EXTRANEOUSNESS-AWARE IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual imitation learning is an effective approach for intelligent agents to obtain control policies from visual demonstration sequences. However, standard visual imitation learning assumes expert demonstration that only contains the task-relevant frames. While previous works propose to learn from *noisy* demonstration, it still remains challenging when there are locally consistent yet task irrelevant subsequences in the demonstration. We term this kind of imitation learning "imitation-learning-with-extraneousness" and introduce Extraneousness-Aware Imitation Learning (EIL), a self-supervised approach that learns visuomotor policies from third-person demonstrations where extraneous subsequences exist. EIL learns action-conditioned self-supervised frame embeddings and aligns task-relevant frames across videos while excluding the extraneous parts. Our method allows agents to learn from extraneousness-rich demonstrations by intelligently ignoring irrelevant components. Experimental results show that EIL significantly outperforms strong baselines and approaches the level of training from the perfect demonstration on various simulated continuous control tasks and a "learning-from-slides" task. The project page can be found here: `https://sites.google.com/view/iclr2022eil/home`.

1 INTRODUCTION

Imitation learning (IL) enables intelligent agents to acquire various skills from demonstrations (Argall et al., 2009; Schaal et al., 1997). Recent advances also extend IL to the visual domain (Pathak et al., 2018; Zhang et al., 2018; Young et al., 2020; Zhu et al., 2018). However, in contrast to how humans learn from demonstrations, artificial agents usually require "clean" demonstrations that contain few extraneous components. For example, when learning to cut a potato from videos, a human can naturally ignore the demonstrators' extraneous action of turning off a stove in halfway; but agents will consider the irrelevant action during policy optimization and hence fail to learn with such demonstration data. Such difference effectively prevents agents from leveraging massive unstructured visual data which humans can utilize to learn new skills.

Popular methods of imitation learning include behavioral cloning (Bain & Sammut, 1995; Ross et al., 2011; Pomerleau, 1991) and inverse reinforcement learning (Ng et al., 2000). Both approaches achieve good performance when expert trajectories of observation-action pairs are available. However, when presented with real-world data, vanilla imitation methods often fail due to the insufficient quality of demonstrations. To solve this problem, some recent literature (Tangkaratt et al., 2020a; Brown et al., 2019; Wu et al., 2019; Tangkaratt et al., 2020b) propose methods to perform imitation learning from noisy demonstrations. However, these methods are limited by their requirement of additional labels or a well-trained critic function for data filtration. Some methods also make additional assumptions about noise to leverage domain knowledge. For example, they may assume gaussian distribution for the action noise and/or the local smoothness of state or action throughout the demonstration trajectory. However, many of the real-life data we are interested in usually contain extraneous subsequences, where the agent may perform locally smooth and consistent but task-irrelevant actions. Moreover, many of these proposed methods cannot be applied to the scope of visual imitation learning, where the observations are high-dimensional images instead of states. How can we leverage unannotated visual demonstrations for imitation learning without being negatively impacted by their extraneous sub-sequences?

In this paper, we propose an Extraneousness-Aware Imitation Learning (EIL) that enables agents to smartly imitate from noisy videos demonstrations where extraneous segments are present. Our

method allows agents to identify extraneous subsequences via self-supervised learning and then selectively imitate from task-relevant parts of the videos. Specifically, we train an action-conditioned encoder through temporal cycle-consistency (TCC) (Dwibedi et al., 2019) to obtain embeddings of each frame. Then, we propose an Unsupervised Voting-based Alignment algorithm (UVA) to align the important frames across video clips so that the agent can ignore task-irrelevant frames. In our work, we also introduce a few new tasks to benchmark imitation learning from ubiquitous imperfect data with extraneous actions.

We evaluate our method on both simulated discrete action tasks and continuous control tasks. The experiments demonstrate the ability of the proposed encoder to produce embedding that is useful to both downstream tasks and extraneousness detection. We also demonstrate that our method can automatically filter data in an unsupervised manner and enable agents to learn better policies without additional assumptions.

Our contributions can be summarized as follows: 1) We introduce Extraneousness-Aware Imitation Learning (EIL) that learns selectively from the task-relevant frames by leveraging action-conditioned self-supervised embeddings. 2) We propose visual imitation learning from demonstrations with extraneous segments as meaningful tasks that can be easily transferred to multi-task learning. 3) We show the proposed method can learn to smartly ignore task-irrelevant sub-sequences in demonstration and thus outperform previous visual imitation learning baselines. 4) We propose alignment methods for both cases in which the dataset contains either none or a few perfect demonstrations.

2 Related Works

2.1 Learning from Noisy Demonstration

Traditional imitation learning (Schaal et al., 1997; Argall et al., 2009; Ho & Ermon, 2016) includes behavior cloning (Bain & Sammut, 1995; Pomerleau, 1991) which learns to copy the behaviors from the demonstration and inverse reinforcement learning (Ng et al., 2000) that infers the reward function for learning policies. However, these methods usually assume access to expert demonstrations which are hard to obtain in practice.

Many works try to tackle the imitation learning problem when noisy demonstrations are provided. However, through this line of research (Burchfiel et al., 2016; Wu et al., 2019; Sasaki & Yamashina, 2020; Brown et al., 2019; Tangkaratt et al., 2020b; Kaiser et al., 1995; Grollman & Billard, 2012; Kim et al., 2013), the vast majority of work are done in the low-dimensional state space but not the high-dimensional observations. Moreover, it is common in previous works (Sasaki & Yamashina, 2020) to assume the noise is sampled from an expert policy or a random policy with corresponding probability. Methods designed for such noise might fail completely when the noise are locally consistent and meaningful yet irrelevant to the task while EIL can handle it. Recently, Chen et al. (2021) propose to learn policies from "in-the-wild" videos. while the method achieves impressive results, they focus on dealing with diverse videos without handling the "extraneousness" explicitly.

To learn from imperfect demonstrations, offline Reinforcement Learning (Lange et al., 2012; Fujimoto et al., 2019; Kumar et al., 2020; Hester et al., 2018; Gao et al., 2018) is another powerful paradigm where agents learn from datasets instead of environment interactions. One requirement of offline RL methods is to design reward functions, which usually involve significant human efforts. Our algorithm does not require knowing the true reward function or design a new one as in standard imitation learning methods. Empirically, we find it is hard to train an offline RL algorithm with the learned embeddings as shaped rewards.

2.2 Self-Supervised Learning from Video and Application to Control

Self-supervised learning (SSL) from videos offers a way to learn visual representations with temporal information for different pretext tasks from unlabeled data (Gordon et al., 2020; Pathak et al., 2017; Srivastava et al., 2015; Mathieu et al., 2015; Jayaraman & Grauman, 2015; Agrawal et al., 2015; Goroshin et al., 2015). Besides learning general representation, a more recent line of research utilizes SSL for learning correspondences (Jabri et al., 2020; Wang et al., 2019; Vondrick et al., 2018; Dwibedi et al., 2019; Hadji et al., 2021; Purushwalkam et al., 2020). Specifically, Dwibedi et al. (2019) propose to find correspondences across time in multiple videos with the help of cycle-

consistency (Zhou et al., 2016; Zhu et al., 2017). These methods offer a way to manipulate and leverage real-world data that are usually unlabeled and noisy.

In recent years, SSL also promises to help with visuomotor tasks in control and robotics. For example, Sermanet et al. (2018) learns a self-supervised temporal-consistent embedding for imitation learning and reinforcement learning. XIRL (Zakka et al., 2021) propose to learn a self-supervised embedding that estimates task progress for inverse reinforcement learning. Zhang et al. (2020); Smith et al. (2019); Xiong et al. (2021) directly map the observations such as images to the target domain. The distinction between EIL and previous work is that we tackle the problem when demonstration has noise such as extraneous subsequences rather than different domains such as visual difference, embodiment differences.

3 METHOD

In this section, we describe our problem setting and introduce Extraneousness-Aware Imitation Learning (EIL), a simple yet effective approach for learning visuomotor policies from videos that have extraneous subsequences.

3.1 PROBLEM STATEMENT

In the problem setting, we consider an agent aiming to learn visuomotor policies from K video demonstration sequences $\{\mathcal{D}_i\}_{i=1}^{K}$ where, in the i^{th} video, each frame j contains both observations s_j^i and a_j^i. There are no constraints on the viewpoint, background, and quality. Specifically, for each sequence in the demonstration set, there are S extraneous subsequences $\{\mathcal{E}_m\}_{n=1}^{S}$ that are trying to perform an irrelevant task or random actions. In contrast to existing works that have a strong assumption about the noise, our setting only assumes each video contains more than 50% of task-relevant content. This is a typical assumption when learning from imperfect data (Angluin & Laird, 1988; Natarajan et al., 2013).

The agent operates in a fixed horizon Markov decision process (MDP) $\mathcal{M}$, consisting of the tuple $(\mathcal{S}, \mathcal{A}, p, \mathcal{R}, T)$. $\mathcal{S}$ is the state space (RGB images), $\mathcal{A}$ is the action space, $p(s_{t+1} \| s_t, a_t)$ is the transition dynamics and $\mathcal{R}$ is an unknown reward function. To successfully imitate and accomplish a task, the agents need to reason about what are the relevant data to learn from and rule out the noise. Under such settings, we also consider when there is one reference trajectory T or no reference trajectory at all.

3.2 EXTRANEOUSNESS-AWARE IMITATION LEARNING (EIL)

Overview. EIL is a general framework for imitating from videos with noisy components. It takes in a set of unannotated task demonstration sequences that may contain extraneous subsequences and learns the correct behavior to accomplish the task. First, we explain the intuition behind our approach. Assuming more than one demonstration sequence is given, we can match their temporal embeddings to find the most task-relevant portions of the video. In the case where a perfect reference trajectory (no noise) is available, we can match frames in other sequences to that of the reference trajectory. However, in most cases, it is hard to obtain such a reference trajectory. Hence, an unsupervised alignment algorithm is needed for the matching.

Our approach, EIL, is outlined in figure 1. In figure 1 (a), we learn a temporal representation of each frame conditioned on both its visual observation and action by using temporal cycle consistency loss. After obtaining the representation, as shown in figure 1 (b), we propose an unsupervised voting method to perform video alignment when no reference sequences are available. Finally, as described in figure 1 (c), we perform standard visual imitation learning on top of denoised data derived from the alignment.

Action-conditioned Temporal Cycle Consistency Representation Learning. We first learn representations that encode temporal information for aligning frames across different videos with temporal cycle consistency loss. Specifically, we train an image encoder ψ_I and an action encoder ψ_A that embed the observations and actions into corresponding features $\psi_I(O)$ and $\psi_A(\boldsymbol{a})$. Note that in TCC, the observation O here actually consists of two frames: one "current frame" and one "context

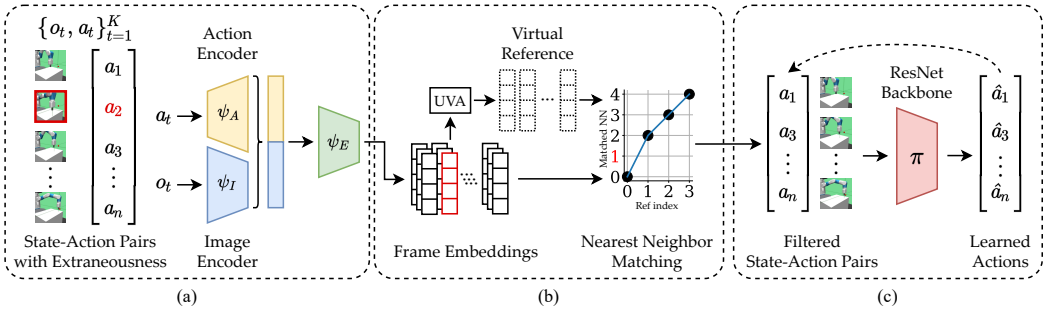


Figure 1: **Extraneousness-Aware Imitation Learning.** Overview figure of EIL. The overall framework contains 3 components: (a) encodes the state action pairs into representation through cycle-consistency loss. (b) takes in the embeddings and process them with unsupervised voting-based alignment (UVA) algorithms. (c) performs visual imitation learning for the aligned state action pairs. We note that (b) can be a simple filtering algorithm when reference trajectories are available.

frame" which is a fixed number of frames ahead of the current frame, to get the temporal information from images, but the action A is only the "current action". Then, we concatentate $\psi_I(O)$ and $\psi_A(\boldsymbol{a})$ to a multi-layer perceptron (MLP) ψ_E to obtain embedding that have temporal correspondence between two sequences. For simplicity, we use two demonstration sequences S and T and their computed embeddings $U = \{u_1, u_2, \cdots, u_N\}$ and $V = \{v_1, v_2, \cdots, v_M\}$ as an example. N and M denotes the sequence lengths respectively.

The main goal here is to encourage cycle-consistency between the two embedding sequences. For any $u_i \in U$, we find the nearest neighbor, $v_j = \arg\min_{v \in V} ||v - u_i||$. Then we repeat the procedure and find $u_k = \arg\min_{u \in U} ||v_j - u||$ which is the nearest neighbor for v_j. When $i = k$, the embedding u_i is cycle-consistent. To optimize the cycle-consistency, we use a differentiable matching loss described in Dwibedi et al. (2019). For the selected u_i, we instead compute the soft nearest neighbor by

$$\tilde{v} = \sum_j^M \alpha_j v_j, \text{where } \alpha_j = \frac{\exp\left(-||u_i - v_j||^2\right)}{\sum_k^M \exp\left(-||u_i - v_k||^2\right)} \tag{1}$$

Then, we compute the "cycle-back" soft nearest neighbor to $\tilde{v}$ similarly:

$$\tilde{u} = \sum_k^N \beta_k u_k, \text{where } \beta_k = \frac{\exp\left(-||\tilde{v} - u_k||^2\right)}{\sum_j^N \exp\left(-||\tilde{v} - u_j||^2\right)} \tag{2}$$

The predicted index $\hat{i}$ can be calculated by $\hat{i} = \sum_k^M \beta_k k$. Since we know the true index i in our selection time, we can minimize the ℓ_2 loss $L = (i - \hat{i})^2$.

We note that under the EIL framework, it is possible to substitute the current embedding with alternatives.

Unsupervised Voting-based Frame Alignment (UVA). After obtaining the embedding for each observation-action pair, we try to align frames and drop extraneous frames according to a frame-wise similarity in the latent space.

To achieve this objective, we propose a voting-based frame matching algorithm that can remove the extraneous parts from a set of sequences. A conceptual illustration is shown in figure 2. For K demonstration sequences, we initially mark the first frame of each video as "alignment frame". Our algorithm can be described as below:

Election Find nearest neighbors of each "alignment frame" in each of the other $K - 1$ videos, that is $K(K-1)$ frames found in total.

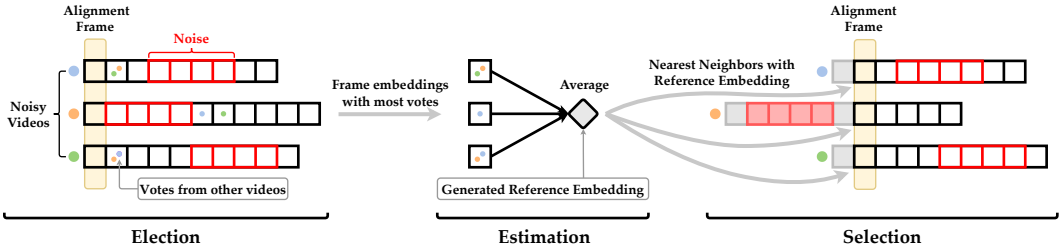


Figure 2: Conceptual illustrations of the unsupervised voting-based alignment (UVA) algorithm.

Estimation In each video, the frame that is nearest neighbor to most other "alignment frame"s is selected as new "alignment frame" of that video. We average the embeddings of all newly selected "alignment frame"s to get a virtual reference embedding.

Selection We select the nearest neighbor frames of the virtual embedding in each video as the new "alignment frame".

We note that the "Estimation" and the "Selection" steps are optionally added to increase the stability of the matching process. Also, "Election" and "Selection" will only choose from frames that are behind the current "alignment frame", which guarantees that the aligned videos are chronologically consistent.

In the simpler setting where we have access to a perfect reference demonstration, our algorithm degenerates to a simple filtering algorithm that uses the reference demonstration for the frame selection. This is done by picking frames in videos that are nearest neighbors to each frame of the reference.

Visual Imitation Learning. As shown in figure 1 (c), we perform standard visual imitation learning to learn a policy π that minimizes distance between the predicted actions and the groundtruth actions using the state-action pairs that are selected previously. Specifically, for continuous actions, we use the ℓ_2 loss:

$$L = ||a_i - \widehat{a}_i||^2 \tag{3}$$

where a_i and $\widehat{a}_i$ are the predicted action and groundtruth action respectively. For discrete actions, we use cross-entropy loss instead. We use ResNet-18 as our policy network to process the image input. As an alternative, we can also substitute the ResNet-18 with the frozen encoder trained during representation learning.

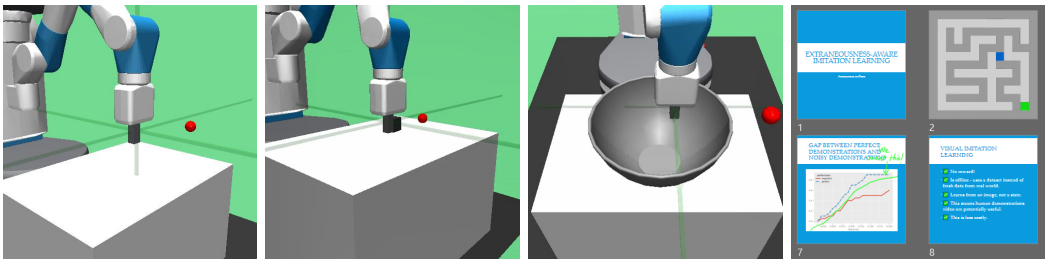
4 Experiment

In this section, we describe the experiment setup and analyze the results. We compare our method with strong baselines on continuous control tasks and a discrete control task, "learning-from-slides". We aim to understand the extraneousness-aware imitation learning problem by answering the following questions: 1) Does EIL as a framework help agent to imitate from visual demonstration that contains extraneousness? 2) Can the action-conditioned self-supervised representation differentiate extraneous components and the task relevant component in the demonstrations? 3) How does EIL perform with different experimental settings and hyperparameters?

4.1 Setup and Dataset

Table 1 summarizes the continuous control tasks we use to evaluate EIL and the type of extraneous subsequences.

We also create a realistic task "Learning-from-Slides". In this task, we mimic how human students learn from course slides or tutorial video demonstrations where the tutor switch between talking and demonstrating. The training dataset is recorded slides that contain both didactic content such as formula or bullet points as well as some visual demonstrations of an interested task, such as an



(a) Reach (b) Push (c) Stir (d) Learning-from-Slides

Figure 3: The continuous control environments (a)(b)(c) and the "learning-from-slides" environment (d). Reach(a) and Push(b) are goal conditioned environments where success is declared when target object is close enough to destination. In *Stir*(c), success is declared if the end-effector trajectory is similar to a target trajectory. The discrete control environment "learning-from-slides" (d) contains slides that demonstrates how to escape a maze but also contains completely irrelevant slides.

Table 1: Different tasks used to evaluate EIL. Each task also has its specific extraneousness actions in the *imperfect demo* dataset, designed to reflect deviations in real-world scenarios.

Task	**Task description**	**Extraneous action in *imperfect demo* dataset**
Reach	The agent needs to move to a goal position.	The agent deviates away from its original trajectory at a random timestep.
Push	The agent needs to push an object to a goal position on the table.	The agent stops pushing and deviates away from the current task at a random timestep.
Stir	The agent needs to place its gripper inside a bowl, then rotate it around the center of the bowl.	The agent moves the gripper outside the bowl for a goal position. This simulates a real-world scenario where a human fetches an object in the middle of stirring.

agent navigating in a maze. The goal for the agent is to learn from those visual demonstrations and accomplish the interested task. The demonstrating frames are paired with actions of the demonstrator agent. We assign a random action for all the non-demonstrating frames. The task visualization can be found in figure 3.

For each task, we collect two datasets – an extraneous dataset as well as a perfect dataset. In the perfect dataset, all the state-action pairs $\{s_i, a_i\}_{i=1\cdots N}$ are sampled from an expert policy π_E. In the extraneous dataset, there are one or more subsequences contain $\{s_i, a_i\}_{i=m\cdots n}$ that contains $n - m + 1$ consecutive steps sampled from another policy π_{any}. We note that we do not have any constraint on the non-expert policy π_{any} which means it can perform meaningful actions for an irrelevant task or totally random actions. We will detail about data collection process in Appendix A.2. In the test time, we either provide one perfect reference trajectory or no reference trajectory.

Metrics and evaluation. To evaluate all the methods, we introduce the metrics. For all goal-conditioned continuous control tasks (reach, push, pick and place), we use both success rate as well as minimum goal-object or gripper-object distance within time horizon as evaluation metrics. All experiments are conducted on 5 random seeds. We average over 20 trials for each seed and report the mean and standard deviation. For goal conditioned tasks like *reach* and *push*, we use a threshold of 0.1 from goal position as the success criteria. In non-goal conditioned task such as *stir*, agent succeeds when it tracks a designated trajectory close enough.

4.2 BASELINES

To verify the effectiveness of EIL, we implement several baselines:

Behavior Cloning We optimize a neural network policy function by minimizing the ℓ_2 or cross-entropy loss for each state-action pair.

Reinforcement Learning with Embedding-based Reward Shaping We train reinforcement learning agents to accomplish the task. Instead of the sparse reward of whether the goal is reached, we use a dense reward. We first obtain a "goal embedding" by averaging over the emdeddings of each video's last frame. The reward is then set to be the ℓ_2 distance between the current state-action embedding and the goal embedding. We note that this baseline has the advantage of interacting with the environment because pure offline RL algorithms are not able to work in this setting.

Time-Contrastive Networks (TCN) (Sermanet et al., 2018) We perform imitation learning with the representation learned by Time-Contrastive Networks. We note that TCN is originally designed for imitation learning and reinforcement learning on synchronized multi-view video demonstrations. In our paper, we adopt this method by mainly using the learned representation in our setup.

4.3 Experimental Results

Table 2: Success rate of different tasks.

Task	Oracle	EIL (Ours)	Behavior Cloning	RL	TCN
Reach	88% $\pm$ 3%	**83%** $\pm$ 3%	76% $\pm$ 4%	0.6% $\pm$ 0.6%	55% $\pm$ 6%
Push	77% $\pm$ 4%	**64%** $\pm$ 5%	54% $\pm$ 5%	22% $\pm$ 3%	-
Stir	100%	**88%** $\pm$ 4%	77% $\pm$ 51%	17% $\pm$ 3%	-
Learning-from-Slides	80% $\pm$ 12%	**75%** $\pm$ 5%	0%	-	-

Table 3: Average and standard deviation of minimum distances for different tasks

Task	Oracle	EIL (Ours)	Behavior Cloning	TCN
Reach	0.0638 $\pm$ 0.0446	**0.0684** $\pm$ 0.0543	0.0874 $\pm$ 0.0722	0.1327 $\pm$ 0.0990
Push	0.0491 $\pm$ 0.0682	**0.0786** $\pm$ 0.0865	0.1023 $\pm$ 0.0797	-
Stir	5.5282 $\pm$ 4.7377	**12.78** $\pm$ 7.46	31.61 $\pm$ 50.13	-

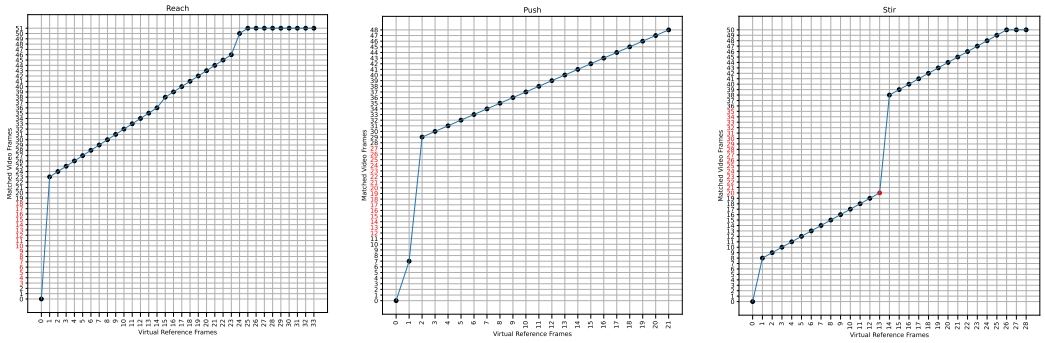
Table 4: Success rate of different tasks without reference demonstration.

Task	Oracle	EIL (Ours) w/o reference	Behavior Cloning	RL	TCN
Reach	88%	**85%**	76%	0.6%	55%
Stir	100%	**88%**	77%	-	-

4.3.1 EIL with one reference demonstration

In this section, we report the performance of different methods when a single perfect demonstration is given. Table 2 summarizes the averaged success rate and standard deviation on all the continuous control tasks. As the table shows, the performance of behavior cloning degrades heavily when extraneousness presents in training data. Reinforcement learning algorithms struggle to accomplish the *reach* task because of the high-dimensional input. On the other hand, the RL baseline achieve 22% success rate on the hard *push* task. This is likely because the learned embedding in the *push* environment is more robust thanks to the additional object. Across all the environments, our method outperforms all the baseline methods and demonstrates its ability to learn from extraneous demonstrations. We also find that EIL outperforms other methods in terms of minimum distance metrics as shown in table 3. We note that for the Stir task, we count the task as successful by a metric that shows how well the gripper is able to adhere to its prescripted trajectory.

Table 2 summarizes the average success rate on the "Learning-from-Slides" task. We find that EIL can successfully accomplish the maze navigation task even when the slides contains extraneous frames. Meanwhile, traditional behavior cloning agents fail to navigate and get stuck in the middle. This experiment shows the potential of EIL to be used in learning behaviors from unlabeled data.



(a) Reach (b) Push (c) Stir

Figure 4: Alignment plots for continuous control tasks. The x-axis is a test-time reference frames. The y-axis is an extraneousness-rich demonstration. We mark the extraneous part with the red color. Each of the black dots represent a matched frame pair. We observe that our proposed method successfully skipped extraneous part while aligning the others.

4.3.2 EIL WITH NO REFERENCE DEMONSTRATION

In this section, we evaluate the performances on the *reach* and *stir* tasks when no perfect demonstration is given. Table 4 summarizes the averaged success rate for EIL and baseline methods on the two tasks. We find that after removing the perfect demonstration trajectory, the performance drops slightly due the lack of guidance. However, the performance of EIL is still comparable to the case where a reference trajectory is provided. Such consistency demonstrates the ability of EIL to solve complex tasks under the situation when reference trajectories are impossible to obtain. The experiment also suggests that it is beneficial to collect at least one good trajectory for marginally better performance when applying EIL.

In figure 4, we visualize the alignment plots for each of the task where the x-axis represents a test-time reference demonstration and the y-axis represents the demonstration with extraneousness (frames colored in red). We find that the proposed algorithm can successfully ignore the extraneous parts in the demonstrations. The border frames near the extraneous part are sometimes categorized as the "good" demonstration. This shows EIL can potentially be further improved by adding end-point-clipping rules if data is abundant. However, we did not add such tricks here. We also find that the performance of the alignment are decreasing while the target task complexity increases. We note that all the models are trained without the reference demonstration represented by x-axis. We only include reference demonstration here for evaluation purpose.

4.4 VISUALIZED RESULTS

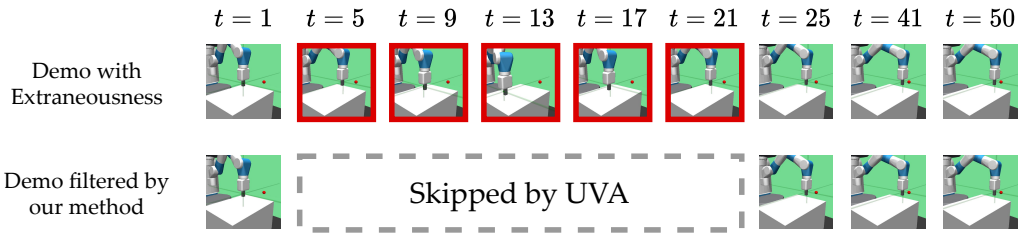


Figure 5: Visualized result of our Unsupervised Voting-based Frame Alignment (UVA) method. The extraneous frames in the video are successfully skipped and improves training results.

In addition, we visualize a perfect demonstration trajectory and the extracted subsequences with EIL in figure 5. From the visualization, we find that when the robot arm deviates from the correct trajectory, it is visually different from the reference demonstrations. More visualized results can be found on our website.

4.5 ABLATION STUDY

Camera settings. In visual imitation learning, the observations are images in the high-dimensional space. The camera extrinsic parameters are therefore important for learning a reasonable policy. In table 5, we show the results with three different camera settings.

The first camera setting is the *default* environment camera angle from *gym* environment. The second setting has the same angle as the former, but the camera position is closer to the table. We call this setting *zoom*. The third camera setting is the third-person view angle that looks at the robot from a higher place. We name this setting *cam3*. We find that our method performs the best when the camera is *zoom*. While our method works for all the camera settings, we find that a good camera pose might further improve the performance.

Table 5: Ablation results on camera settings. Lower is better.

Settings	% Extraneous Frames Matched
default	14.81
cam3	12.12
zoom	**9.68**

Hyperparameters for representation learning. In this section, we ablate the important hyperparameters for learning meaningful embeddings. Although there are many hyperparameters in the training process, we keep most of them fixed.

In table 6, we find the learned embeddings are sensitive to "stride", the hyperparameter that controls the gap between the target frame and the context frames. Hence, with a larger stride usually fit for longer horizon tasks. We find smaller stride gives better performance due to the short horizon of the proposed tasks. For all the experiments, we use the best stride we find in the table without finetuning for specific tasks.

Stride	Minimum Distance
2	**0.0414** $\pm$ 0.0237
3	0.0684 $\pm$ 0.0543
5	0.0722 $\pm$ 0.0505

Table 6: Hyperparameters in representation learning.

Tolerance	Reach	Push	Stir	Slides
2	**0.13**	**0.023**	**0.012**	**0.122**
3	0.15	0.031	0.018	0.136
5	0.21	-	0.021	-

Table 7: Extraneous Frame Fraction for alignment.

Hyperparameters for unsupervised voting-based alignment (UVA). We also ablate important hyperparameters for of the alignment algorithms UVA. In table 7, we find that "tolerance" will influence the resulted virtual embedding length. In practice, we found that UVA sometimes tends to select towards the end of videos too soon. To overcome this shortcoming, we choose from the k^{th} (tolerance) nearest neighbors to virtual reference, instead of one. We choose next alignment frame with smallest frame index among them. We choose 2 as the hyperparameter which best balances performance and chosen frames. We note there are some design choices of how to calculate the virtual embedding, such as using median instead of mean, or using weighted average proportion to the votes each frame get. However, we find these variations don't outperform vanilla mean significantly.

5 CONCLUSION

In this paper, we try to enable an agent to learn from visual demonstrations where temporally consistent yet task-irrelevant subsequences exist. We term this kind of learn problem "imitation-learning-with-extraneousness" and propose Extraneousness-Aware Imitation Learning (EIL), a framework that enables agents to identify extraneous sub-sequences from visual demonstrations via self-supervised learning and learn to accomplish the demonstrated task. To facilitate our study, we also propose continuous control tasks and a "learning from slides" task for benchmarking. We investigate the effectiveness of EIL on the proposed benchmarks and show that EIL outperforms baselines.

6 REPRODUCIBILITY STATEMENT

All the implementation details including hyperparameters, architecture, environment description, data collection as well as evaluation criterion can be found at Appendix A.1. Source code can be downloaded from the project page: `https://sites.google.com/view/iclr2022eil/home`.

REFERENCES

Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pp. 37–45, 2015.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence 15*, pp. 103–129, 1995.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.

Benjamin Burchfiel, Carlo Tomasi, and Ronald Parr. Distance minimization for reward learning from scored trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Annie S Chen, Suraj Nair, and Chelsea Finn. Learning generalizable robotic reward functions from "in-the-wild" human videos. *arXiv preprint arXiv:2103.16817*, 2021.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1801–1810, 2019.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR, 2019.

Yang Gao, Huazhe Xu, Ji Lin, Fisher Yu, Sergey Levine, and Trevor Darrell. Reinforcement learning from imperfect demonstrations. *arXiv preprint arXiv:1802.05313*, 2018.

Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint arXiv:2003.07990*, 2020.

Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Unsupervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pp. 4086–4093, 2015.

Daniel H Grollman and Aude G Billard. Robot learning from failed demonstrations. *International Journal of Social Robotics*, 4(4):331–342, 2012.

Isma Hadji, Konstantinos G Derpanis, and Allan D Jepson. Representation learning via global temporal alignment and cycle-consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11068–11077, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020.

Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1413–1421, 2015.

Michael Kaiser, Holger Friedrich, and Rudiger Dillmann. Obtaining good performance from a bad teacher. In *Programming by Demonstration vs. Learning from Examples Workshop at ML*, volume 95. Citeseer, 1995.

Beomjoon Kim, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup. Learning from limited demonstrations. In *NIPS*, pp. 2859–2867. Citeseer, 2013.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pp. 45–73. Springer, 2012.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26:1196–1204, 2013.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *ICML*, volume 1, pp. 2, 2000.

Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2701–2710, 2017.

Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 2050–2053, 2018.

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

Senthil Purushwalkam, Tian Ye, Saurabh Gupta, and Abhinav Gupta. Aligning videos in space and time. In *European Conference on Computer Vision*, pp. 262–278. Springer, 2020.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Fumihiro Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2020.

Stefan Schaal et al. Learning from demonstration. *Advances in neural information processing systems*, pp. 1040–1046, 1997.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1134–1141. IEEE, 2018.

Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pp. 843–852. PMLR, 2015.

Voot Tangkaratt, Nontawat Charoenphakdee, and Masashi Sugiyama. Robust imitation learning from noisy demonstrations. *arXiv preprint arXiv:2010.10181*, 2020a.

Voot Tangkaratt, Bo Han, Mohammad Emtiyaz Khan, and Masashi Sugiyama. Variational imitation learning with diverse-quality demonstrations. In *International Conference on Machine Learning*, pp. 9407–9417. PMLR, 2020b.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 391–408, 2018.

Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2566–2576, 2019.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pp. 6818–6827. PMLR, 2019.

Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. *arXiv preprint arXiv:2101.07241*, 2021.

Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. *arXiv preprint arXiv:2008.04899*, 2020.

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. *arXiv preprint arXiv:2106.03911*, 2021.

Qiang Zhang, Tete Xiao, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. *arXiv preprint arXiv:2012.09811*, 2020.

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5628–5635. IEEE, 2018.

Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 117–126, 2016.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

Yuke Zhu, Ziyu Wang, Josh Merel, Andrei Rusu, Tom Erez, Serkan Cabi, Saran Tunyasuvunakool, János Kramár, Raia Hadsell, Nando de Freitas, et al. Reinforcement and imitation learning for diverse visuomotor skills. *arXiv preprint arXiv:1802.09564*, 2018.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

Hyperparameters For training the representation encoder, we summarize all the hyperparameters in table 8. For training the imitation learning network, the hyperparameters are summarized in table 9

Table 8: Hyperparameters for Representation Learning

Hyperparameters	**Value**
Batch Size	4
Frame Stride	3
Optimizer	ADAM
Learning Rate	1e-4
Weight Decay	1e-3
Frames per second	30

Table 9: Hyperparameters for imitation learning

Hyperparameters	**Value**
Batch Size	10
Optimizer	ADAM
Learning Rate	3e-4
Weight Decay	0

Data augmentation We use data augmentation during training imitation learning. For each image, we resize it to a 224×224 image.

Architecture We describe the architecture of the state encoder ψ_I and the action encoder ψ_A and the fusion encoder ψ_E. For ψ_I, we have a 4-layer fully-convolutional network to extract image features. The first layer has filter size 7×7 and the rest has filter size 3×3. Then we stack the target frame feature together with context frame features. Then we apply another convolutional layer with filter size 3×3, a 3D max pooling layer and a fully-connected layer to get the embedding. For the action encoder ψ_I, we have a 3 fully connected layers to upsample the action and match the dimension of the embedding ψ_A. After concatenation, the encoder ψ_E is consisted of another MLP with 2 fully-connected layers.

For the imitation learning policy network, we adopt the ResNet-18 (He et al., 2016) architecture. Since we also resolve continuous control tasks, we add another two fully-connected layers after the ResNet-18 for regressing the action.

Reinforcement Learning Baseline In this paragraph, we detail about how we shape the reward for the reinforcement learning baselines. We adopt the existing Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) setting as our RL backbone algorithms. During training, we have two ways to shape the rewards: 1) We compute embeddings for each current frame and the goal frame. The reward is computed as $||embedding - embedding_{goal}||^2$. However, this assumes the embedding can measure the task progress very accurately. 2) when reference trajectory is given, we can guide to RL agent to achieve an embedding from the reference trajectory that is closest to the current frame.

A.2 DATA COLLECTION PROCESS

For collecting perfect dataset, we either train an expert policy using existing algorithms (Andrychowicz et al., 2017) or manually specify the actions. For reach, push, and stir, the action space is 3-dimensional that indicates the motion of robot gripper. For the pick and place task, the action space is 4 dimensional where an additional dimension controls to open/close the gripper.

For the extraneous dataset, we also sample actions from the expert policy until a random time step t. Starting from t, the action is sampled from another policy which tries to accomplish another task T' that is usually reaching some random place in the working space. After finishing T', the robot uses expert policy again for the main task.