Video Imitation GAN: Learning control policies by imitating raw videos using generative adversarial reward estimation

Subhajit Chaudhury, Daiki Kimura, Tu-Hoa Pham, Asim Munawar and Ryuki Tachibana
IBM Research AI - Tokyo
{subhajit, daiki, pham, asim, ryuki}@jp.ibm.com

Abstract

Natural imitation in humans usually consists of mimicking visual demonstrations of another person by continuously refining our skills until our performance is visually akin to the expert demonstrations. In this paper, we are interested in imitation learning of artificial agents in the natural setting - acquiring motor skills by watching raw video demonstrations. Traditional methods for learning from videos rely on extracting meaningful low-dimensional features from the videos followed by a separate hand-crafted reward estimation step based on feature separation between the agent and expert. We propose an imitation learning framework from raw video demonstrations, that reduces the dependence on hand engineered reward functions, by jointly learning the feature extraction and separation estimation steps, using generative adversarial networks. Additionally, we establish the equivalence between adversarial imitation from image manifolds and low-level state distribution matching, under certain conditions. Experimental results show that our proposed imitation learning method from raw videos produces a similar performance to state-of-the-art imitation learning techniques with low-level state and action information available while outperforming existing video imitation methods. Furthermore, we show that our method can learn action policies by imitating video demonstrations available on YouTube with performance comparable to learned agents from true reward signal. Please see the video at https://youtu.be/bvNpV2Q4rOA.

Introduction

Reinforcement learning methods (Sutton and Barto, 1998; Schulman et al., 2015a; Mnih et al., 2015) learn action policies in agents to achieve a given task by maximizing a guiding reward signal. However, in most cases, human skills are picked up by imitating other experts. For example, we learn tasks like writing, dancing, swimming etc. by first watching others perform them, after which we try to imitate those actions. Vision being the most informative of the five input modalities, visual imitation is the most common kind of imitation in humans. Learning a task by visual imitation involves extracting a high-level representation of the demonstrations performed by the expert, followed by matching it with own performance. While humans can perform visual imitation with relative ease, the same is not true for artificial agents, mainly because of the difficulty in robust high-level visual representation extraction and feature matching with expert’s visual demonstrations.

In cases where a well-defined reward is not available, prior methods have focused on learning from expert demonstration (Schaal, 1997) or imitation learning from at expert (Pomerleau, 1991; Ng and Russell, 2000). In most conventional imitation learning methods, a set of expert trajectories consisting of both state and actions information is available but we do not have access to the reward (or cost function), used to achieve the expert behavior. The goal is to learn a new policy, which imitates the expert behavior by maximizing the likelihood of the given demonstrations. However, in the natural imitation setting, exact action information is not always available and the expert is emulated by mimicking only visual demonstrations.

Attempts have also been made towards learning from observations in the absence of action information. These methods include imitating motion capture data (Merel et al., 2017; Peng et al., 2018) or video demonstrations (Serban et al., 2017; Liu et al., 2017). Existing methods for learning from videos, focus on extracting a high-level feature embedding of the image frames ensuring that pair of co-located frames lie close in the embedding space compared to the ones having higher temporal distance. This is then followed by a reward estimation step based on some hand-crafted distance metric between the feature embeddings of the agent and expert images. These methods make a restrictive assumption that the agent’s trajectories need to be time

Figure 1: Proposed imitation learning from expert videos
synchronized to the expert for reward estimation. Furthermore, different kind of environments might require careful parameter tuning for appropriate reward shaping.

In our proposed method, we address both of the above issues. Firstly, we jointly learn the high-level feature embeddings and agent-expert distance metric removing the need for hand-crafted reward shaping. Specifically, we use a generative adversarial network (GAN, Goodfellow et al. (2014)) with the policy network as the generator along with a discriminator, which performs binary classification between the agent and the expert trajectories. The reward signal is extracted from the discriminator output indicating the discrepancy between the current performance of the agent with the expert. The overview of our method is shown in Figure 1. Since the binary classifier is trained by randomly sampling the agent and expert trajectories, there is no need for time synchronization as well. We establish a connection between generative adversarial reward estimation in raw image space and the agent’s state distribution matching to the expert. We show that if there exists a one-to-one mapping between state information (joint angles, velocities etc.) and corresponding visual observations, then learning a classification boundary on the raw image space is equivalent to learning a binary classifier in the agent’s state space. In our experiments, we show that our proposed method can successfully imitate raw video demonstrations producing a similar performance to state-of-the-art imitation learning techniques which uses state-action information in expert trajectories. Furthermore, we empirically show that the policies learned by our method out-performs other video imitation methods even in the presence of noisy expert demonstrations. Lastly, we demonstrate that the proposed method can successfully learn policies that imitate video demonstrations available on YouTube.

Related Works

Imitation learning methods can be broadly divided into two main categories: Behavior Cloning (BC) and Inverse Reinforcement Learning (IRL) as discussed below.

Behavior Cloning: In behavior cloning (Pomerleau (1991); Duan et al. (2017)), the policy \( \pi(\alpha|s) \) learns the optimal action given current state in a supervised manner considering each state-action tuple as independent and identically distributed (i.i.d). While it is more sample efficient compared to model-free reinforcement learning, BC suffers from the problem of compounding errors. Since, it maximizes the likelihood of state-action tuples only at the current time-step, it ignores the effect of current action on future state distribution. Thus, the errors in each step are compounded leading to a policy that is susceptible to deviation from the optimal trajectory (Ross, Gordon, and Bagnell (2011); Ross and Bagnell (2010)).

Inverse Reinforcement Learning: To alleviate the drawbacks of behavior cloning, IRL casts the imitation learning problem in a Markov Decision Process (MDP) setting (Ng and Russell (2000)), where the goal is to estimate the reward signal that best explains the expert trajectories (Abbeel and Ng (2004); Syed and Schapire (2008); Ziebart et al. (2008)). A reinforcement learning loop is executed using the estimated reward signal for policy optimization. Although IRL reduces the effect of compounding errors evident in behavior cloning (Ziebart et al. (2008)) showed that estimating a unique reward function from expert state-action tuples is an ill-posed problem. Recently, Generative Adversarial Imitation Learning (GAIL, Ho and Ermon (2016)) presented the connection between maximum entropy IRL and GANs, which matches the state-action distribution of the agent and expert for imitation learning.

Learning from Observations: The techniques discussed above rely on both state-action information in the expert trajectories for imitation. However, in many practical cases, the expert action information might not be available. Deep Mimir (Peng et al. (2018)) suggested to learn a simple reward-function as the exponential of Euclidean distance between the time-aligned expert and agent trajectories. Although they showed that it can imitate impressive motion capture data, it requires careful reward shaping and assumes that the agent’s state can be reset to arbitrary key-points from the expert demonstrations. Generative adversarial methods, that match the state distribution of the agent with that of the expert (Merel et al. (2017); Henderson et al. (2017); Torabi, Warnell, and Stone (2018a); Torabi, Warnell, and Stone (2018b)) have also been proposed in literature with similar performance to GAIL. Behavior Cloning from Observations (BCO, Torabi, Warnell, and Stone (2018a)) suggest estimating the expert action information from the provided expert observations, by learning an inverse dynamics model. The imitation policy is then learned by simple behavior cloning using the estimated expert actions in the previous step.

Imitation from Videos: The first group of methods for video-based imitation in artificial agents (Sermanet et al. (2017); Aytar et al. (2018); Liu et al. (2017); Kimura et al. (2018)), suggest learning a low-dimensional semantic representation of the video frames. Time-Contrastive Networks (TCN, Sermanet et al. (2017)) propose a self-supervised representation learning method using a triplet loss that ensures that pair of co-occurring image frames (with a threshold from current frame) are close to each other in the embedding space than any negative frame (outside the threshold). Temporal distance classification (TDC, Aytar et al. (2018)) cast representation learning as a multi-class classification problem with labels corresponding to temporal distance between video frames. Subsequently, the next step involves estimating a reward signal based on the distance between learned representations for time-synchronized expert-agent video trajectories. Finn et al. (2017) propose to first learn a policy to perform other tasks using numerous expert demonstrations consisting of state-action information. The previously gathered experience is leveraged to quickly learn the new skill from very few demonstrations while learning the required novel task.

In contrast to the above methods, firstly, our method can learn from only a few expert video demonstrations in the absence of action information with no pre-training stage. Secondly, hand-crafted distance calculation between learned features for reward estimation may be sub-optimal than automatically learning such distance metric. In addition, another restriction which is imposed by the previous methods is that the expert-agent trajectories need to advance by same
time-steps for reward estimation. In our work, we address the above issues by jointly learning the feature extraction and distance metric between such embeddings. We learn a binary classifier that distinguishes the expert’s image distribution from that of the agent, thus removing the need for time-synchronized matching and reward engineering.

**Background**

**Notations**

We consider a finite horizon Markov Decision Process (MDP), defined as \((S, A, P, r, \gamma, p_0)\), where \(S\) is the finite state space, \(A\) is the finite action space, \(P : S \times A \times S \to [0, 1]\) is the transition probability, \(r : (S \times A) \to \mathbb{R}\) is the reward function that is used to learn the policy during reinforcement learning, \(p_0 : S \to [0, 1]\) is the distribution of the initial state \(s_0\), and \(\gamma \in (0, 1)\) is the discount factor. Throughout the paper, we denote an instance, \(s \in S\) as the agent’s states containing low-level information like joint angles and velocities. For every state instance, there is a corresponding visual observation \(I \in I\) depicting the agent’s state in raw pixels. We consider \(\pi : S \times A \to [0, 1]\) to be a stochastic policy that estimates a conditional probability of actions given the state at any given time-step. The expected discounted reward (value function) is denoted as \(E_{\pi}[r(s, a)] = E [\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]\), where \(s_0 \sim p_0\), \(a_t \sim \pi(s_t)\), and \(s_{t+1} \sim P(\cdot|s_t, a_t)\) for \(t \geq 0\). We assume that some finite number of trajectories \(\tau_E\) are available and that we cannot interact with the expert policy during imitation learning.

**Generative Adversarial Networks**

Given some samples from a data distribution \(p_{data}\), GANs learn a generator \(G(z)\), where \(z \sim p_z\) is sampled from the noise distribution, by optimizing the following cost function,

\[
\min_{G} \max_{D \in \{0,1\}} E_{x \sim p_{data}} \log(D(x)) + E_{z \sim p_z} \log(1 - D(G(z))) \tag{1}
\]

The above cost function defines a two player game, where the discriminator tries to classify the data generated by the generator as label 0 and samples from the true data distribution as 1. The discriminator acts as a critic that measures how much the samples generated by the generator matches the true data distribution. The generator is trained by assigning label 1 to the generated samples from G, with fixed discriminator. Thus the generator is fooled to distinguish into believing that the generated samples are from the true data distribution.

GANs provide the benefit that it automatically learns an appropriate loss between the data and generated distributions. The gradients from the above two-step training methods of the binary classifier are sufficient to produce a good generative model even for high dimensional distributions. Deep Convolutional GANs (Radford, Metz, and Chintala (2015)) have shown impressive results on learning the distribution of natural images. Additionally, we do not need paired training data for learning the data distribution. This property was utilized by CycleGAN (Zhu et al. (2017)) to learn a conditional distribution for transforming natural images from one domain to another using unpaired images.

**Occupancy Measure Matching and Generative Adversarial Imitation Learning**

Unlike the task of learning a generative model for image, where the different image samples can be assumed as i.i.d., the distribution of state-action tuple at each time-step, in MDPs, are conditionally dependent on past values. As such, the visitation frequency of state-action pairs for a given policy is defined as the occupancy measure, \(\rho_\pi(s, a) = \pi(a|s) \sum_{s'=0}^{\infty} \gamma^t p(s = s'|s)\). It was shown by Syed, Bowling, and Schapire (2008) that the imitation learning problem (matching expected long term reward, \(E_\pi[r(s, a)]\)) can be reduced to the occupancy measure matching problem between the agent’s policy and the expert. Employing maximum entropy principle to the above occupancy matching problem, we get a general formulation for imitation learning as,

\[
\min_{\pi \in \Pi} d(\rho_\pi(s, a), \rho_{\pi_E}(s, a)) - \lambda H(\pi) \tag{2}
\]

where \(d_{\pi_E}\) is the occupancy measure for the expert and \(d(., .)\) is the distance function between expert and agent’s occupancy measure. \(H(\pi)\) is the entropy of the policy and \(\lambda\) is the weighting factor. Generative adversarial imitation learning (Ho and Ermon (2016)) proposed to minimize the Jensen-Shannon divergence between the agent’s and expert’s occupancy measure and showed that this can be achieved in an adversarial setting by finding the saddle point \((\pi, D)\) of the following cost function,

\[
\min_{\pi \in \Pi} \max_{D \in \{0,1\}} E_{\pi_E}[\log D(s, a)] + E_\pi[\log(1 - D(s, a))] - \lambda H(\pi) \tag{3}
\]

The discriminator \(D(s, a) = p(E|s, a)\), represents the likelihood that the state-action tuple is generated from the expert rather than by the agent. \(\{E, A\}\) are two classes of the binary classifier representing expert and agent. As this two-player game progresses, the discriminator learns to better classify the expert trajectories from agent and the policy learns to generate trajectories similar to the expert.

**Imitation Learning from Observations using GANs**

Similar to GAIL’s objective for imitation learning from expert-state-action information, it has been shown (Merel et al. (2017), Henderson et al. (2017), Torabi, Warnell, and Stone (2018a)) that it is possible to imitate an expert policy from state demonstrations only, even without action information. We define a reward function \(r(s, s')\) which depends only on state transitions, \((s, s')\). We define the 2-step state transition occupancy measure as, \(\rho_\pi(s, s') = \sum_{a} p(s'|a, s)\pi(a|s)\sum_{s''} \gamma^t p(s = s''|s)\).

Following similar arguments outlined in (Ho and Ermon (2016)) and replacing state-action pair with state-transition \((s, s')\), (Torabi, Warnell, and Stone (2018b)) has formally shown that minimizing the Jensen-Shannon divergence between occupancy measures of the state transitions for the
expert and agent, leads to an optimal policy that imitates the expert. We arrive at cost function similar to Eq[3] with state-action pair replaced by state transitions given in Equation[4] as,

$$\min_{\pi} \max_{D \in (0,1)} \mathbb{E}_{\pi}[\log D(s,s')] + \mathbb{E}_{\pi}[\log(1-D(s,s'))]$$

(4)

where the discriminator $D(s,s') = p(E|s,s')$, tries to discriminate between the state-transition samples of the expert and agent. This serves as reward signal for training the policy $\pi$ using policy gradient method similar to GAIL. We removed the entropy term following [Ho and Ermon 2016].

Proposed Algorithm

Video Imitation Generative Adversarial Network

We assume that only a finite set of expert demonstrations, $\tau_E = \{I_0, I_1, ..., I_T\}$, are available consisting of videos depicting the expert’s policy behavior in raw pixels. Similar to natural imitation learning setting for humans, the goal is to learn a low-level control policy $\pi(a|s)$ that performs actions based on state representations, by maximizing the similarity between the agent’s video output with that of the expert.

We first establish a relationship between states and the corresponding rendered images, to formally deduce our video imitation algorithm. Let us assume that $g: S \rightarrow I$ is the render mapping, that maps low-level states $s$ to the high-level image observations $I$. We make the following proposition,

Proposition: Matching the data distribution of the images generated by the agent with the expert demonstrations by optimizing the cost function in Equation[5] is equivalent to matching the low-level state occupancy measure, given that the mapping $g(.)$ is injective in nature.

$$\min_{\pi} \max_{D \in (0,1)} \mathbb{E}_{\pi}[\log D(I,I')] + \mathbb{E}_{\pi}[\log(1-D(I,I'))]$$

(5)

where the discriminator $D(I,I') = p(E|I,I')$ gives the likelihood that the image trajectory $\{I,I'\}$, is generated by the expert policy $\pi_E$. Rather than the expert policy $\pi$.

Proof: We start by restricting the image space to a manifold constrained on the state space, such that, $I = \{g(s), \forall s \in S\}$. Since, we assume that $S$ is a finite state space (similar to [Ho and Ermon 2016]) and $g$ is an injective mapping, then it follows that $g: S \rightarrow I$ is bijective.

Using Bayes rule, we can decompose $D(I,I')$ as,

$$D(I,I') = p(E|I,I') = \frac{p(I,I'|E)p(E)}{p(I,I')} = \frac{p(I,I'|A)p(A) + p(I,I'|E)p(E)}{p(I,I')}$$

(6)

$$= \frac{1}{1 + \beta(I,I')}$$

where $\beta(I,I') = \frac{p(I,I'|A)}{p(I,I'|E)}$ is the likelihood ratio of the image transitions given current agent’s policy to that of the expert policy. We assume that the equal samples from agent and expert are used during the training, $p(E) = p(A)$.

Since there exists a bijective mapping between the high-level image space $(I)$ and the low-level state space $(s)$, we can write

$$\beta(I,I') = \frac{p(I,I'|A)}{p(I,I'|E)} = \frac{p(s,s'|A)\det J^{-1}}{p(s,s'|E)\det J^{-1}} = \frac{p(s,s'|A)}{p(s,s'|E)}$$

(7)

where $J = \frac{dI}{ds}$ is the Jacobian of bijective function $g(\cdot)$. From Equation[7] it follows that $D(I,I') = D(s,s')$. Therefore, learning a discriminator $D(I,I')$ on the image space, is equivalent to learning a binary classifier on the state space that distinguishes between the expert and agent trajectories. Thus, using an estimated reward function based on the image discriminator for policy optimization would lead to occupancy measure matching in the low-level state space which in turn will learn a policy that imitates the expert, as shown in previous section. This concludes the proof.

We outline the practical algorithm for Video Imitation Generative Adversarial Network (VIGN) as algorithm[1].

Algorithm 1 VIGN($\tau_E$)

Require: $\tau_E$: Expert video demonstrations without action
1: - Randomly initialize the parameters $\theta$ for policy $\pi$ and $\phi$ for the discriminator $D$
2: for $k$ from 0 to $\infty$, until convergence do
3: Execute $\pi_\theta$ and store the state transition $\{s,a,s'\} \rightarrow \tau_s$ for $T$ time-steps.
4: Render corresponding raw images $I = g(s)$ and store generated video $(I_1, I_2, ..., I_T)$ in image buffer, $\tau_I$
5: Perform a gradient step for discriminator parameters from $\phi_k$ to $\phi_{k+1}$ using loss as,
6: Estimate reward from the discriminator as,
7: Update policy parameters from $\theta_k$ to $\theta_{k+1}$, using TRPO update subject to KL constraint
8: end for
9: Return $\pi_\theta$

Similar to GANs, the algorithm learns a discriminator that assigns label 0 to the agent generated video frames and label 1 to the expert’s frames. The discriminator is used to provide an estimate of the reward function, $r(I,I')$ that is used to learn a control policy from low-level states using TRPO [Schulman et al. 2015b]. Variance reduction in policy gradients is performed following [Schulman et al. 2015b].

1Since, we constrain the image space to a manifold governed by the state space, the rank of image manifold will be same as the dimensionality of state space. Thus, we can find maximally independent set of rows for $J$ to construct a full rank Jacobian matrix.
Figure 2: (a) Generative adversarial state imitation method learns a robust reward function compared to hand-engineered reward, producing better performance than TRPO agent (b) Comparing SIGAN to various methods. BC and GAIL use expert state-action tuples, while DeepMimic and SIGAN use only state information. SIGAN performs similarly as GAIL while being superior in performance to BC and DeepMimic.

Intuitively, unlike the traditional method for generating images from i.i.d. random noise using GANs, since the generated image distribution, in our case, lies on a manifold constrained by the agent’s state distribution, matching the agent-expert image distribution ensure state distributions matching.

Our GAN-based reward estimation automatically learns both feature representation from images and the distance between such embeddings in a joint fashion from the data-distribution of expert and the agent. This reduces the need for hand-crafted reward shaping. Secondly, since we train the discriminator by comparing random samples drawn from both agent and expert distributions, there is no need for them to be time-synchronized. Thus, the proposed method addresses both the issues of hand-crafted reward estimation and time-synchronized agent-expert matching that are faced by previous works.

Validity of the injective assumption: Our assumption for injective state to image matching, is reasonable for most environments where the agent resides in a two dimensional world. For agents residing in a 3D world there can be some degenerate cases where this assumption is not valid. Since images capture only a 2D projection of the 3D world, such cases may include moving along the axis of the camera with increasing size (such that the projection on the camera does not change) or occlusions. At those degenerate points, the Jacobian $J$ will be singular and Equation 7 will not be valid. However, we perform empirical evaluation on both 2D and 3D environments showing that if the expert trajectory does not contain such rare degenerate cases, our proposed method can recover a favorable policy.

Experiments

We perform initial experiments for imitation learning from low-level expert state trajectories, which we refer to as State Imitation Generative Adversarial Networks (SIGAN) in our experiments. Subsequently, we evaluate the proposed VI-GAN algorithm from raw videos as expert demonstrations. Qualitative comparison are shown in the attached supplementary video submission.

A neural network policy, consisting of 2 fully connected layers of 64 ReLU activated units each, was used for all experiments. For the discriminator, we used 2 fully connected layers of 128 ReLU activated units for imitation from states and a convolution neural network with similar discriminator architecture as in [Radford, Metz, and Chintala (2015)] for video imitation. TRPO was used for policy optimization for all experiment including expert policies. Other parameters were set similar to [Ho and Ermon (2016)]. We evaluate our algorithms on a toy reaching example in addition to 5 physics-based environments: two classical control task of CartPole and Pendulum, along with 3 continuous control tasks of Hopper, Walker2d and HalfCheetah simulated in MuJoCo [Todorov, Erez, and Tassa (2012)].

Toy Example: GridWorldMaze

In this task, the agent’s goal is to navigate through a maze and reach the target without colliding into the walls. The reward is the negative distance of the agent from target and -200 for colliding with walls. We gathered 500 demonstrations, where an expert human traced the trajectories from start position to the target, via mouse movements.

Policy trained via SIGAN was compared with TRPO agent trained using the above reward signal, as shown in Figure 2a. Due to the complex arrangement of obstacles in the environment, the hand-engineered reward signal did not learn a good policy whereas the adversarial reward estimation learns a robust reward function which is indicated by the superior final policy performance.

Imitation from Low-Level States

We consider imitation learning from expert trajectories consisting of low-level state information, in the absence of action information. The reward $r_t$ at time-step $t$ is estimated by the likelihood of $n$-step observation concatenation,
\( r_t = -\log(1 - D(s_t, s_{t+1}, \ldots, s_{t+n})) \). The motivation behind concatenating \( n \)-step states, is to enable our reward estimator to better approximate the original reward signal \( r(s, a) \), which the expert was trained with. We compared the performance of SIGAN for 3 different values of \( n \) with behavior cloning, GAIL and DeepMimic (which uses heuristic rewards) as shown in Figure 2b. Results show that adversarial state imitation can recover a policy that performs at-par with GAIL and out-performs other methods, similar to the results reported in prior works.

**Imitation from Raw Videos**

We evaluate our proposed VIGAN method for imitating raw video demonstrations captured from expert policy. Each trajectory of the video demonstrations consisted of 800 image frames (32 secs at 25 fps) for Hopper and Walker2d environments. CartPole and Pendulum used 200 frames per trajectory. Both the rendered videos from expert and the agent were resized to \( 64 \times 64 \times 3 \) color images for training the discriminator.

We compare our proposed method to the following video imitation methods in addition to GAIL.

**DeepMimic + PixelLoss** : In this method, the reward was simply computed as Euclidean distance between normalized images (in the range \([-1, 1]\)) rendered from the agent and expert. We found that taking exponential of the distance, provides a more stable performance. Thus the reward at time \( t \) is computed as, \( r_t = \exp(-2(||I^e_t - I^r_t||^2_2)) \), where \( I^e_t \) and \( I^r_t \) are normalized images sampled from the expert demonstrations and agent policy, respectively.

**DeepMimic + Single View TCN** : We used Single View TCN for self-supervised representation learning using implementation of triplet loss provided by the author. The triplet loss encourages embeddings for co-located images to lie close to each other while separating embeddings for images that are not semantically related. The reward at time \( t \) was computed as \( r_t = \exp(-2||x^e_t - x^r_t||^2_2) \), where \( x^e_t \) and \( x^r_t \) are the agent’s and expert’s render images.

Quantitative evaluation for all the methods is given in Table 1. Learning from raw pixel information did not provide good performance for complex environments because it does not capture the high-level semantic information about the agent’s state. Single View TCN + DeepMimic performed well in some cases (Walker2d, Pendulum) but did not consistently produce a good performance for all cases. We believe that using reward shaping with careful parameter tuning might lead to improved performance for TCN. Our method consistently performed better than the other video imitation methods and was comparable to GAIL’s performance which was trained on low-level state and action trajectories. Furthermore, our reward estimator does not need much parameter tuning (only number of discriminator steps per iteration was varied) across environments, because VIGAN automatically finds the separation between agent’s current video trajectories to that of the expert, ensuring robust estimation of the reward signal from raw pixels.

**Imitation from Noisy Videos**

In this experiment, we added noise to the input video demonstrations to evaluate the robustness of the proposed algorithm to small viewpoint changes. Noise was added in the form of shaking camera, which was simulated by randomly cropping each video frame by 0 to 5% from all four sides. Such noisy video demonstrations might break the injective nature of render mapping because the same low-level state might be mapped to different image observations.

To our surprise, we found that imitation from noisy demonstrations performed similarly to the non-noisy case and GAIL as shown in Figures 3a and 3b. For the Walker2d environment, we found that learning the discriminator with two image transition did not produce good performance. Therefore we used 3 consecutive image transitions, \( D(I_i, I_{i+1}, I_{i+2}) \), for training the discriminator which produced better results in comparison. Qualitative comparison of the learned policies is shown in the supplementary video.

**Imitating YouTube Videos**

Finally, we used our proposed method to imitate video demonstrations available on YouTube which is the closest to
natural imitation learning in humans. We chose the Bipedal-Walker environment from OpenAI gym because we found two videos for this environment with different walking styles trained by others. Description of the videos is given below.

**Video 1**: This video shows learning stages of the agent using Evolutionary Algorithms. We used a 10 second clip from 0:46 to 0:55 corresponding to the behavior of best agent (generation 512). The agent in this video cannot reach the end of environment.

**Video 2**: In this 85 second video, the agent is seen walking in an unusual style. It shows 4 trajectories in total and the agent can reach the end of walking course in this case.

Since the raw expert video demonstrations from YouTube contained additional artifacts like text and window borders, we cropped the videos to keep only the area around the agent’s location and resized each frame to $64 \times 64 \times 3$ images. Same image cropping transformation was applied to the images rendered from the learning agent’s policy.

We compare the learning curve of our algorithm (for video 2) with an expert agent trained via TRPO using the true reward provided by OpenAI gym (Brockman et al. 2016), as shown in Figure 3c. Quantitative evaluation shows that our method successfully learned a policy, by imitating just a short duration of YouTube videos, that performs at par with the expert policy learned using the true reward signal. It is to be noted that since the expert demonstrations, in this case, was directly taken from YouTube, we had no knowledge of the framerate at which the video was recorded. For previous methods, that use time-synchronized reward estimation from videos, additional hyperparameters might be required to match the framerate of the expert demonstrations with the agent. However, our method does not require any such time alignment step. Qualitative comparison of the trained policies (attached supplementary video), shows that the learned policies mimic the original YouTube videos to a large degree of similarity.

**Conclusion**

We proposed an imitation learning method for recovering control policies from a limited number of raw video demonstrations using generative adversarial video matching for reward estimation. We showed that if there exists a one-to-one mapping between the low-level states and image frames, adversarial imitation from videos and agent-expert state trajectories matching, are equivalent problems. Our proposed method consistently out-performed other video imitation methods and recovered a good policy even in the presence of noise. We further demonstrate that our video imitation method can learn policies imitating youtube videos trained by others. In the future, we would like to extend our method to imitate complex video demonstrations with changing background contexts, for example, Montezuma’s Revenge or Torcs driving simulator, where hierarchical adversarial reward estimation based on semantic video clustering, would be required.

**Acknowledgment**

We are thankful to Akshay L Aradhya and Anton Pechenko for allowing us to use their YouTube videos for this research project. We would like to thank Jayakorn Vongkulbhisal and Hiroshi Kajino for helpful technical discussions.

**References**

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In Proceedings of the twenty-first international conference on Machine learning. 1. ACM.

Aytar, Y.; Pfaff, T.; Budden, D.; Paine, T. L.; Wang, Z.; and de Freitas, N. 2018. Playing hard exploration games by watching youtube. arXiv preprint arXiv:1805.11592.
Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. arXiv preprint arXiv:1606.01540.

Duan, Y.; Andrychowicz, M.; Stadie, B.; Jonathan Ho, O.; Schneider, J.; Sutskever, I.; Abbeel, P.; and Zaremba, W. 2017. One-shot imitation learning. In NIPS 30.

Finn, C.; Yu, T.; Zhang, T.; Abbeel, P.; and Levine, S. 2017. One-shot visual imitation learning via meta-learning. In Conference on Robot Learning, 357–368.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In Advances in neural information processing systems, 2672–2680.

Henderson, P.; Chang, W.-D.; Bacon, P.-L.; Meger, D.; Pineau, J.; and Precup, D. 2017. Optiongan: Learning joint reward-policy options using generative adversarial inverse reinforcement learning. arXiv preprint arXiv:1709.06683.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In Advances in Neural Information Processing Systems, 4565–4573.

Kimura, D.; Chaudhury, S.; Tachibana, R.; and Dsngupta, S. 2018. Internal model from observations for reward shaping. ICML Workshop on Adaptive Learning Agents.

Liu, Y.; Gupta, A.; Abbeel, P.; and Levine, S. 2017. Imitation from observation: Learning to imitate behaviors from raw video via context translation. arXiv preprint arXiv:1707.03374.

Merel, J.; Tassa, Y.; Srinivasan, S.; Lemmon, J.; Wang, Z.; Wayne, G.; and Heess, N. 2017. Learning human behaviors from motion capture by adversarial imitation. arXiv preprint arXiv:1707.02201.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. Nature 518(7540):529–533.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In Proceedings of the Seventeenth International Conference on Machine Learning, ICML ’00, 663–670. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Peng, X. B.; Abbeel, P.; Levine, S.; and van de Panne, M. 2018. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Trans. Graph. 37(4):143:1–143:14.

Pomerleau, D. A. 1991. Efficient training of artificial neural networks for autonomous navigation. Neural Computation 3(1):88–97.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Ross, S., and Bagnell, D. 2010. Efficient reductions for imitation learning. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, 661–668.

Ross, S.; Gordon, G. J.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In International Conference on Artificial Intelligence and Statistics, 627–635.

Schaal, S. 1997. Learning from demonstration. In Advances in neural information processing systems, 1040–1046.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015a. Trust region policy optimization. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 1889–1897.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2015b. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438.

Sermanet, P.; Lynch, C.; Chebotar, Y.; Hsu, J.; Jang, E.; Schaal, S.; and Levine, S. 2017. Time-contrastive networks: Self-supervised learning from video. arXiv preprint arXiv:1704.06888.

Sutton, R. S., and Barto, A. G. 1998. Reinforcement learning: An introduction, volume 1. MIT press Cambridge.

Syed, U., and Schapire, R. E. 2008. A game-theoretic approach to apprenticeship learning. In Advances in neural information processing systems, 1449–1456.

Syed, U.; Bowling, M.; and Schapire, R. E. 2008. Apprenticeship learning using linear programming. In Proceedings of the 25th international conference on Machine learning, 1032–1039. ACM.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, 5026–5033. IEEE.

Torabi, F.; Warnell, G.; and Stone, P. 2018a. Behavioral cloning from observation. arXiv preprint arXiv:1805.01954.

Torabi, F.; Warnell, G.; and Stone, P. 2018b. Generative adversarial imitation from observation. arXiv preprint arXiv:1807.06158.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In IEEE International Conference on Computer Vision.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In AAAI, volume 8, 1433–1438. Chicago, IL, USA.