Rank2Reward: Learning Shaped Reward Functions from Passive Video

Daniel Yang[1], Davin Tjia[2], Jacob Berg[2], Dima Damen[3], Pulkit Agrawal[1], Abhishek Gupta[2]

***Abstract*— Teaching robots novel skills with demonstrations via human-in-the-loop data collection techniques like kinesthetic teaching or teleoperation puts a heavy burden on human supervisors. In contrast to this paradigm, it is often significantly easier to provide raw, action-free visual data of tasks being performed. Moreover, this data can even be mined from video datasets or the web. Ideally, this data can serve to guide robot learning for new tasks in novel environments, informing both "what" to do and "how" to do it. A powerful way to encode both the "what" and the "how" is to infer a well-shaped reward function for reinforcement learning. The challenge is determining how to ground visual demonstration inputs into a well-shaped and informative reward function. We propose a technique `Rank2Reward` for learning behaviors from videos of tasks being performed without access to any low-level states and actions. We do so by leveraging the videos to learn a reward function that measures incremental "progress" through a task by learning how to temporally rank the video frames in a demonstration. By inferring an appropriate ranking, the reward function is able to guide reinforcement learning by indicating when task progress is being made. This ranking function can be integrated into an adversarial imitation learning scheme resulting in an algorithm that can learn behaviors without exploiting the learned reward function. We demonstrate the effectiveness of `Rank2Reward` at learning behaviors from raw video on a number of tabletop manipulation tasks in both simulations and on a real-world robotic arm. We also demonstrate how `Rank2Reward` can be easily extended to be applicable to web-scale video datasets. Code and videos are available at https://rank2reward.github.io**

I. Introduction

Robot learning via reinforcement learning (RL) directly in the real world show the promise of continual improvement, with minimal modeling assumptions [1]–[5]. However, the promise of plug-and-play reinforcement learning hides a significant challenge — *where do reward functions come from?* Reward function design is a non-trivial task; rewards must be unbiased while still guiding exploration toward optimal behaviors with "dense supervision". While this may be possible to provide in certain simulation environments [6], [7], it is much more challenging in the real world.

A natural strategy for reward function design is data-driven algorithms such as inverse RL [8], [9] for reward inference. These methods rely on expert demonstrations to infer reward functions, learning reward functions that maximize the likelihood of demonstrations while being uninformative about other trajectories [10]–[12]. This approach can be powerful but typically suffers from two significant challenges - (1) demonstration data in the form of state-action tuples can be challenging to obtain without expensive techniques such as kinesthetic teaching or teleoperation, and (2) learned reward functions may explain the expert data well, but may not be "well-shaped", providing no guidance for exploration. For data-driven reward functions to be a practical alternative to hand-crafted reward functions, they must both be easy to provide and make policy optimization easy.

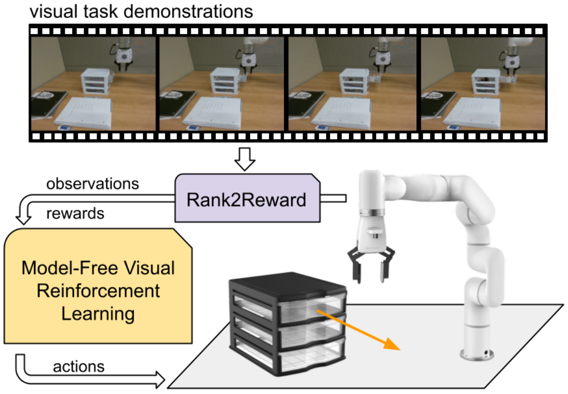
visual task demonstrations
observations
rewards
Rank2Reward
Model-Free Visual Reinforcement Learning
actions

Fig. 1: Depiction of the problem setting in `Rank2Reward` - inferring well-shaped and calibrated reward functions from video demonstrations that enable effective policy optimization.

As opposed to expensive forms of demonstrations such as kinesthetic teaching or teleoperation, a natural and easy-to-obtain source of interaction data is video observations of tasks being performed. These are abundantly present in computer vision datasets [13]–[16]. This data contains both "what" tasks are interesting in an environment and "how" to accomplish these tasks. In this work, we show how raw videos of tasks being performed can serve as supervision for a simple reward learning method that satisfies both desiderata above - (1) is easy to provide, and (2) effectively guides exploration for RL by providing informative shaping.

The key insight that we exploit in this work is that video demonstrations typically make monotonic progress toward a goal. Under this assumption, a natural reward function is simply how much *progress* has been made along a successful trajectory. This framing allows us to recast reward function learning as the problem of learning to order frames within a video. By predicting an ordering of video frames, we can infer a notion of progress along a trajectory using techniques from learning from preferences [17], [18]. Since the progress along a trajectory is strictly monotonic, the resulting reward function is well-shaped for policy optimization.

Notably, since the ranking function is trained purely on expert video data, it cannot meaningfully provide a reward

[1] D. Yang and P. Agrawal are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA {`dxyang, pulkitag`}`@mit.edu`

[2] D. Tjia, J. Berg, and A. Gupta are with the Department of Computer Science, University of Washington, Seattle, WA, USA {`davin05, jacob33, abhgupta`}`@cs.washington.edu`

[3] D. Damen is with the School of Computer Science, University of Bristol, Bristol, UK `dima.damen@bristol.ac.uk`

signal to states and trajectories not covered in the expert dataset. To remedy this, we show how to formulate policy search with learned ranking rewards as a constrained policy optimization problem in which the policy is constrained to stay close to the distribution of expert data. We show how this can be further simplified to a weighted variant of adversarial imitation learning, alternating between (1) learning a discriminator to differentiate expert from on-policy trajectories and (2) policy search using reward as a combination of the learned ranking function and the learned discriminator. This results in a simple yet performant algorithm for policy learning from video demonstrations without actions - rank frames in video demonstrations and use this ranking to reweight adversarial imitation learning. We show the efficacy of this learned reward function to guide policy learning for both tasks in simulation and real-world robotic manipulation.

II. Related works

Inverse RL: Inverse reinforcement learning (IRL), [8], [9], [19] aims to infer rewards from demonstrations such that demonstrations are scored highly, while *other* trajectories are scored suboptimally. Various IRL techniques aim to instantiate this idea using techniques such as max-margin planning [20], maximum entropy inverse RL [12], [21], [22], and feature matching [19]. Generative adversarial imitation learning (GAIL) [10] and similar methods [11], [12], [23], proposed treating IRL as an adversarial game. A challenge most IRL techniques face is that while the reward is correct at optimality [24], [25], the rewards are poorly shaped, offering no learning signal. We show that a simple ranking based objective can allow us to easily infer well-shaped rewards.

Imitation from observation: Imitation-from-observation considers how to learn from high dimensional *action-free* demonstrations such as videos [26]. One class of these imitation-from-observation techniques tries to label the observation-only dataset with inferred actions from an inverse dynamics model, and run standard imitation learning [27]–[29]. Specifically for tasks with human hands, other methods infer actions using off the shelf hand and object pose estimation algorithms before applying standard imitation learning [30]–[32], but these methods require instrumentation of the environment with calibrated, depth-sensing cameras and presume known pose detectors. [33], [34] learn video classifiers and use these as rewards for reinforcement learning. This can be effective at deciding "what" to do but fails to provide shaped rewards. Other techniques to learn from videos include learning representations from video to assign rewards [35], using temporal contrastive learning [36]–[38], and regressing onto temporal differences between frames to provide an exploration bonus reward for RL [39]. While this can be effective in certain scenarios, these learned distances are inaccurate out of distribution and are prone to exploitation. As opposed to these methods, `Rank2Reward` aims to provide a reward function that is *both* well-shaped and calibrated on out-of-distribution states. The closest work in this line to ours is Time Contrastive Networks (TCN) [35]. While TCN may learn a useful representation from contrastive learning across time and viewpoints, this embedding space does not contain any notion of progress towards achieving a goal, as distance in input space does not correspond to moving towards or away from the goal, and relies on performing feature tracking on a specific expert trajectory which requires temporal alignment. In contrast, `Rank2Reward` learns an ordered ranking space which both encodes progress towards the goal *and* is agnostic to time required to reach the state.

Ranking-based approaches in video understanding: Modeling the evolution of human actions in video through temporal frame ranking was first proposed in [40]. Using a ranking loss, the approach learns a representation of an action that successfully orders the frames in that video. Subsequently, ranking was successfully used for end-to-end classification [41], modeling progression to completion [42] as well as skill development [43]. In this work, we take inspiration from using the temporal frame ranking loss.

Ranking-based reward learning: Our work leverages a frame ranking objective to infer well-shaped rewards. The idea of ranking based objectives being used has been recently explored in the context of reinforcement learning from human feedback [17], [18], [44]–[46]. This can be used to leverage binary comparisons, provided by a human, to train a reward function that can be used for RL. In contrast, our work does not need any external human comparisons or preferences. Instead, we simply rank frames within a video according to their temporal progression.

III. Preliminaries

Consider a robot learning how to perform a task as a finite horizon Markov decision process (MDP), $\mathcal{M}$, consisting of the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \rho_0, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(s'|s, a)$ is the transition function, ρ_0 is the initial state distribution, and γ is the discount factor. In this work, $\mathcal{S}$ can be high-dimensional images, but we assume that the environment is fully observable, thereby retaining the Markov property. This can be relaxed by either stacking frames or leveraging history-conditioned policies [47].

Here, we want our robot to learn a policy, π^*, to complete a particular task. However, the true reward function $r(s)$ is not available to the learning agent. Instead, we have a set of N expert demonstration trajectories, $\mathcal{D}^e = \{\tau_k\}_{k=1}^N$ where each trajectory consists of a sequence, $\tau_k = \{s_0^k, s_1^k, \ldots, s_T^k\}$. Without loss of generality, we assume that the expert dataset is drawn IID from some expert policy π^e, with corresponding state-action marginal $d^e(s, a)$. Unlike typical imitation learning settings, no actions are available.

Since the reward function is unknown, the goal is first to infer an appropriate reward function $\hat{r}(s)$ from the expert data and then use this for policy optimization as $\pi^* \leftarrow \arg\max_\pi \mathbb{E}\left[\sum_t \gamma^t \hat{r}(s)\right]$ similar to standard reinforcement learning settings. Using the notation of the state-action marginal, $d^\pi(s, a)$, we can rewrite this policy optimization objective as $\pi^* \leftarrow \arg\max_\pi \mathbb{E}_{d^\pi(s,a)}\left[\hat{r}(s)\right]$, where $d^\pi(s, a) = (1-\gamma)\sum_{t=0}^\infty P(s_t = s, a_t = a | s_0 \sim \rho_0(s), a_t \sim \pi(a_t|s_t), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t))$ is the standard state-action

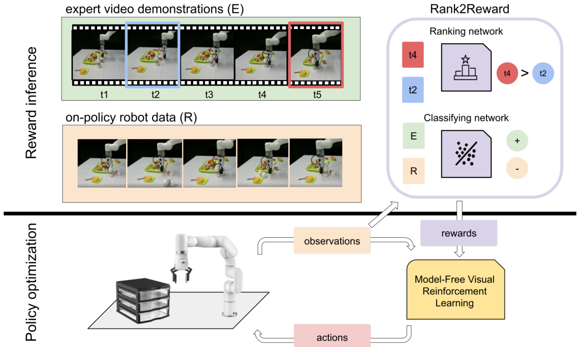

Fig. 2: Rank2Reward- given video demonstrations from a human supervisor, Rank2Reward learns a ranking function that temporally orders frames and a classifier D_ϕ between expert and on-policy data. Combined, they yield a well-shaped reward function for real-world RL.

occupancy measure. Note that as in most IRL settings, the process of inferring $\hat{r}(s)$ and learning π^* can be interleaved.

IV. Rank2Reward: Learning shaped reward functions by frame ranking

We propose a simple and scalable method for reward function inference from raw video demonstrations without requiring any actions. Our proposed technique, Rank2Reward as shown in Fig. 2, can learn *well-shaped* reward functions that guide exploration for challenging tasks while being resilient to exploitation by the policy during reinforcement learning. The key idea is to learn how to order frames temporally in a trajectory. In doing so, we can infer whether states visited by a policy are making *progress* along a trajectory, therefore learning policies that maximize progress. We then show how this objective in itself is prone to exploitation during policy learning and propose a constrained policy learning objective that prevents this exploitation. The resulting algorithm resembles a weighted adversarial imitation learning, providing well-shaped rewards for learning.

A. Learning a measure of progress by ranking

The key assumption that we make in this work is that the provided demonstrations are optimal and that optimal trajectories make monotonic progress towards the goal. This suggests that the true reward for the task (which is unknown) is positive and non-zero for all states, a common occurrence in a huge variety of problems especially goal-reaching tasks [48]–[50]. For most problems, we can find a valid reward function satisfying this assumption. Formally stated:

Assumption 1. *True (unknown) reward $r(s, a) > \epsilon$, where $\epsilon > 0$. This suggests that value functions of optimal policies $V^*(s)$ are monotonically increasing.*

However, this true reward is unknown, and IRL techniques can recover ill-shaped and hard-to-optimize rewards $\hat{r}(s)$. Instead, we can leverage Assumption 1 to directly measure whether states are making *progress* along a trajectory. More specifically, we make the observation that progress along a trajectory can be measured by simply learning a function that can rank different image frames in a trajectory according to their temporal ordering. To do so, we build on recent work in preference modeling [17], [18] to learn measures of progress by learning how to rank pairs of frames in terms of their natural ordering. Preference modeling methods such as the Bradley-Terry model [51] aim to learn a utility function $\hat{u}(s)$ such that the likelihood of "preferring" a state s_i^k over a different state s_j^k for some expert trajectory τ_k, is given by $p(s_i^k > s_j^k) = \frac{\exp \hat{u}_\theta(s_i^k)}{\exp \hat{u}_\theta(s_j^k) + \exp \hat{u}_\theta(s_i^k)}$.

Framing temporal ranking as preference modeling allows us to, without any additional human annotation, generate a set of preference labels for trajectories in the expert dataset $\mathcal{D}^e$ given a sampled pairs of states, (s_i^k, s_j^k) along the same expert trajectory. Along τ_k, s_i^k is preferred over s_j^k if it occurs later (i.e., $i > j$). According to the Bradley-Terry model, this suggests that s_i^k should have a higher reward than s_j^k (i.e., $\hat{u}_\theta(s_i^k) > \hat{u}_\theta(s_j^k)$), thereby incentivizing progress along a trajectory. This paradigm naturally lends itself to a training objective for $\hat{u}_\theta$ by simply finding parameters θ that maximize likelihood over the preference model, resulting in simple cross-entropy classification:

$$\max_\theta \mathbb{E}_{\substack{\tau_k \sim \mathcal{D}^e \\ s_i^k, s_j^k, \sim \tau_k}} \Bigg[\mathbb{1}_{i>j} \left[\log \frac{\exp \hat{u}_\theta(s_i^k)}{\exp \hat{u}_\theta(s_j^k) + \exp \hat{u}_\theta(s_i^k)} \right] + \mathbb{1}_{i<j} \left[\log \frac{\exp \hat{u}_\theta(s_j^k)}{\exp \hat{u}_\theta(s_j^k) + \exp \hat{u}_\theta(s_i^k)} \right] \Bigg] \quad (1)$$

This training objective results in learning a utility function $\hat{u}_\theta$ that is monotonically increasing along a trajectory. This utility function can naturally be converted into a reward function by noting that policy optimization aims to learn policies that maximize the likelihood (and thereby log-likelihood) of progress. By utilizing the likelihood under the Bradley-Terry model and setting the utility of the start of the trajectory $\hat{u}(s_0) = 0$, the likelihood that a state s makes progress over the initial state s_0 can be written as $p(s > s_0) = \frac{\exp \hat{u}_\theta(s)}{\exp \hat{u}_\theta(s_0) + \exp \hat{u}_\theta(s)} = \frac{1}{1 + \exp(\hat{u}_\theta(s_0) - \hat{u}_\theta(s))} = \frac{1}{1 + \exp(-\hat{u}_\theta(s))}$. This is essentially just a sigmoid function applied to the learned ranking utilities $\hat{u}(s)$, appropriately normalizing it. For the sake of notation, we denote the likelihood of making progress $p_{\text{RF}}(s) = \frac{1}{1 + \exp(-\hat{u}_\theta(s))}$.

A policy that maximizes the log-likelihood of progress of all states can then be obtained as $\max_\pi \mathbb{E}_\pi [\log \cup_{i=1}^n p(s_i > s_0)] = \mathbb{E}_\pi [\sum_{i=1}^n \log p_{\text{RF}}(s_i))] = \mathbb{E}_{s,a \sim d^\pi(s,a)} [\log p_{\text{RF}}(s)]$, where we use the state-action marginal form of the value function. This objective is amenable to any standard policy optimization framework such as model-free reinforcement learning [52]–[54], with $\hat{r}(s) = \log p_{\text{RF}}(s)$ as the reward. Note that since it is monotonically increasing along optimal trajectories, p_{RF} is a value function-like measure rather than a sparse reward type of measure. As discussed in prior work [48], optimizing value function-like measures can lead to policies that both learn quickly and have bounded suboptimality. The ranking function $\hat{u}_\theta(s)$ can be trained via Eq 1 solely from expert trajectories, and the policy can be then optimized with $\hat{r}(s)$.

B. Incorporating learned rankings into policy optimization

Naively optimizing this objective naturally leads to the policy exploiting the learned reward function in "out-of-distribution" states. Since the reward function $\hat{r}(s)$ has only been learned on states from the expert dataset, $s \sim \mathcal{D}^e$, it may overestimate rewards at other states leading to arbitrarily incorrect policies. To remedy this, we incorporate pessimism [55]–[57] — penalizing the policy for deviation from the training distribution. Given the expert dataset $\mathcal{D}^e$ and it's corresponding state marginal $d^e(s)$, as well as the current policy π and it's state marginal $d^\pi(s)$, we can formulate a pessimistic policy optimization objective as:

$$\max_\pi \mathbb{E}_{s \sim d^\pi, a \sim \pi(a|s)} \left[\log p_{\text{RF}}(s)\right] - \alpha D_{KL}(d^\pi(s), d^e(s)) \quad (2)$$

This objective aims to maximize the likelihood of progress as defined in Sec. IV-A. However, it does so while ensuring that the state marginal of the policy remains close to the expert data distribution via a penalty on the divergence $D_{\text{KL}}(d^\pi(s), d^e(s)) = \mathbb{E}_{s \sim d^\pi(s)}\left[\log \frac{d^\pi(s)}{d^e(s)}\right]$ between the marginal densities of the policy and the expert data.

This objective is challenging to optimize since the likelihoods under the expert marginal data distribution $d^e(s)$ and the policy marginal distribution $d^\pi(s)$ are both unknown and require density estimation [58]–[60]. Instead, we will show that the objective in Eq. (2) can be recast as a weighted adversarial imitation learning algorithm that circumvents explicit density estimation. By substituting the definition of the KL divergence in Eq. (2) we have the following objective:

$$\begin{aligned} &\max_\pi \mathbb{E}_{s,a \sim d^\pi} \left[\log p_{\text{RF}}(s)\right] - \alpha \mathbb{E}_{s \sim d^\pi(s)}\left[\log \frac{d^\pi(s)}{d^e(s)}\right] \\ &= \mathbb{E}_{s,a \sim d^\pi}\left[\log\left(p_{\text{RF}}(s)\left(\frac{d^e(s)}{d^\pi(s)}\right)^\alpha\right)\right] \end{aligned} \quad (3)$$

To estimate the density ratio $\frac{d^e(s)}{d^\pi(s)}$, we can make use of the fact that despite the likelihoods of $d^e(s)$ and $d^\pi(s)$ not being known, given samples from distributions $d^e(s)$, $d^\pi(s)$, a classifier trained to distinguish between this samples can be used to estimate a density ratio. A classifier $D_\phi(s)$ trained to distinguish between $d^e(s)$, $d^\pi(s)$ can provide us $\frac{d^e(s)}{d^\pi(s)} = \frac{D_\phi(s)}{1-D_\phi(s)}$ [61]. This reduces to:

$$\max_\pi \mathbb{E}_{s,a \sim d^\pi}\left[\log\left(p_{\text{RF}}(s)\left(\frac{D_\phi(s)}{1 - D_\phi(s)}\right)^\alpha\right)\right] \quad (4)$$

This equivalence suggests a simple algorithm for optimizing Eq. (4) - alternate between (1) training a classifier $D_\phi(s)$ to distinguish between states drawn from the expert video demonstrations and on-policy data collected by the policy π using standard binary classification with cross-entropy, and (2) perform policy optimization combining the learned classifier $D_\phi(s)$ together with the learned ranking function in $\hat{r}(s) = p_{\text{RF}}(s)\left(\frac{D_\phi(s)}{1-D_\phi(s)}\right)^\alpha$. This is similar to adversarial imitation learning methods like GAIL [10] but weighted with a learned ranking function p_{RF}.

Algorithm 1 Rank2Reward

```
Require: Expert demonstration data $\mathcal{D}^e = \{\tau_k\}_{k=1}^N$
Initialize policy $\pi$, empty replay buffer $\mathcal{D}_{\text{RB}}$
Initialize utility $\hat{u}_\theta$ and classifier $D_\phi$ functions for $\hat{r}(s)$.
// Train the utility ranking function $\hat{u}_\theta$
for step $n$ in $\{1, \ldots, N_{\text{ranking}}\}$ do
    Sample state pairs $s_i^k, s_j^k$ from each trajectory, $\tau_k$
    Learn $\hat{u}_\theta$ with batch $\{(s_{t_1}, s_{t_2})_k\}_{k=1}^{bs}$ using Eq. (1)
end for
// Joint policy optimization and reward learning
for step $n$ in $\{1, \ldots, \text{N}\}$ do
    With $\pi$, collect transitions $\{\tau_l\}_{l=1}^M$ and store in $\mathcal{D}_{\text{RB}}$
    if $n \,\%\,$ reward_update_frequency $== 0$ then
        Sample batch of states $s_e$ from expert $\mathcal{D}^e$
        Sample batch of states $s_\pi$ from replay buffer $\mathcal{D}_{\text{RB}}$
        Update $D_\phi$ to classify $s_e$ from $s_\pi$ with BCE.
    end if
    Sample batch of transitions $s_\pi$ from $\mathcal{D}_{\text{RB}}$
    Update $\pi$ to maximize returns using Eq. (5)
end for
```

C. Practical algorithm overview

In Algorithm 1, we show how our method for learning a reward function can be used with any off the shelf reinforcement learning algorithm - here, off-policy learning methods [53], [54] for data efficiency. Notably, the ranking component $p_{\text{RF}}(s)$ of the reward can be learned offline solely from expert data, independent of learning the policy, and only the classification component $D_\theta(s)$ depends on both the expert data and data collected by the current learned policy. Our final simplified learned, estimated reward function is:

$$\hat{r}(s) = \log p_{\text{RF}}(s) + \alpha\Big(\log D_\phi(s) - \log(1 - D_\phi(s))\Big) \quad (5)$$

V. Experimental Results

In our experiments, we evaluate our proposed technique for learning reward functions from video demonstrations on simulated and real-world manipulation tasks. We also experiment with scaling our approach to internet-scale in-the-wild data from Ego4D [13]. In simulation, we leverage a model-free RL method for learning from images, DrQv2 [53], [62], while in the real world, we leverage a data-efficient actor-critic technique [54], [63]. All experiments use a frozen pre-trained visual feature extractor from [64].

A. Evaluation Environments

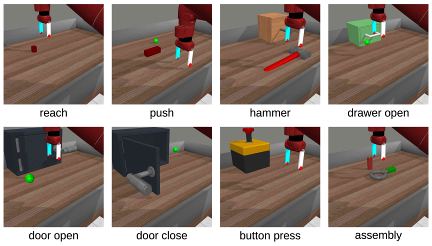

Fig. 3: Meta-world [6] environments: (1) reach, (2) push, (3) hammer, (4) drawer open, (5) door open, (6) door close, (7) button press, (8) assembly

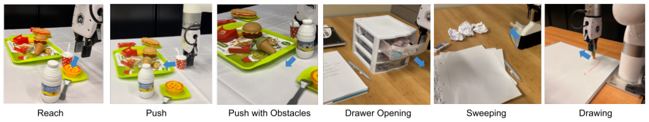

Fig. 4: Real-world environments including standard tasks (reaching and pushing), tasks where exploration is non-trivial (pushing with obstacles and drawer opening), and tasks where state estimation is non-trivial (sweeping and drawing). The blue arrows indicate the directions to go.

Meta-world [6]: This simulation benchmark consists of a table-top Sawyer robot arm, as shown in Fig. 3. We instantiate environments with random initial states and use image-based observations. We compare our method against baselines by measuring the average episodic return based on the hand-defined rewards available with the benchmark.

Real-World xArm Environment: We utilize a table-top mounted 5 DoF xArm5 manipulator. We perform end-effector positional control, where the action space is normalized delta positions, and use image-based observations. We test `Rank2Reward` on 6 real-world tasks, as shown in Fig. 4. Our more complex tasks highlight situations where exploration is non-trivial, techniques like object tracking are ineffective, and reward specification overall is difficult.

B. Baseline Comparisons

We compare `Rank2Reward` with the following baselines: **(1) GAIL** [10] Reward function is a classifier of whether state comes from expert demonstration trajectories. This is similar to our method *without* the ranking term and is akin to ensuring the policy state visitation distribution matches that of the expert data. **(2) AIRL** [22] The reward function is similar to GAIL above but scaled with $r_{\text{AIRL}} = \log(r_{\text{GAIL}}) - \log(1 - r_{\text{GAIL}})$. **(3) VICE** [11] The reward function is similar to GAIL above but instead of learning a classifier of whether the state comes from the expert demonstration *trajectories*, VICE classifies whether a state is the expert *goal state*. **(4) SOIL** [29] Learn an inverse dynamics model and uses the data to infer actions for the expert states and uses this for imitation learning. **(5) TCN** [35] We utilize the single-view variant of TCN to learn an embedding space and perform feature tracking with an expert demonstration to generate rewards. **(6) ROT** [65] A recent method using on optimal transport-based trajectory matching. To compare in a similar setting, we do not utilize the behavioral cloning initialization and regularization components of ROT, as our method presumes no access to expert actions. **(7) Ranking only** This ablation uses only the ranking function, without the adversarial classifier, to generate reward.

C. Simulated Experiments

To quantify performance, we examine the average episodic return from 10 evaluation episodes averaged over 3 seeds as shown in Fig. 5. While all methods achieve non-zero returns, our method learns quickly and more effectively than baselines for state-only imitation learning and reward assignment. Performance on `hammer`, `drawer open`, `button press`, and `assembly` is significantly better than baselines, while learning curves on `reach`, `push`, `door open` show comparable or slightly better performance between our method and AIRL [22]. Notably, TCN [35] and ROT [65] are methods designed for learning from video whereas most other methods focus on low dimensional states which makes their comparisons more insightful. Our method performs similarly or slightly better than ROT in most environments. However, in `door open` and `push`, ROT noticeably outperforms our method, while in the simplest domain `reach`, ROT struggles to achieve high rewards. In all domains TCN performs similarly to some baseline methods, but does not achieve comparable performance to `Rank2Reward`.

D. Real-World Robotic Experiments

We evaluate `Rank2Reward` on real-world tasks (Sec. V-A) and compare our performance with GAIL [10] as shown in Table I. We use different success metrics for our evaluation tasks — distance to goal in reaching and pushing, number of environment steps required to learn the task for pushing with obstacle, drawer opening, and drawing, and percent of objects not successfully swept for sweeping. `Rank2Reward` is able to effectively learn behaviors across real-world robotics domains purely from image observations and video demonstrations. Our baselines are unable to reliably learn any of our more complex tasks beyond reaching and pushing, while `Rank2Reward` can learn in under 2 hours of real-world interaction even for the more challenging tasks.

E. Ego4D experiments

To show that `Rank2Reward` is a generalizable and scalable paradigm, we apply our approach to the Hand & Object Interactions data from Ego4D [13]. From 27,000 segments processed at 10 fps, we have 2.16 million frames. We train with 20,000 segments, and leave the rest for evaluation. From each clip, we utilize the last frame as the goal frame and learn a ranking component conditioned on the goal frame.

For the discriminator, we sample a positive frame from the same clip as the goal and a negative frame from a different clip as the goal, and train D_ϕ to classify whether a given frame and the goal frame come from the same video. These negative frames with goals that do not match can be thought of as counterfactual examples. We randomly select four segments from the evaluation set and present the output of `Rank2Reward` when evaluated with the true goal and a counterfactual goal in Fig. 6. When a state is evaluated with the true goal, reward is overall increasing whereas when evaluated with a counterfactual goal, it is both not increasing and has an overall lower value. Such a defined and well-shaped reward landscape on diverse, real-world data holds promising value in lowering the difficulty of providing expert data to learn robotic tasks.

F. Reward function analysis

We visualize the shaping of our reward function over policy optimization in a continuous 2D two-wall maze environment, where the agent starts at the top left and the goal is at the bottom right. Given 20 expert demonstrations (Fig. 7a), we visualize the fixed ranking function over the landscape where greedily moving towards highly ranked states does not

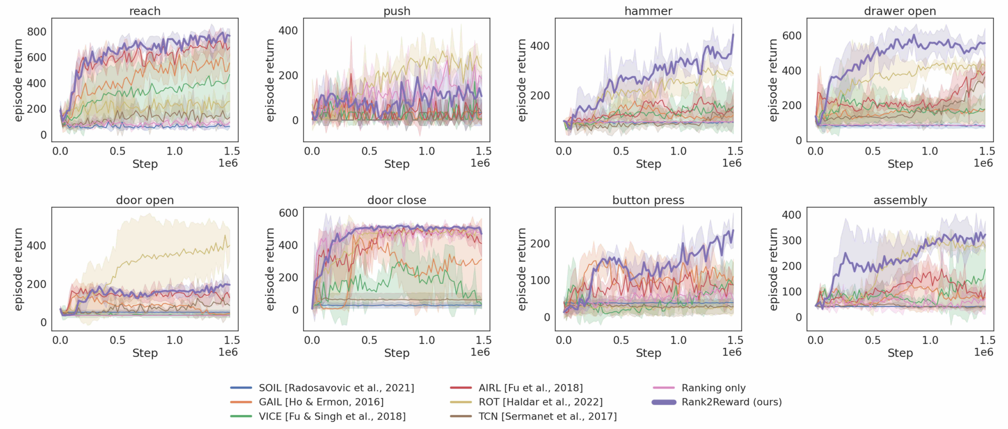

Fig. 5: Episodic return from 10 evaluation episodes averaged over 3 seeds plotted over the course of training the DrQ-v2 agent for 1.5 million steps. Higher is better. Our method (purple) distinctly outperforms other methods on `hammer`, `drawer open`, `button press`, and `assembly`, while performing similarly to the best baseline with `reach`, and `door close` and worse than the best baseline in `push` and `door open`.

Task	Reaching	Pushing	Pushing w/ Obst	Drawer Opening	Sweeping	Drawing
Metric	$L2$-dist	$L2$-dist	StS	StS	Incompletion	StS
`Rank2Reward`	0.31	0	4391	6079	0%	79
GAIL	0.43	1.26	FAILED	FAILED	60%	FAILED

TABLE I: Evaluation results of real-world training using `Rank2Reward` against the baseline GAIL. For all metrics below, lower is better. $L2$-dist is the $L2$-distance in centimeters (cm) from the goal state at the end. Steps to Success (StS) measures the number of environment steps required to successfully learn the task. Incompletion refers to the percentage of objects that were not successfully swept off the table.

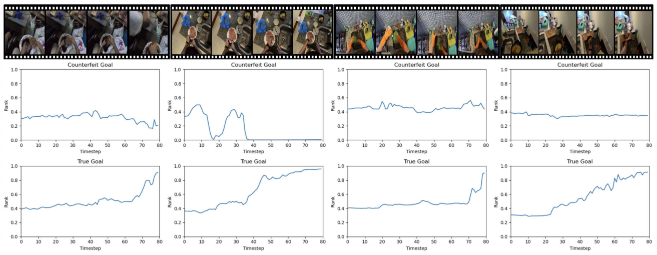

Fig. 6: Visualization of `Rank2Reward` output when four random input videos from Ego4D are evaluated for the true goal and a counterfactual goals that do not correspond to the input.

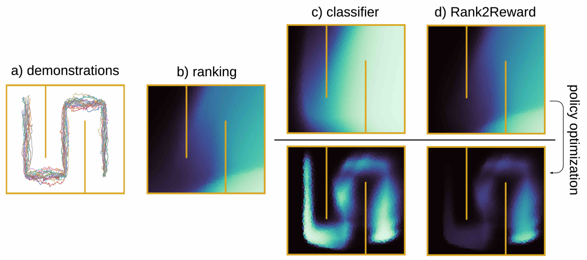

Fig. 7: Visualization of different components of `Rank2Reward` in a two-wall 2D maze environment where state is position. (a) Demonstration data (b) $p_{\text{RF}}(s)$. Note the spurious high values assigned to regions not visited by the expert. (c) $D_\phi(s)$, before and after policy optimization. Note the induced down weighting of out of expert distribution data (d) `Rank2Reward`, $D_\phi(s) * p_{\text{RF}}(s)$, before and after policy optimization.

necessarily lead to the goal (Fig. 7b). We show the evolution of the classifier during policy optimization (Fig. 7c, top and bottom). Initially, the classifier gives higher values to states on the right half of the maze with some slight shaping from states likely to be stumbled upon by random exploration. However, over RL the classifier better distinguishes states similar to the expert demonstrations from other states. When combined with our ranking function which was well-shaped but had spuriously high output for out of distribution inputs, we see that our final reward function is both well-shaped and well-defined across the whole state space (Fig. 7d, bottom).

VI. Conclusion and Limitations

In this work, we have shown that learning how to rank visual observations from a demonstration can be used to infer well-shaped reward functions. `Rank2Reward` is simple to use, easily integrating into many popular off the shelf RL algorithms. By combining how to rank with how to classify expert demonstration data from policy-collected data, our learned reward function is interpretable yet performant.

Limitations and Future Work Notably, there is an embodiment shift between human demonstration videos like those found in [13], [14] and our robot manipulators. To make use of internet-scale data, we must use representations that generalize across manipulators, perhaps building on [38], [66], [67]. Also, the rewards trained here are still single-task and it would be unscalable to have a different reward and agent for every task. Next, the classifier D_ϕ is sensitive to changes in the background and dynamic scenes. Real world deployment will require further invariant representations during reward inference.

VII. Acknowledgements

This research was supported by NSF Robust Intelligence Grant 2212310, the Hyundai Motor Company, and an Amazon Research Award. We thank members of the WEIRD lab at UW and the Improbable AI lab at MIT for helpful discussions and feedback and the HYAK compute cluster as well as MIT Supercloud for compute support.

References

[1] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[2] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[3] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[4] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *Robotics: Science and Systems*, 2022.

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

[6] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[7] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, 2020.

[8] S. Russell, "Learning agents for uncertain environments," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 101–103.

[9] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[10] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[11] J. Fu, A. Singh, D. Ghosh, L. Yang, and S. Levine, "Variational inverse control with events: A general framework for data-driven reward definition," 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., 2018.

[12] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, ser. JMLR Workshop and Conference Proceedings, M. Balcan and K. Q. Weinberger, Eds., vol. 48. JMLR.org, 2016, pp. 49–58. [Online]. Available: http://proceedings.mlr.press/v48/finn16.html

[13] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, "Ego4d: Around the World in 3,000 Hours of Egocentric Video," in *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

[14] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018.

[15] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The" something something" video database for learning and evaluating visual common sense," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5842–5850.

[16] J. Materzynska, T. Xiao, R. Herzig, H. Xu, X. Wang, and T. Darrell, "Something-Else: Compositional Action Recognition with Spatial-Temporal Interaction Networks," *arXiv:1912.09930 [cs]*, Sep. 2020, arXiv: 1912.09930. [Online]. Available: http://arxiv.org/abs/1912.09930

[17] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4299–4307.

[18] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 730–27 744.

[19] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Twenty-first international conference on Machine learning - ICML '04*. Banff, Alberta, Canada: ACM Press, 2004, p. 1.

[20] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 729–736.

[21] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, D. Fox and C. P. Gomes, Eds. AAAI Press, 2008, pp. 1433–1438.

[22] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adverserial inverse reinforcement learning," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=rkHywl-A-

[23] A. Singh, L. Yang, C. Finn, and S. Levine, "End-to-end robotic reinforcement learning without reward engineering," in *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019*, A. Bicchi, H. Kress-Gazit, and S. Hutchinson, Eds., 2019. [Online]. Available: https://doi.org/10.15607/RSS.2019.XV.073

[24] Q. Cai, M. Hong, Y. Chen, and Z. Wang, "On the global convergence of imitation learning: A case for linear quadratic regulator," *arXiv preprint arXiv:1901.03674*, 2019.

[25] M. Chen, Y. Wang, T. Liu, Z. Yang, X. Li, Z. Wang, and T. Zhao, "On computation and generalization of generative adversarial imitation learning," in *International Conference on Learning Representations (ICLR)*, 2020.

[26] F. Torabi, G. Warnell, and P. Stone, "Recent advances in imitation learning from observation," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, S. Kraus, Ed. ijcai.org, 2019, pp. 6325–6331. [Online]. Available: https://doi.org/10.24963/ijcai.2019/882

[27] K. Schmeckpeper, O. Rybkin, K. Daniilidis, S. Levine, and C. Finn, "Reinforcement learning with videos: Combining offline observations with interaction," *CoRL*, 2020.

[28] B. Baker, I. Akkaya, P. Zhokov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune, "Video pretraining (vpt): Learning to act by watching unlabeled online videos," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 639–24 654, 2022.

[29] I. Radosavovic, X. Wang, L. Pinto, and J. Malik, "State-only imitation learning for dexterous manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 7865–7871.

[30] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "Dexmv: Imitation learning for dexterous manipulation from human videos," in *European Conference on Computer Vision*. Springer, 2022, pp. 570–587.

[31] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," in *Robotics: Science and Systems XVIII*. Robotics: Science and Systems Foundation, 2022. [Online]. Available: https://www.roboticsproceedings.org/rss18/p026.pdf

[32] K. Shaw, S. Bahl, and D. Pathak, "Videodex: Learning dexterity

from internet videos," in *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 2022, pp. 654–665. [Online]. Available: https://proceedings.mlr.press/v205/shaw23a.html

[33] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, "Concept2robot: Learning manipulation concepts from instructions and human demonstrations," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1419–1434, 2021.

[34] A. S. Chen, S. Nair, and C. Finn, "Learning generalizable robotic reward functions from "in-the-wild" human videos," in *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, D. A. Shell, M. Toussaint, and M. A. Hsieh, Eds., 2021. [Online]. Available: https://doi.org/10.15607/RSS.2021.XVII.012

[35] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1134–1141.

[36] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," in *International Conference on Learning Representations (ICLR)*, 2023.

[37] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," in *International Conference on Machine Learning (ICML)*, 2023.

[38] M. Alakuijala, G. Dulac-Arnold, J. Mairal, J. Ponce, and C. Schmid, "Learning reward functions for robotic manipulation by observing humans," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[39] J. Bruce, A. Anand, B. Mazoure, and R. Fergus, "Learning about progress from experts," in *International Conference on Learning Representations (ICLR)*, 2023.

[40] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[41] B. Fernando and S. Gould, "Learning end-to-end video classification with rank-pooling," in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, 2016, pp. 1187–1196.

[42] F. Becattini, T. Uricchio, L. Seidenari, L. Ballan, and A. D. Bimbo, "Am i done? predicting action progress in videos," *ACM Trans. Multimedia Comput. Commun. Appl.*, 2020.

[43] H. Doughty, D. Damen, and W. Mayol-Cuevas, "Who's better? who's best? pairwise deep ranking for skill determination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[44] E. Biyik, "Learning preferences for interactive autonomy," *CoRR*, vol. abs/2210.10899, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2210.10899

[45] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," in *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, N. M. Amato, S. S. Srinivasa, N. Ayanian, and S. Kuindersma, Eds., 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p53.html

[46] Y. Liu, G. Datta, E. R. Novoseller, and D. S. Brown, "Efficient preference-based reinforcement learning using learned dynamics models," *CoRR*, vol. abs/2301.04741, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2301.04741

[47] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[48] K. Hartikainen, X. Geng, T. Haarnoja, and S. Levine, "Dynamical distance learning for semi-supervised and unsupervised skill discovery," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=H1lmhaVtvr

[49] D. Ghosh, A. Gupta, A. Reddy, J. Fu, C. M. Devin, B. Eysenbach, and S. Levine, "Learning to reach goals via iterated supervised learning," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=rALA0Xo6yNJ

[50] M. Liu, M. Zhu, and W. Zhang, "Goal-conditioned reinforcement learning: Problems and solutions," *CoRR*, vol. abs/2201.08299, 2022. [Online]. Available: https://arxiv.org/abs/2201.08299

[51] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[52] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 1856–1865. [Online]. Available: http://proceedings.mlr.press/v80/haarnoja18b.html

[53] D. Yarats, I. Kostrikov, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=GY6-6sTvGaf

[54] L. Smith, I. Kostrikov, and S. Levine, "A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning," *arXiv preprint arXiv:2208.07860*, 2022.

[55] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[56] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 2052–2062.

[57] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *CoRR*, vol. abs/2005.01643, 2020. [Online]. Available: https://arxiv.org/abs/2005.01643

[58] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[59] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, "Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems*, vol. 29, 2016.

[60] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[61] B. Eysenbach, S. Chaudhari, S. Asawa, S. Levine, and R. Salakhutdinov, "Off-dynamics reinforcement learning: Training for transfer with domain classifiers," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[62] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Mastering visual continuous control: Improved data-augmented reinforcement learning," in *International Conference on Learning Representations*, 2022.

[63] T. Hiraoka, T. Imagawa, T. Hashimoto, T. Onishi, and Y. Tsuruoka, "Dropout q-functions for doubly efficient reinforcement learning," in *International Conference on Learning Representations*, 2022.

[64] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *CoRL*, 2022.

[65] S. Haldar, V. Mathur, D. Yarats, and L. Pinto, "Watch and match: Supercharging imitation with regularized optimal transport," *CoRL*, 2022.

[66] K. Zakka, A. Zeng, P. Florence, J. Tompson, J. Bohg, and D. Dwibedi, "XIRL: cross-embodiment inverse reinforcement learning," *CoRL*, 2021.

[67] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration," in *IEEE International Conference on Computer Vision Workshops*, 2021.