Unsupervised Perceptual Rewards for Imitation Learning

Pierre Sermanet*  Kelvin Xu†  Sergey Levine∗  Sergey Levine
sermanet,kelvinxx,slevine@google.com
Google Brain

Abstract—Reward function design and exploration time are arguably the biggest obstacles to the deployment of reinforcement learning (RL) agents in the real world. In many real-world tasks, designing a reward function takes considerable hand engineering and often requires additional and potentially visible sensors to be installed just to measure whether the task has been executed successfully. Furthermore, many interesting tasks consist of multiple implicit intermediate steps that must be executed in sequence. Even when the final outcome can be measured, it does not necessarily provide feedback on these intermediate steps or sub-goals. To address these issues, we propose leveraging the abstraction power of intermediate visual representations learned by deep models to quickly infer perceptual reward functions from small numbers of demonstrations. We present a method that is able to identify key intermediate steps of a task from only a handful of demonstration sequences, and automatically identify the most discriminative features for identifying these steps. This method makes use of the features in a pre-trained deep model, but does not require any explicit specification of sub-goals. The resulting reward functions, which are dense and smooth, can then be used by an RL agent to learn to perform the task in real-world settings. To evaluate the learned reward functions, we present qualitative results on two real-world tasks and a quantitative evaluation against a human-designed reward function. We also demonstrate that our method can be used to learn a complex real-world door opening skill using a real robot, even when the demonstration used for reward learning is provided by a human using their own hand. To our knowledge, these are the first results showing that complex robotic manipulation skills can be learned directly and without supervised labels from a video of a human performing the task. Supplementary material and dataset are available at sermanet.github.io/rewards

I. INTRODUCTION

Social learning, such as imitation, plays a critical role in allowing humans and animals to acquire complex skills in the real world. Humans can use this weak form of supervision to acquire behaviors from very small numbers of demonstrations, in sharp contrast to deep reinforcement learning (RL) methods, which typically require extensive training data. In this work, we make use of two ideas to develop a scalable and efficient imitation learning method: first, imitation makes use of extensive prior knowledge to quickly glean the “gist” of a new task from even a small number of demonstrations; second, imitation involves both observation and trial-and-error learning (RL). Building on these ideas, we propose a reward learning method for understanding the intent of a user demonstration through the use of pre-trained visual features, which provide

the “prior knowledge” for efficient imitation. Our algorithm aims to discover not only the high-level goal of a task, but also the implicit sub-goals and steps that comprise more complex behaviors. Extracting such sub-goals can allow the agent to make maximal use of information contained in a demonstration. Once the reward function has been extracted, the agent can use its own experience at the task to determine the physical structure of the behavior, even when the reward is provided by an agent with a substantially different embodiment (e.g. a human providing a demonstration for a robot).

To our knowledge, our method is the first reward learning technique that learns generalizable vision-based reward functions for complex robotic manipulation skills from only a few demonstrations provided directly by a human. Although prior methods have demonstrated reward learning with vision for real-world robotic tasks, they have either required kinesthetic demonstrations with robot state for reward learning [12], or else required low-dimensional state spaces and numerous demonstrations [34]. The contributions of this paper are:

- A method for perceptual reward learning from only a few demonstrations of real-world tasks. Reward functions are dense and incremental, with automated unsupervised...
Few demonstrations
Unsupervised discovery of intermediate steps
Feature selection maximizing step discrimination across all videos

REAL ROBOT
OFFLINE COMPUTATION

Learning agent with Reinforcement Learning
Kinesthetic demonstration (successful ~10% of the time)
PI² approach from [Chebotar et al.]

Demonstrator (human or robot)

FEW D EjXJUMLAM^IJEIUX.

A. Related Work

Our method can be viewed as an instance of learning from demonstration (LfD), which has long been studied as a strategy to overcome the challenging problem of learning robotic tasks from scratch [7] (also see [3] for a survey). Previous work in LfD has successfully replicated gestures [8] and dual-arm manipulation [4] on real robotic systems. LfD can however be challenging when the human demonstrator and robot do not share the same “embodiment”. A mismatch between the robot’s sensor/actions and the recorded demonstration can lead to suboptimal behaviour. Previous work has addressed this issue by either providing additional feedback [33], or alternatively providing demonstrations which only give a coarse solutions which can be iteratively refined [24]. Our method of using perceptual rewards can be seen as an attempt to circumvent the need for shared embodiment in the general problem of learning from humans. To accomplish this, we first learn a visual reward from human demonstrations, and then learn how to perform the task using the robot’s own experience via reinforcement learning.

Our approach can be seen as an instance of the more general inverse reinforcement learning framework [20]. Inverse reinforcement learning can be performed with a variety of algorithms, ranging from margin-based methods [2, 27] to methods based on probabilistic models [25, 36]. Most similar in spirit to our method is the recently proposed SWIRL algorithm [1] and non-parametric reward segmentation approaches [26, 19, 21], which like our method attempts to decompose

- The first vision-based reward learning method that can learn a complex robotic manipulation task from a few human demonstrations in real-world robotic experiments.
- A set of empirical experiments that show that the learned visual representations inside a pre-trained deep model are general enough to be directly used to represent goals and sub-goals for manipulation skills in new scenes without retraining.

Fig. 2: Method overview. Given a few demonstration videos of the same action, our method discovers intermediate steps, then trains a classifier for each step on top of the mid and high-level representations of a pre-trained deep model (in this work, we use all activations starting from the first “mixed” layer that follows the first 5 convolutional layers). The step classifiers are then combined to produce a single reward function per step prior to learning. These intermediate rewards are combined into a single reward function. The reward function is then used by a real robot to learn the perform the demonstrated task as show in door.
the task into sub-tasks to learn a sequence of local rewards. These approaches differs in that they consider only problems with low-dimensional state spaces while our method can be applied to raw pixels. Our experimental evaluation shows that our method can learn a complex real-world door opening skill using visual rewards from a human demonstration.

We use visual features obtained from training a deep convolutional network on a standard image classification task to represent our visual rewards. Deep learning has been used in a range of applications within robotic perception and learning [29, 23, 6]. In work that has specifically examined the problem of learning reward functions from images, one common approach to image-based reward functions has been to directly specify a “target image” by showing the learner the raw pixels of a successful task completion state, and then using distance to that image (or its latent representation) as a reward function [18, 12, 32]. However, this approach has several shortcomings. First, the use of a target image presupposes that the system can achieve a substantially similar visual state, which precludes generalization to semantically similar but visually distinct situations. Second, the use of a target image does not provide the learner with information about which facet of the image is more or less important for task success, which might result in the learner excessively emphasizing irrelevant factors of variation (such as the color of a door due to light and shadow) at the expense of relevant factors (such as whether or not the door is open or closed).

A few recently proposed IRL algorithms have sought to combine IRL with vision and deep network representations [14, 34]. However, scaling IRL to high-dimensional systems and open-ended reward representations is very challenging. The previous work closest to ours used images together with robot state information (joint angles and end effector pose), with demonstrations provided through kinesthetic teaching [14]. The approach we propose in this work, which can be interpreted as a simple and efficient approximation to IRL, can use demonstrations that consist of videos of a human performing the task using their own body, and can acquire reward functions with intermediate sub-goals using just a few examples. This kind of efficient vision-based reward learning from videos of humans has not been demonstrated in prior IRL work. The idea of perceptual reward functions using raw pixels was also explored by [11] which, while sharing the same spirit as this work, was limited to synthetic tasks and used single images as perceptual goals rather than multiple demonstration videos.

II. SIMPLE INVERSE REINFORCEMENT LEARNING WITH VISUAL FEATURES

The key insight in our approach is that we can exploit the semantically meaningful and powerful features in a pre-trained deep neural network to infer task goals and sub-goals using a very simple approximate inverse reinforcement learning method. The pre-trained network effectively transfers prior knowledge about the visual world to make imitation learning fast and robust. Our approach can be interpreted as a simple approximation to inverse reinforcement learning under a particular choice of system dynamics, as discussed in Section II-A. While this approximation is somewhat simplistic, it affords an efficient and scalable learning rule that minimizes overfitting even when trained on a small number of demonstrations. As depicted in Fig. 2, our algorithm first segments the demonstrations into segments based on perceptual similarity, as described in Section II-B. Intuitively, the resulting segments correspond to sub-goals or steps of the task. The segments can then be used as a supervision signal for learning steps classifiers, described in Section II-C, which produces a single perception reward function for each step of the task. The combined reward function can then be used with a reinforcement learning algorithm to learn the demonstrated behavior. Although this method for extracting reward functions is exceedingly simple, its power comes from the use of highly general and robust pre-trained visual features, and our key empirical result is that such features are sufficient to acquire effective and generalizable reward functions for real-world manipulation skills.

We use the Inception network [30] pre-trained for ImageNet classification [10] to obtain the visual features for representing the learned rewards. It is well known that visual features in such networks are quite general and can be reused for other visual tasks. However, it is less clear if sparse subsets of such features can be used directly to represent goals and sub-goals for real-world manipulation skills. Our experimental evaluation suggests that indeed they can, and that the resulting reward representations are robust and reliable enough for real-world robotic learning without any finetuning of the features. In this work, we use all activations starting from the first “mixed” layer that follows the first 5 convolutional layers (this layer’s activation map is of size 35x35x256 given a 299x299 input). While this paper focuses on visual perception, the approach could be generalized to other modalities (e.g. audio and tactile).

A. Inverse Reinforcement Learning with Time-Independent Gaussian Models

In this work, we use a very simple approximation to the MaxEnt IRL model [36], a popular probabilistic approach to IRL. We will use $s_t$ to denote the visual feature activations at time $t$, which constitute the state, $s_{it}$ to denote the $i$th feature at time $t$, and $\tau = \{s_1, \ldots, s_T\}$ to denote a sequence or trajectory of these activations in a video. In MaxEnt IRL, the demonstrated trajectories $\tau$ are assumed to be drawn from a Boltzmann distribution according to:

$$p(\tau) = p(s_1, \ldots, s_T) = \frac{1}{Z} \exp \left( \sum_{t=1}^{T} R(s_t) \right),$$

where $R(s_t)$ is the unknown reward function. The principal computational challenge in MaxEnt IRL is to approximate $Z$, since the states at each time step are not independent, but are constrained by the system dynamics. In deterministic systems, where $s_{t+1} = f(s_t, a_t)$ for actions $a_t$ and dynamics
When faced with a difficult learning problem in extremely low-data regimes, a standard solution is to resort to simple, biased models, so as to minimize overfitting. We adopt precisely this approach in our work: instead of approximating the complex posterior distribution over trajectories under nonlinear dynamics, we use a simple biased model that affords efficient learning and minimizes overfitting. Specifically, we assume that all trajectories are dynamically feasible, and that the distribution over each activation at each time step is independent of the activation at the previous time step, which Naive Bayes uses an independence assumption to the IRL equivalent of a Naive Bayes model: in the same way that Naive Bayes is the minimum size of a split.

**Algorithm 1** Recursive similarity maximization, where \( \text{AverageStd}(\cdot) \) is a function that computes the average standard deviation over a set of frames or over a set of values, \( \text{Join}(\cdot) \) is a function that joins values or lists together into a single list, \( n \) is the number of splits desired and \( \text{min}_\text{size} \) is the minimum size of a split.

```plaintext
Algorithm 1

```
splits. We also experiment with a greedy binary version of this algorithm (Algorithm 2 detailed in Appendix A in [28]): first split the entire sequence in two, then recursively split each new segment in two. While not exactly minimizing the variance across all segments, it is significantly more efficient ($O(n^2 \log m)$) and yields qualitatively sensible results.

C. Steps Classification

In this section we explore learning a steps classifier on top of the pre-trained deep model using a regular linear classifier and a custom feature selection classifier.

One could consider a naive Bayes classifier here, however we show in Fig. 3 that it overfits badly when using all features. Other options are to select a small set of features, or use an alternative classification approach. We first discuss a feature selection approach that is efficient and simple. Then we discuss the alternative logistic regression approach which is simple but less principled. As our experimental results in Table II show, the performance slightly exceeds the selection method, though requires using all of the features.

Intent understanding requires identifying highly discriminative features of a specific goal while remaining invariant to unrelated variation (e.g. lighting, color, viewpoint). The relevant discriminative features may be very diverse and more or less abstract, which motivates our intuition to tap into the activations of deep models at different depths. Deep models cover a large set of representations that can be useful, from spatially dense and simple features in the lower layers (e.g. a large collection of detected edges) to gradually more spatially sparse and abstract features (e.g. few object classes).

![Average steps accuracy](image)

Fig. 3: Feature selection classification accuracy on the pouring validation set for 2-steps classification. By varying the number of features $n$ selected, we show that the method yields good results in the region $n = [32, 64]$, but collapses to 0% accuracy starting at $n = 8192$.

Within this high-dimensional representation, we hypothesize that there exists a subset of mid to high-level features that can readily and compactly discriminate previously unseen inputs. We investigate that hypothesis using a simple feature selection method described in Appendix C in [28]. The existence of a small subset of discriminative features can be useful for reducing overfitting in low data regimes, but more importantly can allow drastic reduction of the search space for the unsupervised steps discovery. Indeed since each frame is described by millions of features, finding patterns of feature correlations across videos leads to a combinatorial explosion. However, the problem may become tractable if there exists a low-dimensional subset of features that leads to reasonably accurate steps classification. We test and discuss that hypothesis in Section III-A2.

We also train a simple linear layer which takes as input the same mid to high level activations used for steps discovery and outputs a score for each step. This linear layer is trained using logistic regression and the underlying deep model is not fine-tuned. Despite the large input (1,453,824 units) and the low data regime (11 to 19 videos of 30 to 50 frames each), we show that this model does not severely overfit to the training data and perform slightly better than the feature selection method as described in Section III-A2.

D. Using Perceptual Rewards for Robotic Learning

In order to use our learned perceptual reward functions in a complete skill learning system, we must also choose a reinforcement learning algorithm and a policy representation. While in principle any reinforcement learning algorithm could be suitable for this task, we chose a method that is efficient for evaluation on real-world robotic systems in order to validate our approach. The method we use is based on the PI$^2$ reinforcement learning algorithm [31]. Our implementation, which is discussed in more detail in Appendix D in [28], uses a relatively simple linear-Gaussian parameterization of the policy, which corresponds to a sequence of open-loop torque commands with fixed linear feedback to correct for perturbations. This method also requires initialization from example demonstrations to learn complex manipulation tasks efficiently. A more complex neural network policy could also be used [9], and more sophisticated RL algorithms could also learn skills without demonstration initialization. However, since the main purpose of this component is to validate the learned reward functions, we used this simple approach to test our rewards quickly and efficiently.

III. Experiments

In this section, we discuss our empirical evaluation, starting with an analysis of the learned reward functions in terms of both qualitative reward structure and quantitative segmentation accuracy. We then present results for a real-world validation of our method on robotic door opening.

A. Perceptual Rewards Evaluation

We report results on two demonstrated tasks: door opening and liquid pouring. We collected about a dozen training videos for each task using a smart phone. As an example, Fig. 4 shows the entire training set used for the pouring task.
1) Qualitative Analysis: While a door opening sensor can be engineered using sensors hidden in the door, measuring pouring or container tilting would be quite complicated, would visually alter the scene, and is unrealistic for learning in the wild. Visual reward functions are therefore an excellent choice for complex physical phenomena such as liquid pouring. In Fig. 5, we present the combined reward functions for test videos on the pouring task, and Fig. 11 in [28] shows the intermediate rewards for each sub-goal. We plot the predicted reward functions for both successful and failed task executions in Fig. 12 in [28]. We observe that for “missed” executions where the task is only partially performed, the intermediate steps are correctly classified. Fig. 10 in [28] details qualitative results of unsupervised step segmentation for the door opening and pouring tasks. For the door task, the 2-segments splits are often quite in line with what one can expect, while a 3-segments split is less accurate. We also observe that the method is robust to the presence or absence of the handle on the door, as well as its opening direction. We find that for the pouring task, the 4-segments split often yields the most sensible break down. It is interesting to note that the 2-segment split usually occurs when the glass is about half full.

Failure Cases

The intermediate reward function for the door opening task which corresponds to a human hand manipulating the door handle seems rather noisy or wrong in Fig. 11b, 11c and 11e in [28] ("action1" on the y-axis of the plots). The reward function in Fig. 12f in [28] remains flat while liquid is being poured into the glass. The liquid being somewhat transparent, we suspect that it looks too similar to the transparent glass for the function to fire.

2) Quantitative Analysis: We evaluate the quantitative accuracy of the unsupervised steps discovery in Table I (see Table III in [28] for more details), while Table II presents quantitative generalization results for the learned reward on a test video of each task. For each video, ground truth intermediate steps were provided by human supervision for the purpose of evaluation. While this ground truth is subjective, since each task can be broken down in multiple ways, it is consistent for the simple tasks in our experiments. We use the Jaccard similarity measure (intersection over union) to indicate how much a detected step overlaps with its corresponding ground truth.

In Table I, we compare our method against a random baseline. Because we assume the same step order in all demonstrations, we also order the random steps in time to provide a fair baseline. Note that the random baseline performs fairly well because the steps are distributed somewhat uniformly in time. Should the steps be much less temporally uniform, the random baseline would be expected to perform very poorly, while our method should maintain similar performance. We compare splitting between 2 and 3 steps and find that, for both tasks, 2 steps are easier to discover, probably because these tasks exhibit one strong visual change each while the other steps are more subtle. Note that our unsupervised segmentation only works when full sequences are available while our learned reward functions can be used in real-time without accessing future frames. Hence in these experiments we evaluate the unsupervised segmentation on the training set only and evaluate the reward functions on the test set.

TABLE I: Unsupervised steps discovery accuracy (Jaccard overlap on training sets) versus the ordered random steps baseline.

<table>
<thead>
<tr>
<th>dataset</th>
<th>method</th>
<th>2 steps average</th>
<th>3 steps average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ordered random steps</td>
<td></td>
<td></td>
</tr>
<tr>
<td>door</td>
<td>unsupervised steps</td>
<td>52.5%</td>
<td>55.4%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>76.1%</td>
<td>66.9%</td>
</tr>
<tr>
<td>pouring</td>
<td>ordered random steps</td>
<td>65.9%</td>
<td>52.9%</td>
</tr>
<tr>
<td></td>
<td>unsupervised steps</td>
<td>91.6%</td>
<td>58.8%</td>
</tr>
</tbody>
</table>

In Table II, we evaluate the reward functions individually for each step on the test set. For that purpose, we binarize the reward function using a threshold of 0.5. The random baseline simply outputs true or false at each timestep. We observe that the learned feature selection and linear classifier functions outperform the baseline by about a factor of 2. It is not clear exactly what the minimum level of accuracy is required to successfully learn to perform these tasks, but we show in section III-B2 that the reward accuracy on the door task is sufficient to reach 100% success rate with a real robot. Individual steps accuracy details can be found in Table IV in [28].

Surprisingly, the linear classifier performs well and does not appear to overfit on our relatively small training set. Although the feature selection algorithm is rather close to the linear classifier compared to the baseline, using feature selection to
Fig. 5: Examples of "pouring" reward functions. We show here a few successful examples, see Fig. 12 in [28] for results on the entire test set. In Fig. 5a we observe a continuous and incremental reward as the task progresses and saturating as it is completed. Fig. 5b increases as the bottle appears but successfully detects that the task is not completed, while in Fig. 5c it successfully detects that the action is already completed from the start.

avoid overfitting is not necessary. However the idea that a small subset of features (32 in this case) can lead to reasonable classification accuracy is verified and an important piece of information for drastically reducing the search space for future work in unsupervised steps discovery. Additionally, we show in Fig. 3 that the feature selection approach works well when the number of features \( n \) is in the region \([32, 64]\) but collapses to 0% accuracy when \( n > 8192 \).

B. Real-World Robotic Door Opening

In this section, we aim to answer the question of whether our previously visualized reward function can be used to learn a real-world robotic motion skill. We experiment on a door opening skill, where we adapt a demonstrated door opening to a novel configuration, such as different position or orientation of the door. Following the experimental protocol in prior work [9], we adapt an imperfect kinesthetic demonstration which we ensure succeeds occasionally (about 10% of the time). These demonstrations consist only of robot poses, and do not include images. We then use a variety of different video demonstrations, which contain images but not robot poses, to learn the reward function. These videos include demonstrations with other doors, and even demonstrations provided by a human using their own body, rather than through kinesthetic teaching with the robot. Across all experiments, we use a total of 11 videos. We provide videos of experiments in 1.

Figure 6 shows the experimental setup. We use a 7-DoF robotic arm with a two-finger gripper, and a camera placed above the shoulder, which provides monocular RGB images. For our baseline PI\(^2\) policy, we closely follow the setup of [9] which uses an IMU sensor in the door handle to provide both a cost and feedback as part of the state of the controller. In contrast, in our approach we remove this sensor both from the state representation provided to PI\(^2\) and in our reward replace the target IMU state with the output of a deep neural network.

1) Data: We experiment with a range of different demonstrations from which we derive our reward function, varying both the source demo (human vs robotic), the number of subgoals we extract, and the appearance of the door. We record monocular RGB images on a camera placed above the shoulder of the arm. The door is cropped from the images, and then the resulting image is re-sized such that the shortest side is 299 dimensional with preserved aspect ratio. The input into our convolutional feature extractor [30] is the 299x299 center crop.

2) Qualitative Analysis: We evaluate our reward functions qualitatively by plotting our perceptual reward functions below the demonstrations with a variety of door types and demonstrators (e.g. robot or human). As can be seen in Fig. 7 and in real experiments Fig. 8, we show that the reward functions are useful to a robotic arm while only showing human demonstrations as depicted in Fig. 13 in [28]. Moreover we exhibit robustness variations in appearance.

3) Quantitative Analysis: In comparing the success rate of visual reward versus a baseline PI\(^2\) method that use s the ground truth reward function obtained by instrumenting the door with an IMU. We run PI\(^2\) for 11 iterations with 10 sampled trajectories at each iteration. As can be seen in Fig. 8, we obtain similar convergence speeds to our baseline.
**IV. Conclusion**

In this paper, we present a method for automatically identifying important intermediate goal given a few visual demonstrations of a task. By leveraging the general features learned from pre-trained deep models, we propose a method for rapidly learning an incremental reward function from human demonstrations which we successfully demonstrate on a real robotic learning task.

We show that pre-trained models are general enough to be used without retraining. We also show there exists a small subset of pre-trained features that are highly discriminative even for previously unseen scenes and which can be used to reduce the search space for future work in unsupervised subgoal discovery.

In this work, we studied imitation in a setting where the viewpoint of the robot/demonstrator is fixed. An interesting avenue of future work would be to analyze the impact of changes in viewpoint. The ability to learn from a learn from broad diverse sets of experience also ties into the goal of lifelong robotic learning. Continuous learning using unsupervised rewards promises to substantially increase the variety and diversity of experiences, resulting in more robust and general robotic skills.

Acknowledgments: We thank Vincent Vanhoucke for helpful discussions and feedback, Mrinal Kalakrishnan and Ali Yahya for indispensable guidance throughout this project and Yevgen Chebotar for his PI$^2$ implementation. We also thank the anonymous reviewers for their feedback and constructive comments.