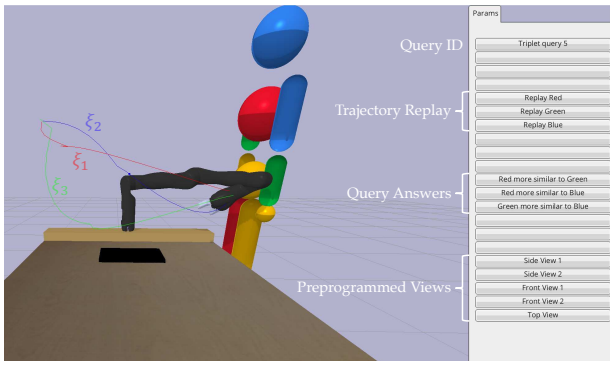
4.1 Environments

GridRobot (Figure 2a) is a 5-by-5 gridworld with two obstacles and a laptop (the blue, green, and black boxes). Trajectories are sequences of 9 states with the start and end in opposite corners. The 19-dimensional input consists of the x and y coordinates of each state and a discretized angle in $\{-90°, -60°, -30°, 0°, 30°, 60°, 90°\}$ at the end state. The simulated human answers queries based on 4 features ϕ^* in this world: Euclidean distances to each object, and the absolute value of the angle orientation.

JacoRobot (Figure 2b) is a pybullet [24] simulated environment with a 7-DoF Jaco robot arm on a tabletop, with a human and laptop in the environment. Trajectories are length 21, and each state consists of 97 dimensions: the xyz positions of all robot joints and objects, and their rotation matrices. This results in a 2037-dimensional input space, much larger than for GridRobot. The 4 features of interest ϕ^* for the simulated human are: a) *table* — distance of the robot's End-Effector (EE) to the table; b) *upright* — EE orientation relative to upright, to consider whether objects are carried upright; c) *laptop* — xy-plane distance of the EE to a laptop, to consider whether the EE passes over the laptop at any height; d) *proxemics* [45] — proxemic xy-plane distance of the EE to the human, where the EE is considered closer to the human when moving in front of the human that to their side.



(a) GridRobot environment.



(b) JacoRobot environment and user study interface.

Figure 2: Visualization of the experimental environments.

In GridRobot the state space is discretized, so the trajectory space Ξ can be enumerated; however, the JacoRobot state space is continuous, so we construct Ξ by smoothly perturbing the shortest path trajectories from 10,000 randomly sampled start-goal pairs (see App. A.1). We generate similarity and preference queries by randomly sampling from Ξ. The simulated human answers similarity queries by computing the 4 feature values for each of the three trajectories and choosing the two that were closest in the feature space. For preference queries, the simulated human computes the ground truth reward and samples the trajectory with the higher reward. The space of true reward functions (used to simulate preference labels) is defined as linear combinations of the 4 features described above. The robot is not given access to the ground-truth features nor the ground-truth reward function but must learn them from similarity and preference labels over raw trajectory observations.

4.2 Qualitative Examples

In Figure 3 we show similar and dissimilar trajectories learned by SIRL in a simplified GridRobot environment with only the laptop and the joint angle. *Top*: the given trajectory stays far from the laptop and holds the cup on its side; SIRL learns that trajectories that share those features are similar, despite being dissimilar in

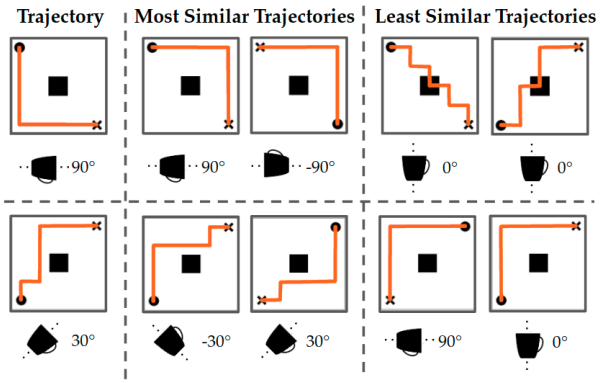


Figure 3: SIRL picks the two most and least similar trajectories to a query trajectory. Top: trajectories are similar in features despite being dissimilar in states. Bottom: trajectories are dissimilar in features despite being close in states.

the state-space. *Bottom*: the trajectory stays close to the laptop and holds the cup at an angle; SIRL learns that trajectories that hold the cup upright and stay far from the laptop are dissimilar, despite being similar in the state-space (going up and then right).

4.3 Experimental Setup

Manipulated Variables. We test the importance of user input that is designed to teach the representation by comparing SIRL with multi-task learning techniques from generic preference queries, and unsupervised representation learning. We have 4 baselines: a) ***VAE***, which learns a representation with a variational reconstruction loss [36]; b) ***MultiPref***, a multi-task baseline [30], where we learn the representation ϕ implicitly by training multiple reward functions (each with shared initial layers) via preference learning; c) ***SinglePref***, a hypothetical method that learns from an ideal user who weighs all features equally; d) ***Random***, a randomly initialized embedding, which does not benefit from human data but is also immune from any spurious correlations that might be learned from biased data. For MultiPref, we trained versions with 10 and 50 simulated human preference rewards for good coverage of the reward space. All embeddings have the same network size: for GridRobot we used MLPs with 2 layers, 128 units each, mapping to 6 output neurons, while for JacoRobot we used 1024 units to handle the larger input space (see App. A.2). For a fair comparison, we gave SIRL, SinglePref, and MultiPref equal amounts of human data for pre-training: N similarity queries for SIRL, and N preference queries (used for a single human for SinglePref or equally distributed amongst humans for MultiPref). We also performed ablations with and without VAE pre-training and found that SinglePref and MultiPref are better without VAE (see App. A.3).

Dependent Measures. To test the quality of the learned representations, we use two metrics: *Feature Prediction Error (FPE)* and *Test Preference Accuracy (TPA)*. The ***FPE*** metric is inspired by prior work that argues that good representations are linearly separable [23, 38, 49]. Our goal is to measure whether the embeddings contain the necessary information to recover the 4 ground-truth features in each environment. We generate data sets of sampled trajectories labeled with their ground truth (normalized) feature vector $\mathcal{D}_{FPE} = \{\xi, \phi^*\}$. We freeze each embedding and add a linear regression layer on top to predict the feature vector for a given trajectory. We split $\mathcal{D}_{FPE}$ into 80% training and 20% test pairs, and *FPE* is the mean squared error (MSE) on the test set between the predicted feature vector and the ground truth feature vector. For the human query methods, we report *FPE* with increasing number of representation training queries N.

For ***TPA***, we test whether good representations necessarily lead to good learning of general preferences. We use the trained embeddings as the base for 20 randomly selected test preference rewards. For each R_{θ_i}, we generate a set of labeled preference queries $\mathcal{D}_{pref}^{\theta_i} = \{\xi_A, \xi_B, l\}$, which we split into 80% for training and 20% for test. We train each reward model with M preference queries per test reward, and we vary M. All preference networks have the same architecture: we take the embedding ϕ pre-trained with the respective method, and add new fully connected layers to learn a reward function from trajectory preference labels. For GridRobot we used MLPs with 2 layers of 128 units, and for JacoRobot we used 1024 units. We found that all methods apart from SIRL worked better with unfrozen embeddings (App. A.3). We report TPA as the preference accuracy for the learned reward models on the test preference set, averaged across the test human preferences.

Hypotheses. We test two hypotheses:

H1. Using similarity queries specifically designed to teach the representation (SIRL) leads to representations more predictive of the true features (lower *FPE*) than unsupervised (VAE), implicit (MultiPref, SinglePref), or random representations.

H2. The SIRL representations result in more generalizable reward learning (higher *TPA*) than unsupervised (VAE), implicit (MultiPref, SinglePref), or random representations.

4.4 Results

In Figure 4 we show the *FPE* score for both environments with varying representation queries N from 100 to 1000. For GridRobot, both versions of SIRL (with or without VAE pre-training) perform similarly and outperform all baselines, supporting H1. When pre-training with preference queries, MultiPref with 10 humans performs better than SinglePref or MultiPref with 50 humans: SinglePref may be overfitting to the one human preference it has seen, while when MultiPref has to split its data budget among 50 humans it ends up learning a worse representation than Random. There is a balance to be struck between the diversity in human training rewards covered and the amount of pre-training data each reward gets, a trade-off which SIRL avoids because similarity queries are agnostic to the particular human reward. For the more complex JacoRobot, both versions of SIRL outperform all baselines, in line with H1, although SIRL without VAE scores better than with it.

In Figure 5 we present the *TPA* score for both environments with a varying amount of test preference queries M from 10 to 190, and N = 100, 500, and 1000. For GridRobot, each respective method performs comparably with different Ns, suggesting that this is a simple enough environment that low amounts of representation data are sufficient. For JacoRobot, this is not the case: with just 100 queries, SIRL with VAE pre-training performs like VAE, SIRL without pre-training has random performance (since it's frozen), and the preference baselines all perform close to Random, as if they

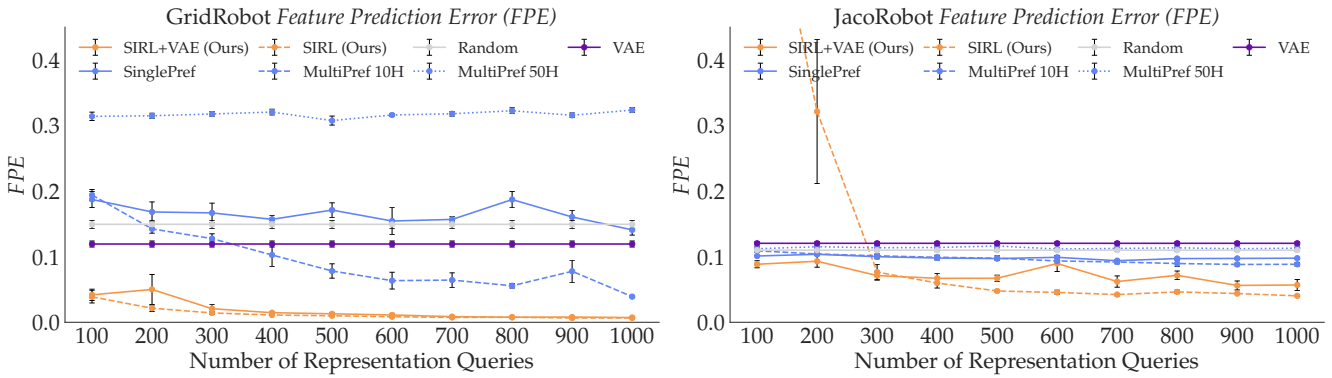


Figure 4: *FPE* for the GridRobot (left) and JacoRobot (right) environments with simulated human data. With enough data, SIRL learns representations more predictive of the true features ϕ^* in both simple and complex environments.

weren't trained with queries at all. For larger N, both versions of SIRL start performing better than the baselines, suggesting that with enough data a good representation can be learned.

Focusing on $N = 1000$, our results support H2: both SIRLs outperform all baselines in both environments, although for JacoRobot SIRL without VAE is better than with VAE. In the GridRobot environment VAE pre-training helps SIRL. However, while VAE performs comparably to other baselines in GridRobot, it severely underperforms in JacoRobot. This suggests that the reconstruction loss struggles to recover a helpful starting representation when the input space is higher dimensional and correlated. As a result, using the VAE pre-training to warmstart SIRL hinders performance when compared to starting from a blank slate. When comparing the preference-based baselines, in GridRobot they all perform similarly apart from MultiPref with 50 humans. In JacoRobot we see a trend that more preference humans does not necessarily result in better performance. This confirms our observation from Figure 4 that deciding on an appropriate number of human preferences to use for multi-task pretraining is challenging, a problem that SIRL bypasses.

Summary. With enough representation data, SIRL learns representations more predictive of the true features (H1), leading to learning more generalizable rewards (H2). This does not necessarily mean that SIRL representations are perfectly aligned with causal features — they are just *better* aligned, so the learned rewards are also better. When VAE pre-training recovers sensible starting representations it further reduces the amount of human data SIRL needs, otherwise it hurts performance. Lastly, surprisingly, Random is often better than pre-training with preference queries: preference-based methods may learn features that correlate with the training data but are not necessarily causal, and an incorrectly biased representation is worse for learning downstream rewards than starting from scratch.

5 USER STUDY

We now present a user study with novice users that provide similarity queries via an interface for the JacoRobot environment.

5.1 Experiment Design

We ran a user study in the JacoRobot environment, modified for only two features: *table* and *laptop* (we removed the humanoid in the environment). We designed an interface where people can click and drag to change the view, and press buttons to replay trajectories and record their query answer (Figure 2b). We chose to display the Euclidean path of each trajectory in the query traces, as we found that to help users more easily compare trajectories to one another.

The study has two phases: collecting similarity queries and collecting preference queries. In the first phase, we introduce the user to the interface and describe the two features of interest. Because similarity queries are preference-agnostic, we describe examples of possible preferences akin to the ones in Sec. 1, but we do not bias the participant towards any specific preference yet. Each participant practices answering a set of pre-selected, unrecorded similarity queries, and then answers 100 recorded similarity queries. In the second phase, we describe a scenario in the environment that has a specific preference associated with it (e.g. "There's smoke in the kitchen, so the robot should stay high from the table" or "There is smoke in the kitchen and the robot's mug is empty, so you want to stay far from the table and close to the laptop.") and assign different preference scenarios to each participant. Each person practices unrecorded preference queries, then answers 100 preference queries.

Participants. We recruited 10 users (3 female, 6 male, 1 non-binary, aged 20-28) from the campus community to give queries. Most users had technical background, so we caution that our results will speak to SIRL's usability with this population rather than more generally.

Manipulated Variables. Guided by the results in Figure 5, we compare our best performing method, SIRL without VAE, to Random, the best performing realistic baseline. For SIRL we collect 100 similarity queries from each participant and train a shared representation using all of their data.

Dependent Measures. We present the same two metrics from Sec. 4, *FPE* and *TPA*. For *TPA*, we collect 100 preference queries for each user's unique preference, we use 70% for training individual reward networks which we evaluate on the remaining 30% queries (Real). We compute *TPA* with cross-validation on 50 splits. To demonstrate how well SIRL works for new people who don't contribute to learning the similarity embedding, we also train SIRL on the similarity queries of 9 of the users and compute *TPA* on the held-out user's preference data (Held-out), for each user, respectively. Lastly, because real data tends to be noisy, we also compute *TPA* with 70

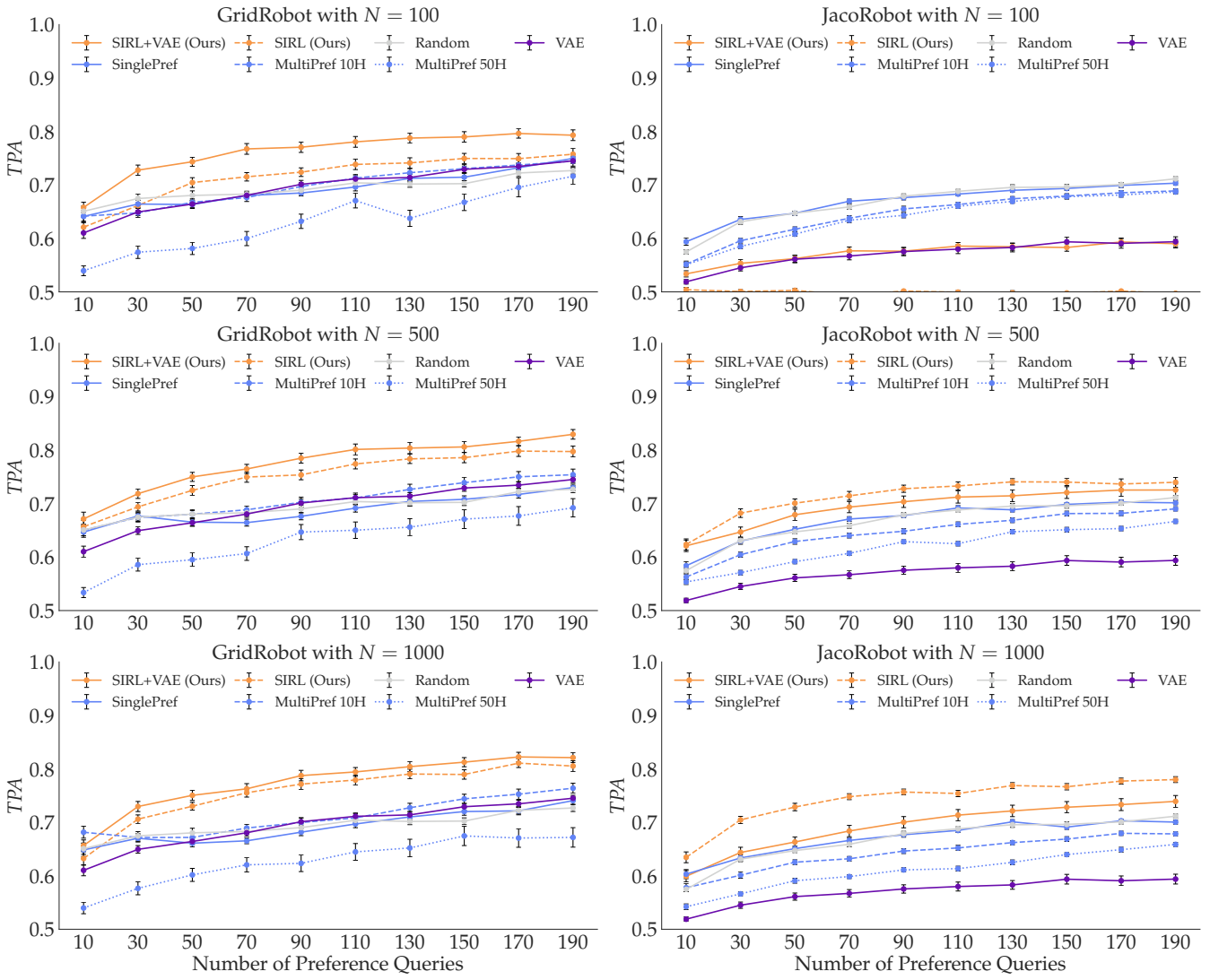


Figure 5: *TPA* for GridRobot (top) and JacoRobot (bottom) with simulated human data. With enough data, SIRL recovers more generalizable rewards than unsupervised, preference-trained, or random representations.

simulated preference queries for 10 different rewards, which we also evaluate on a simulated test set (Simulated).

Hypotheses. Our hypotheses for the study are:

H3. Similarity queries (SIRL) recover more salient features than a random representation (lower *FPE*), even with novice user data.

H4. The SIRL representation results in more generalizable reward learning (higher *TPA*), even with novice similarity queries.

5.2 Analysis

Figure 6 summarizes the results. On the left, SIRL recovers a representation twice as predictive of the true features, supporting H3. A 2-sided t-test ($p < .0001$) confirms this. This suggests SIRL can recover aspects of people's feature representation even with noisy similarity queries from novice users. On the right (Real), SIRL recovers more generalizable rewards on average than Random, providing evidence for H4. Furthermore, using the SIRL representation on a novel user (Held-out) also performs better than Random, and the result appears almost indistinguishable from Real. This suggests that similarity queries can be effectively crowd-sourced and the resulting representation works well for novel user preferences. Lastly, training with simulated preference queries slightly improves performance for both methods, suggesting that noise in the human preference data can be substantial. Three ANOVAs with method as a factor find a significant main effect ($F(1, 18) = 6.0175$, $p = .0246$, $F(1, 18) = 4.7547$, $p = .0427$, and $F(1, 18) = 16.1068$, $p < .001$, respectively). For each of the 3 cases, we also separated the 6 humans that were assigned preferences pertaining to both features (e.g. "There is smoke in the kitchen and the robot's mug is empty, so stay far from the table and close to the laptop."). SIRL performance is slightly better when using preference data from this subset of users, hinting that perhaps the learned representation entangled the two features.

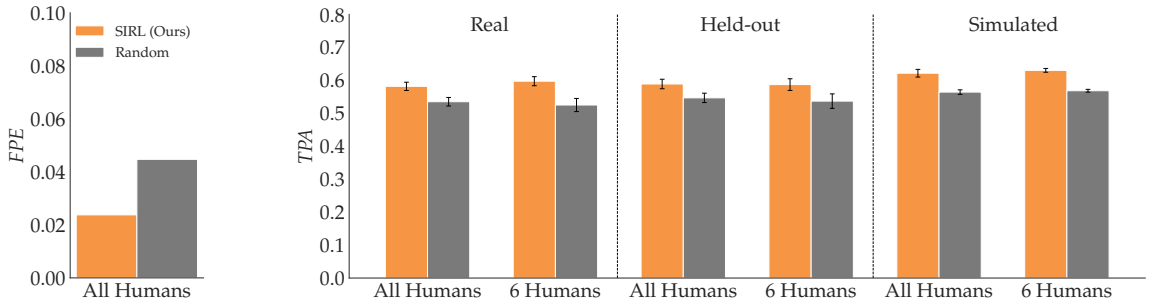


Figure 6: Study values for *FPE*, and *TPA* with real and simulated preferences. Even with novice similarity queries, SIRL recovers representations both more predictive of the true features and more useful for learning different user rewards than the baseline.

Overall, the quantitative results support H3 and H4, providing evidence that SIRL can recover more human-aligned representations. Subjectively, some users found the 2D interface deceiving at times, as they would judge trajectory similarity differently based on the viewpoint. This is a natural artifact of visualizing a 3D world in 2D, but future work should investigate better interfaces. Some users reported struggling to trade off the two features when comparing trajectories. This is in part due to the fact that we almost "engineered" their internal representation, so a more in-the-wild study could determine whether similarity queries are indeed preference-agnostic. Lastly, some queries were easier than others: users' time-to-answer varied across triplets suggesting that future work could use it as a confidence metric for how much to trust their answer.

6 DISCUSSION AND LIMITATIONS

In this work, our goal was to tackle the problem of learning good representations that capture (and disentangle) the features that matter, while excluding spurious features. If we had such a representation, then learning rewards that capture different preferences and tasks on top of it would lead to generalizable models that reliably incentivize the right behavior across different situations, rather than picking up on correlates and being unable to distinguish good from bad behaviors on new data. Our idea was to implicitly tap into this representation by asking people what they find similar versus not, because two behaviors will be similar if and only if their representations are similar. We introduced a novel human input type, trajectory similarity queries, and tested that it leads to better representations than those learned through self-supervision or via multi-task learning: it enables learning rewards from the same training data that better rank behaviors on test data.

That said, we need to be explicit that this is not the be-all end-all solution to our goal above. The representations learned, as we see in simulation, are not perfectly aligned with causal features — they are just *better* aligned. The learned rewards are not perfect — they are just better than alternatives. Similarity queries do not solve the problem fully, potentially because they suffer from the same issue preference queries do: when multiple important features change, it becomes harder to make a judgement call on what is more similar. The advantage that we see from similarity queries, though, is that rewards for particular tasks might ignore or down-weigh certain features that matter in other tasks, while similarity queries are task-agnostic and implicitly capture the distribution of tasks in the human's head. Rather than having to specify a task distribution for multi-task learning, with similarity queries we are (implicitly) asking the user to leverage their more general-purpose representation of the world. But thinking about how to overcome the challenge that multiple changing features make these queries harder to answer opens the door to exciting ideas for future work. For instance, what if we iteratively built the space, and based similarity queries on some current estimate of what are the important features; over time, as the representation becomes more aligned with the human's, the queries would get better at honing in on specific features.

Another obvious limitation is that we did not do an in-the-wild study. In theory, similarity queries should be used when people already have a robot they are familiar with and, thus, have a distribution of tasks they care about in their everyday contexts, but in our study we needed to explain to users these contexts and what might be important. In doing so, we almost "engineered" their internal representation. As robots become more prevalent, a follow-up study where users are given much less structure and allowed to actually tap into *their* unaltered representation might be possible.

In a sense, with SIRL we build a foundation model, and this sometimes requires hundreds of queries to learn a good representation. While we don't think having this much data when pre-training is unreasonable, especially since it leads to significant desirable performance improvement over baselines, sample-complexity is crucial to address as we scale to more complex robotic tasks. Because similarity queries are task agnostic, we can crowd-source the queries from multiple people (as we did in the user study) and rely on this economy at scale to alleviate user burden. Future work could also look at active querying methodologies to ask the person for more informative similarity queries and reduce data amounts.

A further avenue of work is extending this beyond reward learning, using SIRL representations directly for policy learning or learning exploration functions. We also emphasize that similarity queries are not a replacement for self-supervised learning; instead, we view them as complementary — self-supervised learning might be able to leverage expert-designed heuristics to eliminate many of the spurious features, while SIRL might serve to fine-tune the representation. How to properly combined the two remains an open question.

Overall, similarity queries are a step towards recovering human-aligned representations. They improve upon the state of the art, and can benefit from further exploration in how to combine them with other inputs and self-supervision, and how to make them easier through better interfaces and query selection algorithms.

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Machine Learning (ICML), International Conference on*. ACM.

[2] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. 2007. Generalized non-metric multidimensional scaling. In *Artificial Intelligence and Statistics*. PMLR, 11–18.

[3] Ehsan Amid, Aristides Gionis, and Antti Ukkonen. 2015. A Kernel-Learning Approach to Semi-supervised Clustering with Relative Distance Comparisons, Vol. 9284. https://doi.org/10.1007/978-3-319-23528-8_14

[4] Yusuf Aytar, Tobias Pfaff, David Budden, Tom Le Paine, Ziyu Wang, and Nando de Freitas. 2018. Playing Hard Exploration Games by Watching YouTube. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 2935–2945.

[5] Monica Babes, Vukosi N Marivate, Kaushik Subramanian, and Michael L Littman. 2011. Apprenticeship learning about multiple intentions. In *ICML*.

[6] Andrea Bajcsy, Dylan P. Losey, Marcia K. O'Malley, and Anca D. Dragan. 2017. Learning Robot Objectives from Physical Human Interaction. In *Proceedings of the 1st Annual Conference on Robot Learning (Proceedings of Machine Learning Research, Vol. 78)*, Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (Eds.). PMLR, 217–226. http://proceedings.mlr.press/v78/bajcsy17a.html

[7] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. 119.1–119.11. https://doi.org/10.5244/C.30.119

[8] Erdem Biyik and Dorsa Sadigh. 2018. Batch active preference-based learning of reward functions. In *Conference on robot learning*. PMLR, 519–528.

[9] Andreea Bobu, Chris Paxton, Wei Yang, Balakumar Sundaralingam, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. 2022. Learning Perceptual Concepts by Bootstrapping From Human Queries. *IEEE Robotics Autom. Lett.* 7, 4 (2022), 11260–11267. https://doi.org/10.1109/LRA.2022.3196164

[10] Andreea Bobu, Marius Wiggert, Claire Tomlin, and Anca D. Dragan. 0. Inducing Structure in Reward Learning by Learning Features. *The International Journal of Robotics Research* 0, 0 (0), 02783649221078031. https://doi.org/10.1177/02783649221078031 arXiv:https://doi.org/10.1177/02783649221078031

[11] Andreea Bobu, Marius Wiggert, Claire Tomlin, and Anca D. Dragan. 2021. Feature Expansive Reward Learning: Rethinking Human Input. In *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction* (Boulder, CO, USA) *(HRI '21)*. Association for Computing Machinery, New York, NY, USA, 216–224. https://doi.org/10.1145/3434073.3444667

[12] Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 39, 3/4 (1952), 324–345.

[13] Daniel Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. 2020. Safe Imitation Learning via Fast Bayesian Reward Inference from Preferences. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 1165–1177. http://proceedings.mlr.press/v119/brown20a.html

[14] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. 2019. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*. PMLR, 783–792.

[15] Kalesha Bullard, Sonia Chernova, and Andrea Lockerd Thomaz. 2018. Human-Driven Feature Selection for a Robotic Agent Learning Classification Tasks from Demonstration. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*. IEEE, 6923–6930. https://doi.org/10.1109/ICRA.2018.8461012

[16] Maya Cakmak and Andrea Lockerd Thomaz. 2012. Designing robot learners that ask good questions. In *International Conference on Human-Robot Interaction, HRI'12, Boston, MA, USA - March 05 - 08, 2012*, Holly A. Yanco, Aaron Steinfeld, Vanessa Evers, and Odest Chadwicke Jenkins (Eds.). ACM, 17–24. https://doi.org/10.1145/2157689.2157693

[17] Annie S. Chen, Suraj Nair, and Chelsea Finn. 2021. Learning Generalizable Robotic Reward Functions from "In-The-Wild" Human Videos. *CoRR* abs/2103.16817 (2021). arXiv:2103.16817 https://arxiv.org/abs/2103.16817

[18] Ricky T. Q. Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating Sources of Disentanglement in Variational Autoencoders. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc.

[19] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[20] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 2180–2188.

[21] Jaedeug Choi and Kee-Eung Kim. 2012. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. *Advances in Neural Information Processing Systems* 25 (2012).

[22] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf

[23] Adam Coates and A. Ng. 2012. Learning Feature Representations with K-Means. In *Neural Networks: Tricks of the Trade*.

[24] Erwin Coumans and Yunfei Bai. 2016–2019. PyBullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org.

[25] Cagatay Demiralp, Michael Bernstein, and Jeffrey Heer. 2014. Learning Perceptual Kernels for Visualization Design. *IEEE Transactions on Visualization and Computer Graphics* 20. https://doi.org/10.1109/TVCG.2014.2346978

[26] Christos Dimitrakakis and Constantin A Rothkopf. 2011. Bayesian multitask inverse reinforcement learning. In *European workshop on reinforcement learning*. Springer, 273–284.

[27] Carl Doersch, Abhinav Kumar Gupta, and Alexei A. Efros. 2015. Unsupervised Visual Representation Learning by Context Prediction. *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), 1422–1430.

[28] A. D. Dragan, K. Muelling, J. Andrew Bagnell, and S. S. Srinivasa. 2015. Movement primitives via optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2339–2346. https://doi.org/10.1109/ICRA.2015.7139510

[29] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 1126–1135.

[30] Adam Gleave and Oliver Habryka. 2018. Multi-task maximum entropy inverse reinforcement learning. *arXiv preprint arXiv:1805.08882* (2018).

[31] Bradley Hayes and Brian Scassellati. 2014. Discovering task constraints through observation and active learning. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*. IEEE, 4442–4449. https://doi.org/10.1109/IROS.2014.6943191

[32] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.

[33] Chao Huang, Wenhao Luo, and Rui Liu. 2021. Meta Preference Learning for Fast User Adaptation in Human-Supervisory Multi-Robot Deployments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 5851–5856.

[34] Hong Jun Jeon, Smitha Milli, and Anca Dragan. 2020. Reward-rational (implicit) choice: A unifying formalism for reward learning. *Advances in Neural Information Processing Systems* 33 (2020), 4415–4426.

[35] Maurice George Kendall. 1948. Rank correlation methods. (1948).

[36] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1312.6114

[37] Johannes Kulick, Marc Toussaint, Tobias Lang, and Manuel Lopes. 2013. Active Learning for Teaching a Robot Grounded Relational Symbols. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, Francesca Rossi (Ed.). IJCAI/AAAI, 1451–1457. http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6706

[38] Cheng-I Lai. 2019. Contrastive Predictive Coding Based Feature for Automatic Speaker Verification. *arXiv preprint arXiv:1904.01575* (2019).

[39] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2020. CURL: Contrastive Unsupervised Representations for Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 5639–5650. https://proceedings.mlr.press/v119/laskin20a.html

[40] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).

[41] Kejun Li, Maegan Tucker, Erdem Bıyık, Ellen Novoseller, Joel W Burdick, Yanan Sui, Dorsa Sadigh, Yisong Yue, and Aaron D Ames. 2021. Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3212–3218.

[42] Hoai Luu-Duc and Jun Miura. 2019. An Incremental Feature Set Refinement in a Programming by Demonstration Scenario. In *4th IEEE International Conference on Advanced Robotics and Mechatronics, ICARM 2019, Toyonaka, Japan, July 3-5, 2019*. IEEE, 372–377. https://doi.org/10.1109/ICARM.2019.8833723

[43] Zhao Mandi, Pieter Abbeel, and Stephen James. 2022. On the Effectiveness of Fine-tuning Versus Meta-reinforcement Learning. *arXiv preprint arXiv:2206.03271* (2022).

[44] Brian McFee, Gert Lanckriet, and Tony Jebara. 2011. Learning Multi-modal Similarity. *Journal of machine learning research* 12, 2 (2011).

[45] Jonathan Mumm and Bilge Mutlu. 2011. Human-robot proxemics: physical and psychological distancing in human-robot interaction. In *Proceedings of the 6th international conference on Human-robot interaction*. 331–338.

[46] Kentaro Nishi and Masamichi Shimosaka. 2020. Fine-grained driving behavior prediction via context-aware multi-task inverse reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2281–2287.

[47] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. 2018. An Algorithmic Perspective on Imitation Learning. *Foundations and Trends in Robotics* 7, 1-2 (2018), 1–179.

[48] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Fred Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. 2018. Zero-Shot Visual Imitation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2131–21313. https://doi.org/10.1109/CVPRW.2018.00278

[49] C. J. Reed, X. Yue, A. Nrusimha, S. Ebrahimi, V. Vijaykumar, R. Mao, B. Li, S. Zhang, D. Guillory, S. Metzger, K. Keutzer, and T. Darrell. 2022. Self-Supervised Pretraining Improves Self-Supervised Pretraining. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE Computer Society, Los Alamitos, CA, USA, 1050–1060. https://doi.org/10.1109/WACV51458.2022.00112

[50] Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. 2017. Active preference-based learning of reward functions. In *Robotics: Science and systems*.

[51] Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Richard Zemel. 2019. Smile: Scalable meta inverse reinforcement learning through context-conditional policies. *Advances in Neural Information Processing Systems* 32 (2019).

[52] Arjun Sripathy, Andreea Bobu, Zhongyu Li, Koushil Sreenath, Daniel S. Brown, and Anca D. Dragan. 2022. Teaching Robots to Span the Space of Functional Expressive Motion. https://doi.org/10.48550/ARXIV.2203.02091

[53] Neil Stewart, Gordon DA Brown, and Nick Chater. 2005. Absolute identification by relative judgment. *Psychological review* 112, 4 (2005), 881.

[54] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. 2011. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033* (2011).

[55] Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. 2017. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research* 18, 136 (2017), 1–46.

[56] Kelvin Xu, Ellis Ratner, Anca Dragan, Sergey Levine, and Chelsea Finn. 2019. Learning a Prior over Intent via Meta-Inverse Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6952–6962. https://proceedings.mlr.press/v97/xu19d.html

[57] Kelvin Xu, Ellis Ratner, Anca Dragan, Sergey Levine, and Chelsea Finn. 2019. Learning a prior over intent via meta-inverse reinforcement learning. In *International Conference on Machine Learning*. PMLR, 6952–6962.

[58] Lantao Yu, Tianhe Yu, Chelsea Finn, and Stefano Ermon. 2019. Meta-inverse reinforcement learning with probabilistic context variables. *Advances in Neural Information Processing Systems* 32 (2019).