A Hierarchical Approach for Learning Multi-Step Sequential Tasks from Video Demonstrations

Mostafa Hussein*, Madison Clark-Turner*, and Momotaz Begum

***Abstract*— We are interested in learning the high-level policy of multi-step sequential (MSS) tasks, such as activities of daily living, from video demonstrations. Videos of MSS tasks are typically long in duration and exhibit large feature variance, especially when captured in non-engineered settings. Learning task policy from such videos using state-of-the-art end-to-end approaches is sample inefficient due to a reliance on pixel-level information. Understanding the unique temporal structures of MSS tasks can make policy learning easier and sample efficient. However, understanding this temporal structure requires analyzing the entire content of the video or a task which is a complex and under-explored area in the current literature. We propose a hierarchical solution to this problem where i) an automated feature selection process extrapolates temporally grounded task-relevant features from the video and then ii) a stochastic policy learning model learns a feature-constrained task policy. The proposed model is sample efficient (trained using less than one hour of demonstration data) as a result of substituting selected temporally-grounded features for pixel-level information. We demonstrate the efficacy of our proposed framework by teaching a YuMi robot a long-duration, multi-step task – tea making – from videos. We present results on sample efficiency and robustness against data loss. We also compare our performance to that of a state-of-the-art approach of task learning from real visual demonstrations.**

I. Introduction

We envision a future where robots serve lay users in homes and workplaces. Visual demonstration is probably the most convenient way for a layperson to teach a robot new tasks. Many real-world tasks that a service robot is expected to perform have multiple-steps and are sequential, e.g. making a cup of tea, following a recipe when cooking, or preparing a dinner table, etc. We recognize such processes as MSS tasks (multi-step, sequential tasks) in the remainder of this paper. Our goal is to learn MSS tasks from visual demonstrations of human collected in natural environments. However, this is a challenging problem setting for vision-based learning from demonstrations (LfD) since i) demonstration videos have long duration (typically tens of seconds) and exhibit large variations among visual features, ii) videos must be explored entirely to understand the broader temporal context of the task, and iii) the task must be learned from only a handful of video demonstrations; as there is no access to simulators to provide additional demonstrations or to allow the robot to explore the effect of its primitive actions. To the best of our knowledge, no framework exists that can learn a MSS task under these constraints.

* Denotes equal contribution. Authors are with the Cognitive Assistive Robotics Lab, University of New Hampshire, {Mostafa.Hussein, Madison.Clark-Turner, Momotaz.Begum}@unh.edu

Interest in task learning from videos has increased in recent years [9], [11], [16], thanks to advances in convolutional neural networks (CNNs). However, the majority of vision-based LfD research deals with learning low level trajectories for simple manipulation tasks [12], [21], [33]. These are single-step tasks that can be learned from isolated images without understanding the temporal context. On the contrary, effective learning of most MSS tasks require an understanding of the broader temporal context [16], e.g. in the context of a tea-making task, *the water should be boiled before it is added to a cup*. Regardless of the policy learning method (e.g. behavior cloning (BC) [35], reinforcement learning (RL) [4], or inverse RL [29]), almost all vision-based LfD works require a large amount of training data which are generated through either simulators [11] or alternative real-world means that are not feasible for MSS tasks such as automated self-play which can occur over thousands of hours [16], [32]. A handful of recent works focus on learning MMS tasks from videos [25], [26], [28], [34], [43], [47]. Some of them require specialized training data such as videos from multiple view-points [47], some need access to a simulator [25], while others require a separate training phase where the robot learns through interacting with the environment [34], [43]. The MSS task learning work most closely related to our work is the GTI [26] which also do not assume any access to a simulator. The method however suffers from sample inefficiency. Additionally, all of these works deal with MSS tasks consisting of only three to five steps and are not long in duration. Another family of research leverages the idea of meta-learning for task learning from a very few video demonstrations [15], [38], [46]. Although some of these methods may learn a task from as few as a single demonstration, training the meta learner requires a massive number of demonstrations of related tasks. Overall, learning MSS tasks in a sample efficient manner from long-duration videos while also understanding the broader temporal context of the task is a challenging LfD problem that is largely under-explored. We propose a hierarchical solution to this problem. The proposed approach leverages two facts: **first**, understanding the innate temporal structures of MSS tasks from videos will benefit policy learning and **secondly**, instead of probing every pixel in a video when conducting policy learning (an approach employed by CNN-based approaches [15], [16], [21]), focusing only on task-relevant features makes the learning more sample efficient. However, *how to find features in a video that are task-relevant?* is itself a critical research question. We hypothesize that the first fact provides the answer: through capturing

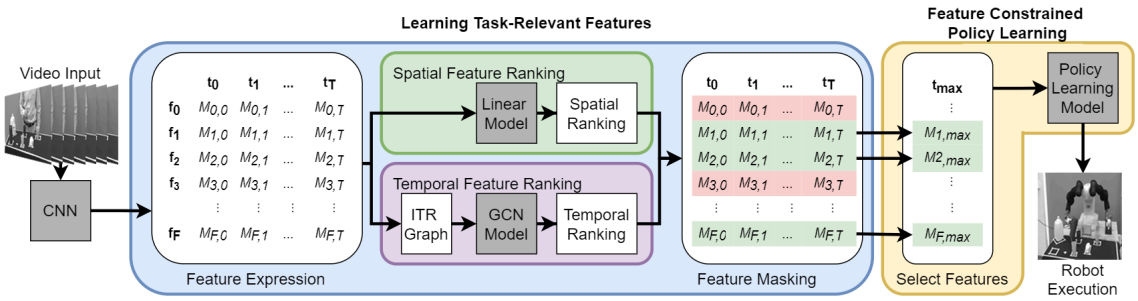


Fig. 1: The proposed approach: task relevant temporally grounded visual features are extracted from input videos (blue box; Section III-A) and used by a policy learner to learn a high-level task policy (yellow box; Section III-B)

temporal structure of the task. Accordingly, we propose an approach for learning MSS tasks from videos in two steps:

Step 1. Learning task-relevant features: This step identifies a set of temporally grounded features that characterize the nature of the MSS task from the demonstration videos. We propose a method to identify temporally grounded visual features from raw videos while leveraging concepts from Allen's interval algebra [1]. We show empirically that these features describe MSS tasks better than visual features that are only spatial in nature. This step requires that video demonstrations of the task are segmented, to create state-action pairs.

Step 2. Feature-constrained policy learning: This step learns a stochastic policy that is biased only toward the features (from Step 1) that characterize the task. We accomplish this with a policy learner that leverages the concept of feature expectation matching (FEM) and the maximum entropy principle (MEP). Fig. 1 shows an overview of this process. Note that the very concept of separating learning of features and policy is not new [24]. However, hand-selecting features in all prior works (as opposed to choosing features in a data-driven manner) severely impact the generalization ability of the learning process. The proposed approach overcomes this limitation. It is important to note that our focus is to learn the high-level policy – the sequences of sub-tasks that can reach the desired goal state – not the motor actions. We used kinesthetic teaching [5] to teach the motor actions required for executing different sub-tasks.

Our primary contribution lies in proposing a novel vision-based LfD framework where a feature selection strategy works with a complementary policy learning algorithm to learn MSS tasks from only a handful of video demonstrations. We demonstrate the proposed framework's ability to learn a MSS task – namely, tea-making – from videos. The task contains 7 sub-tasks and with a mean duration of 2 minutes. Also, we empirically demonstrate that i) visual features that understand the temporal context of the task contribute more toward policy learning than those that are purely spatial in nature, ii) task-relevant features make the policy learning sample efficient, and iii) the proposed framework is not highly sensitive to video segmentation error, as might be caused by a video segmentation algorithm. The proposed approach assumes a known set of actions, a common assumption in all high-level policy learning tasks. Additionally, the features being used to train the model are assumed to be present in the visual dataset. We compared against one of the most recent approaches GTI [26] as it is the closest to our work.

II. Related Work

A. Policy learning from visual data

A majority of the existing vision-based LfD architectures rely on CNNs to learn neural network policies as supervised learning, also called behavior cloning (BC). Almost all of these works learn single-step continuous control tasks such as navigation [31], manipulation [21], [33], and grasping [12]. Due to the reactive nature of these tasks, these approaches can use CNN inference on individual frames of the video data rather than processing the entire video as an observation in its own right. They often require large pools of publicly available training data [39], leverage established simulators to augment their datasets [11], or investigate tasks that are simple enough that robots can autonomously collect visual experiences without the use of human demonstrators [32]. These accommodations are unrealistic for MSS tasks.

Recently, hierarchical imitation learning based approaches are showing a great promise for learning MSS tasks from videos [25], [26], [28], [34], [43]. In these methods a high-level policy predicts sub-tasks and a low-level policy computes motor commands to execute different sub-tasks. The major critic of this line of work is that they require different types of special accommodations. For example, the work in [34] require a third network – in addition to a sequence to sequence network [40] that generates the sequence of sub-tasks – to detect and generate the position of each object in the task. The method in [47] require images from multiple view-points to train two clustering algorithms – for sub-task identification – and two neural networks – for generating required actions – triggering the need of extensive amount of parameter tuning to achieve the reported accuracy. However, the most constraining requirement in [47] is the joint angles of the robot which nullifies the appeal of the visual imitation learning. The methods in [25], [28] rely on a hierarchical model to predict the sub-goal sequence and then use an RL technique to generate the low-level motor commands. However, using an RL technique requires a task simulation that might not be available for each task. Alternatively, the robot can be allowed to interact with the world to learn

safe movements – an approach which is feasible in the controlled lab setting yet completely non-viable when a lay user is training the robot in the wild. A common critique of all existing work on visual imitation learning is sample inefficiency – it requires a between 300 to 700 samples to train the complex set of perception networks involved with policy learning [34].

In the recent years meta learning based approaches are showing promises in task learning from a very few demonstrations. [38], [46]. Although a specific task can be learned from as few as a single demonstration, the meta learner is trained with a large number of demonstrations of related tasks – e.g. 7194 demonstrations in [46]. To this date, meta learning allows to generalize to a new task that has a very high degree of similarity (both in task structure and the environment) with the learned ones.

B. Learning broader temporal context from videos

Several deep learning models have been developed to learn temporal information from video data but they represent data in a manner that is ill-suited for the vision-based LfD problem setting we are interested in. For example, recurrent neural networks (i.e. long short-term memory [42]) develop temporal features by aggregating frames of the visual input sequentially. The resulting information is notoriously difficult to interpret and learned representations often fixate on frame-to-frame variances, inhibiting the ability to capture long-term dependencies in the data [45]. Convolution-based approaches (1D convolutions [41], 3D convolutions [7], and pyramidal structures [44]) are recognized for placing a greater significance on spatial features as opposed to the temporal contents of video [6]. Capturing temporal features with graphical approaches is the best option for generating easily accessible, temporally focused representations of the data. However, these methods are novel and existing implementations explore spatial relationships instead of temporal features [45].

III. Vision-based Learning of MSS Tasks: A Hierarchical Approach

A. Learning task-relevant features

The hallmark of most MSS tasks is their high-level temporal structure, e.g. in the context of a tea-making task, *turning on the kettle after pouring the water in* or *stirring the cup after milk or sugar is added* etc. We aim to learn this to facilitate policy learning in a sample efficient manner. We propose a novel, temporally-informed ranking approach that selects a subset of visual features from a video based on their temporal significance. First, we identify temporal features expressed in videos using a novel wrapper that explicitly investigates the temporal importance of CNN-learned visual features. This temporal information is used to generate a ranking over the available visual features.

1) Temporal Feature Identification: Temporal feature identification occurs as a pipeline using the learned spatial features to generate a graph of temporal relationships. This graphical structure, an interval temporal relationship (ITR) graph, is used to train a graph convolutional network (GCN) which is then queried by the proposed ranking approach. Fig. 2 shows an overview of this process.

Spatially-expressed visual features are obtained from any standard CNN backbone model that has been fine-tuned to identify some of the spatial features present in a task-related dataset. The backbone model and the fine-tuning procedure used in this work are discussed in Section IV-B. Frames from the demonstration videos (Fig. 2a) are parsed into a CNN backbone model to generate activation maps M (Fig. 2b). Each activation map is a 4-dimensional tensor ($F \times H \times W \times T$) denoting the relative expression of learned features, where F: the number of features, H: height, W: width, and T: time. Fig. 2b uses darker shades to denote stronger feature expression. We propose a method to infer the presence of temporal features in a video observation from an activation map. We begin by reducing the spatial dimensions of our activation map. We apply the maximum function over H and W of the activation map to reduce the representation to two dimensions ($F \times T$). The reduced representation, shown in Fig. 2c, clearly indicates the relative expression of each feature at each time. We threshold this expression in order to determine when a feature is and is not being expressed. This distinction is useful for distinguishing the nature of the temporal relationship that exists between two spatial features. We select a different threshold value (Φ_f) for each feature ($f \in F$) using the average expression of that feature over the entire dataset. We investigated other thresholding metrics and found none performed as well as the one described. Values in the activation map that fall below (Φ_f) are set to 0. The thresholded activation map (Fig. 2d) can be transformed into a graph. Each node in this graph denotes a different *spatial event*, a period when a specific spatial feature is actively expressed in the observation. We define these nodes with a four tuple $< t_s, t_e, f, f_{max} >$ containing the time step when the feature began (t_s) and stopped (t_e) being expressed, the feature label (f), and the maximum expression observed from that feature over the duration it was expressed (f_{max}). Directed edges between these nodes are identified by leveraging Allen's interval algebra (IA), a principled set of relationships that define how two temporal events overlap [1]. Using 7 temporal relationships (*before* (b), *meets* (m), *overlaps* (o), *during* (d), *starts* (s), *finishes* (f), and *equals* (e)) we relate each pair of nodes in our graph using t_s and t_e. The complete process results in a graph (Fig. 2e) whose nodes represent periods of event expression (captured by f and f_{max}) and whose edges denote the temporal relationships that exist between those nodes. This graph representation is used to train a GCN classifier which learns the discriminatory temporal features present in the ITR graph. GCN inference begins at the nodes of an ITR graph and with iterative convolutions extends a complex representation to neighboring nodes along the graph's edges (temporal relationships). Informative temporal relationships in the graph are recognized and reinforced while uninformative relationships are pruned.

2) Feature Ranking: We have used a popular ranking approach, erasure ranking [22], to identify the most important

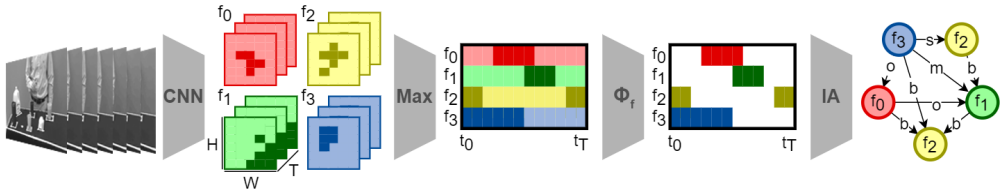


Fig. 2: Extraction of temporal feature graph. See section III A for details

Fig. 3: Task Setup

features in the videos by class label ($c \in C$). Erasure ranking iterates through each of the input features and evaluates the trained model's performance when a specific feature is removed. The greater the impact on the model's logit values (V) the greater the significance that the given feature contributes towards the trained model. The ranking (R) of a feature is evaluated as a normalized sum over all examples (e) of a given class in the demonstration set ($\mathcal{D}$) and can be calculated using:

$$R(f,c) = \sum_{e \in \mathcal{D}} \frac{V(e,c,f) - V(e,c,\neg f)}{V(e,c,f)} \tag{1}$$

We rank the temporally-grounded visual features extracted using the method discussed in Section III-A.1. For this we apply erasure search over the feature labels that compose the ITR graph, removing each node (and all connected edges) that have the given feature label f. We also rank features that are purely spatial in nature. For that, we train a linear model using the reduced activation maps in Fig. 2c, which do not capture the explicit temporal relationships between features, and investigate how the accuracy performs as each feature is removed. Our goal is to compare the efficacy of temporally-grounded and purely spatial features in policy learning.

Each video in the demonstration set $\mathcal{D}$ is hand-segmented into M segments where M is the number of actions in the task. For each video segment, a set of ranked features $\{s_j\}_{j=1}^{M}$ is extracted using the methods discussed in this section, $s = \{f_n\}_{n=1}^{N}$, N: the number of features from each segment. These sets, along with the action labels $\{a_j\}_{j=1}^{M}$, are passed to the policy learning algorithm as state-action pairs (s_j, a_j) to perform feature-constrained policy learning. It is important to note that video segmentation is required only during training, not at run-time. This step can be automated using any of these recent action segmentation approaches(e.g. [13], [23]. Action segmentation is a standalone field of research and it is beyond the scope of this paper.

B. Feature-constrained policy learning

The goal of the policy learner is to learn a stochastic policy $\pi(a|s)$ that matches the expert policy $\tilde{\pi}(a|s)$ demonstrated in the videos. For methodological correctness, we assume, as our basic framework, a Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, \rho_0)$ with the stochastic shortest path objective (assuming some terminal states) [4], where $\mathcal{S}$ is the state space ($s \in \mathcal{S}$), $\mathcal{A}$ is the action space ($a \in \mathcal{A}$), and $\rho_0 \in \Delta^{\mathcal{S}}$ represents the distribution over the initial state. Here, $\Delta^{\mathcal{S}}$ denotes the probability simplex over the set $\mathcal{S}$. The unknown transition probabilities are $P : \mathcal{S} \times \mathcal{A} \to \Delta^{\mathcal{S}}$ and the unknown rewards are $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. We also assume that the distribution $\tilde{\pi} : \mathcal{S} \in \Delta^{A}$ over the states that represents the expert's probability of visiting the state is uniform over the states in $\mathcal{D}$.

We use the features identified in Section III-A to learn $\pi(a|s)$ and assume nothing else about the model, to be as uniform as possible with all of the unknowns. We learn such a policy $\pi(a|s)$ by leveraging two ideas: feature expectation matching (FEM) [8], [30] and Maximum Entropy Principle (MEP).

1) Learning a policy that matches a set of feature expectations: We want to learn a policy $\pi(a|s)$ that accords with a set of continuous task-relevant features $f_i, i = 1, 2, \ldots, n$ that are derived from the demonstration set $\mathcal{D}$ by the feature learning module discussed in Section III-A. The $f_i(s,a)$ are the value of the feature f while we are in a state s and taking action a. A popular way to impose such a restriction on $\pi(a|s)$ is to make it satisfy the following equality, also known as FEM [8], [30], [49], where the feature expectations are computed as follows:

$$\begin{aligned} \mathbb{E}_{\tilde{\pi}}[f_i] &= \mathbb{E}_{\pi}[f_i], i \in \{1, 2, \ldots, n\} \\ \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \tilde{p}(s)\tilde{\pi}(a|s) f_i(s,a) &= \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \tilde{p}(s)\pi(a|s) f_i(s,a) \end{aligned} \tag{2}$$

This is a general definition of the features in $f_i(s,a)$. If the features used to describe an action are different, we would need to loop over the actions but this is uncommon occurrence as we demonstrate in our experiments (Section IV-C.1). Here $\tilde{p}(s)$ is empirical distribution of the states. Equation (2) will give us a policy $\pi(a|s)$ that has the same expected values for feature f_i as seen in the demonstration $\mathcal{D}$. Simplified, if we see in the videos that an action a was taken an average q times in state s, the learned policy will take that action at that state with a probability q/M where M is the total number of video demonstrations. The only issue here is that in the space of all possible $\pi(a|s)$, there are many distributions which observe this constraint. To pick one specific policy, we further constrain $\pi(a|s)$ to be completely unbiased to all other features that may exist in the feature space. A popular way to realize this is finding the distribution $\pi(a|s)$ that has the maximum entropy [3].

2) Learning a policy that has the maximum causal entropy: The causal entropy of the condition distribution

$\pi(a|s)$ is expressed as follows.

$$H(\pi) \equiv -\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \tilde{p}(s)\pi(a \mid s)\log\pi(a \mid s) \tag{3}$$

Among all $\pi(a|s)$, we aim to pick the one that has the maximum entropy. Accordingly, we solve the following optimization problem to compute the policy $\pi(a|s)$:

$$\begin{aligned} \max_{\pi\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} \quad & H(\pi) \equiv -\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \tilde{p}(s)\pi(a|s)\log\pi(a|s) \\ \text{s.t.} \quad & \mathbb{E}_{\tilde{\pi}}[f_i] - \mathbb{E}_{\pi}[f_i] = 0 \quad i = 1,\ldots,n \\ & \sum_{a\in\mathcal{A}} \pi(a|s) - 1 = 0 \quad \forall\ s\in\mathcal{S} \end{aligned} \tag{4}$$

Using the standard convex duality arguments (for space we have omitted the full proof), we can see that the optimal solution π to (4) must satisfy, for some Lagrange multipliers $\lambda_i \in \mathbb{R}^N$, that [2], [3]:

$$-\Bigg\{ \max_{\lambda}\ \Lambda(\lambda) \equiv -\sum_{s\in\mathcal{S}} \tilde{p}(s)\log z_\lambda(s) + \sum_{i=1}^{N}\lambda_i \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \tilde{\pi}(s,a) f(s,a) \Bigg\} \tag{5}$$

where $z_\lambda(s) = \sum_{a\in\mathcal{A}} \exp\left(\sum_{i=1}^{N}\lambda_i f_i(s,a)\right)$ is a normalization constant. As mentioned earlier, it is likely that each action will be defined by different features, therefore, we need to specify $f_i(s,a)$. However, in our experiment we used the same set of features for all the actions, a common practice among most learning tasks [17], [49]. By solving Equation (5) we can get a generalized solution:

$$\pi(a|s) = (z_\lambda(s))^{-1} \cdot \exp\left(\sum_{i=1}^{N}\lambda_i f_i(s,a)\right) \tag{6}$$

The sample efficiency of the proposed framework can be understood through Equation (6). The policy expression in (6) relies only on n task-related features extracted from videos using the method discussed in Section III-A. As long as these features represent the task with a certain accuracy, the policy in Equation (6) will allow a robot to execute the task as seen in the video demonstrations.

Note that a widely popular approach for inverse reinforcement learning from demonstrations, Max-Ent IRL [49] and later variants [17], also matches feature expectations while maximizing entropy. Our policy learner, however, follows the idea of behavior cloning (BC) [18]. This is primarily because BC does not require access to additional data (collected with a simulator) nor does it require the knowledge of the full system dynamics as in [48], [49]. The BC, therefore, is an appropriate tool for learning real-world tasks.

C. Policy Execution

The only run-time assumption we make is that the task always begins from the same initial state. The policy execution follows these steps at run-time: 1) Image frames captured by a camera are fed to the feature learning module discussed in Section III-A. We process 64 frames at a time. This generates f_i, a set of features representing the current state. These features are pruned according to the ranking method selected by the user (temporal or spatial). Of the remaining features we select the max value of the feature f_i. This is the final representation of the current state of the task. 2) Using the current state of the task and the learned model in (6) we generate a probability distribution over the high-level actions and choose the high-level action with the highest probability to perform next. 3) Finally, the appropriate controls are sent to the robot to execute the high-level action. We used kinesthetic teaching [5] to teach the robot how to generate the controls required by the different high-level actions (see the attached video). Note that video segmentation is required only during training, not at run-time.

IV. Evaluations

We conducted three experiments to evaluate the proposed framework with respect to the contribution of temporally-grounded features towards accuracy, sample efficiency, and robustness against video-segmentation accuracy. We also compared the performance of the proposed method against that of GTI [26] – a closely related work on learning MSS tasks from video demonstrations. All experiments are done in the context of learning a tea-making task.

A. Demonstrations

We created a video dataset that focused on a singular MSS task: making a cup of tea. Fig. 3 captures the environment where the demonstrations were collected from four participants in an IRB-approved study. Tea making tools (pitchers, teabag, sugar, etc.) were placed in fixed locations on a table. The available actions were: *turn on/off the oven*, *add water*, *add sugar*, *add milk*, *add teabag*, and *stir*. We Participants were asked to use all of the actions while following one specific sequence so that we can study the effectiveness of our feature selection approach and our ability to learn a correct policy purely from videos). Each participants provided 12 demonstrations, resulting in 48 videos. Each video lasts for 2 minutes on average, making the data collection process very quick (less than one hour). The video demonstrations were subsequently segmented and labeled by hand. Videos were collected at 30fps and down-sampled to 10fps. Our dataset and code are available here[10].

B. Feature and policy learning

As discussed in Section III-A, we rely on a CNN-backbone to identify spatial features. Our work specifically uses VGG-16 [37] to capture the presence of spatial features in the actions, though other models could be employed. VGG-16 (originally trained on ImageNet) was fine-tuned on the tea-making dataset. The CNN was trained to recognize the action labels using 64 frames sampled uniformly from each video. The frames were reduced to 224×224 pixels in size and subject to background subtraction before being fed into the CNN. The model was trained using an Adam optimizer with a learning rate of $1e-3$ over 50 epochs. The ITR

graph contains an exponential number of edges for each spatial feature being investigated, to constrain the number of computations required by our model we apply a bottleneck layer prior to the inference layer of the network, condensing the number of features inferred by the model from 2048 to 32.

As discussed in Section III-A, a GCN was used to perform temporally-grounded feature ranking. Specifically, we employed R-GCN for its ability to learn from discrete edge labels [36]. A simple linear layer was used to make class inference when performing purely spatially grounded feature ranking. In both cases, the networks were trained using the same optimizer, learning rate, and number of epochs as were used to train the backbone model.

C. Experiments

1) Policy accuracy and the role of good features: The goal of this experiment is to demonstrate how task-relevant features, identified through our proposed approach, help to learn an accurate policy in a sample efficient manner. The accuracy of a policy is defined as *(Number of correctly chosen actions / Total number of executed actions)* $\times 100$. The proposed approach follows the principles of BC for policy learning and this accuracy metric provides a clear indication of how well the mapping function maps states to actions. We compare the policy accuracy between: task-features f_i formed using high-ranked temporally-grounded features and task-features formed from high-ranked spatial features. We further investigated the number of task-features required by each cases to achieve different degrees of accuracy. Finally, we investigate the relationship between accuracy and the number of samples (i.e., the video demonstrations). Figs. 4a and 4b show findings from these experiments. For the same number of samples and task-features, the policy accuracy is higher when task-features are temporally-grounded as compared to when they are purely spatial in nature. An accuracy of 100% can be achieved with 30 temporally grounded task-features collected only from 15 demonstrations whereas 30 demonstrations are needed to achieve the same accuracy with the same number of spatial task-features. Even with only 5 demonstrations, 5 temporally grounded task-features can achieve an accuracy of 60%. But more than 25 spatially-grounded features were required to achieve the same accuracy. These results show the significant role temporally grounded features play in learning good policies for MSS tasks from a handful of video demonstrations.

We perform a visualization-based analysis to try and discern the properties of the temporally-grounded features that lead to their greater significance. Fig. 6 shows several of the features ranked highly by the spatial (a) and temporal (b) ranking approaches in the context of the 'Add Water' action. From a qualitative perspective the spatially-grounded features are redundant and capture many of the same visual properties at approximately the same time points. If these properties are not visible in the video, then they are likely to be absent across several features (dark blue and dark yellow). In contrast the features highlighted by the temporal ranking approach are scattered throughout the video. These features capture different aspects of the interaction such as grasping and replacing the pitcher (red) and different stages of pouring the pitcher (blue and yellow).

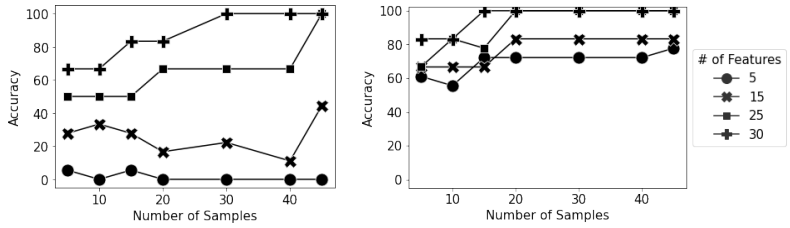
Frame Loss	Model Accuracy
0%	100.0%
10%	100.0%
20%	92.0%
30%	75.0%
40%	75.0%

TABLE I: Robustness

2) Sample efficiency: The goal of this experiment is to compare the sample efficiency of our proposed method with a popular vision-based BC baseline [27], [35]. We define sample efficiency as the number of samples (complete demonstration) that are required to learn a specific task. As a baseline, we compare to the established BC method, which models π_{BC} using a neural network with parameters θ_{BC}. We find these parameters using maximum-likelihood estimation: $\theta_{BC} = \arg\max_\theta \prod_{(s,a)\in D} \pi_{BC}(a|s)$. With a given dataset of state-action pairs, we split the dataset using 70% of the demonstrations for training and the remainder for validation. We train the policy with supervised learning using ADAM [19], until the validation error stops decreasing. Both the proposed policy learner and the BC baseline are trained using the top 30 temporally-grounded features. The sample efficiency results are shown in Fig. 5. Our method dominates the BC baseline, achieving a 100% accuracy using only 15 samples while the later required all 48 samples to achieve the same result. As the feature input is the same, the sample efficiency of our method can be directly attributed to the way we develop our action distribution according to FEM and MEP, which results in a more robust policy. For comparison, the baseline BC method optimizes a maximum likelihood function over the entire dataset.

3) Robustness: The proposed model requires hand segmentation of the demonstration videos to generate task-relevant features. We investigate the sensitivity of the learned policy as a factor of segmentation accuracy. We evaluate this by manually introducing segmentation error to our data by removing frames from the beginning and ends of each video snippet. We then calculate the accuracy of the policy learning model to determine how robust our policy is to data loss. Using the 30 top-ranked temporally grounded features we evaluate the policy accuracy with increasing degrees of data loss (Table. I). The accuracy of the model does not drop until 20% of the frames (12 frames) have been deleted where upon it drops to 92%. Removing additional frames up to 40% (26 frames) reduces the overall accuracy of the model to 75%. With almost half of the video eliminated we were still able to maintain a relatively high policy accuracy.

These results show that the policy is tolerant to segmentation errors. These results suggest that we can replace hand-segmentation with any off-the-shelf video segmentation



(a) Spatial features (b) Temporal features

Fig. 4: Accuracy of the learned policy where the states are defined by the most highly-ranked features according to the two different ranking methodologies.

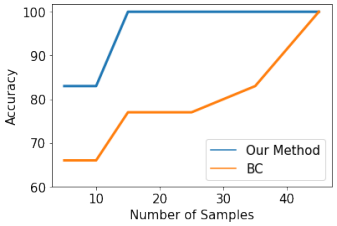
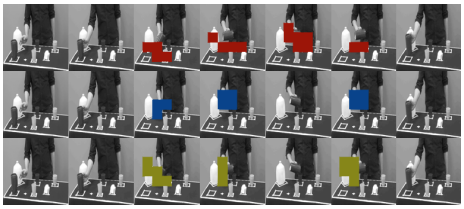
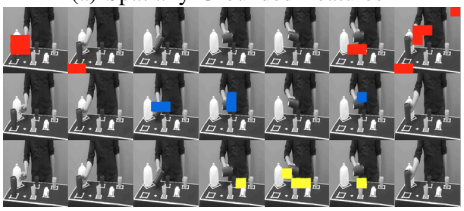


Fig. 5: Sample efficiency results



(a) Spatially-Grounded Features



(b) Temporally-Grounded Features

Fig. 6: The expression of spatial features identified as highly significant to the spatial and temporal models. Frames of the video are depicted in grey-scale with colored regions denoting areas of intense feature expression.

algorithm (e.g. [13], [23]) which will always generate a certain amount of error when segmenting videos. Note that the current implementation makes two assumptions about video segmentation performance namely, segmentation labels are not duplicated and a specific set of actions are visible in each demonstration. Design of video segmentation algorithms with these two attributes is an active research area in computer vision and is a different research topic from the contribution of this paper. However, incorporating such algorithms, as soon as they are available, in the proposed framework is a focus of our future work.

4) Comparison with GTI [26]: The proposed approach and GTI [26] shares the same core idea of learning a MSS task from a handful of videos and without accessing a simulator. We trained the GTI with 48 video demonstrations of the tea-making task. we leveraged the implementation used in [1] to do the comparison.

The GTI and the proposed approach achieved **65.0**% **and 100.0** % accuracy, respectively in learning the tea-making task. In order to make a fair comparison, this accuracy is calculated only in predicting the high-level actions or sub-tasks. This is because the proposed approach assumes no loss in accuracy from kinesthetic teaching-based motor command executions while the GTI learns the motor command in addition to the high-level actions. We can attribute the poor performance of GTI to two factors: a) number of demonstrations: The GTI uses conditional Variational Autoencoder (cVAE) [20] with a ResNet-18 [14], both are known as data-hungry models. Training them only with a handful of videos caused the model to lose accuracy. b) Task-complexity: The original GTI model in [26] was tested with a much simpler task with only 3 to 4 sub-tasks where each sub-task involves moving and/or placing pots and lasted for 5 seconds. Tea-making is a more complex task with longer duration (2 minute) and involves more complex sub-tasks each of which lasted for 15 seconds. The results imply that GTI does not generalize well to long-duration tasks – such as tea-making – investigated in this paper.

[1] https://github.com/UT-Austin-RPL/BUDS

V. Conclusion

Contemporary vision-based LfD models are insufficient to learn MSS tasks in a sample efficient manner. The proposed forked approach toward task learning autonomously learns task-relevant features which then guides feature-constrained policy learning. Evaluated on a tea making MSS task we demonstrated the superiority of a temporally cognizant feature ranking approach compared to traditional spatial feature focused methods, when selecting task-relevant information to drive our policy learner. Additionally, our policy learner was able to leverage these features to ensure high accuracy in a sample sparse dataset. Our future work will investigate unexplored areas such as variations among demonstrations and the presence of repetitive actions.

Acknowledgment

This work was supported in part by the National Science Foundation (IIS-1830597).

References

[1] James F Allen and George Ferguson. Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531–579, 1994.
[2] Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.
[3] Adam Berger, Stephen A Della Pietra, and Vincent J Della Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.
[4] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.
[5] Aude G Billard, Sylvain Calinon, and Florent Guenter. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54(5):370–384, 2006.

[6] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10618–10627, 2020.

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[8] Sonia Chernova and Andrea L Thomaz. Robot learning from human teachers. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(3):1–121, 2014.

[9] Madison Clark-Turner and Momotaz Begum. Deep reinforcement learning of abstract reasoning from demonstrations. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 372–372. ACM, 2018.

[10] Madison Clark-Turner and Mostafa Hussein. Hierarchical learner, 2021. https://github.com/AssistiveRoboticsUNH/hierarchical_learner.

[11] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[12] Elias De Coninck, Tim Verbelen, Pieter Van Molle, Pieter Simoens, and Bart Dhoedt. Learning robots to grasp by demonstration. *Robotics and Autonomous Systems*, 127:103474, 2020.

[13] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.

[14] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016.

[15] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*, 2017.

[16] Wonjoon Goo and Scott Niekum. One-shot learning of multi-step tasks from observation via activity localization in auxiliary video. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7755–7761. IEEE, 2019.

[17] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[18] Jonathan Ho, Jayesh Gupta, and Stefano Ermon. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning*, pages 2760–2769. PMLR, 2016.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[22] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

[23] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6243–6251, 2019.

[24] Toby Jia-Jun Li, Tom Mitchell, and Brad Myers. Interactive task learning from gui-grounded natural language instructions and demonstrations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 215–223, 2020.

[25] Zhihao Li, Zhenglong Sun, Jionglong Su, and Jiaming Zhang. Learning a skill-sequence-dependent policy for long-horizon manipulation tasks. *arXiv preprint arXiv:2105.05484*, 2021.

[26] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. *arXiv preprint arXiv:2003.06085*, 2020.

[27] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[28] Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. *arXiv preprint arXiv:1909.05829*, 2019.

[29] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[30] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *arXiv preprint arXiv:1811.06711*, 2018.

[31] Xinlei Pan, Tingnan Zhang, Brian Ichter, Aleksandra Faust, Jie Tan, and Sehoon Ha. Zero-shot imitation learning from demonstrations for legged robot visual navigation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 679–685. IEEE, 2020.

[32] 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*, pages 2050–2053, 2018.

[33] Affan Pervez, Yuecheng Mao, and Dongheui Lee. Learning deep movement primitives using convolutional neural networks. In *2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids)*, pages 191–197. IEEE, 2017.

[34] Sören Pirk, Karol Hausman, Alexander Toshev, and Mohi Khansari. Modeling long-horizon tasks as sequential interaction landscapes. *arXiv preprint arXiv:2006.04843*, 2020.

[35] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[36] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] 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.

[39] Jaeyong Sung, Seok Hyun Jin, and Ashutosh Saxena. Robobarista: Object part based transfer of manipulation trajectories from crowdsourcing in 3d pointclouds. In *Robotics Research*, pages 701–720. Springer, 2018.

[40] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[41] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[42] Jin Wang, Liang-Chih Yu, K Robert Lai, and Xuejie Zhang. Dimensional sentiment analysis using a regional cnn-lstm model. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 225–230, 2016.

[43] Bohan Wu, Suraj Nair, Li Fei-Fei, and Chelsea Finn. Example-driven model-based reinforcement learning for solving long-horizon visuomotor tasks. *arXiv preprint arXiv:2109.10312*, 2021.

[44] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020.

[45] Jingran Zhang, Fumin Shen, Xing Xu, and Heng Tao Shen. Temporal reasoning graph for activity recognition. *IEEE Transactions on Image Processing*, 29:5491–5506, 2020.

[46] Yuxiang Zhou, Yusuf Aytar, and Konstantinos Bousmalis. Manipulator-independent representations for visual imitation. *arXiv preprint arXiv:2103.09016*, 2021.

[47] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 2022.

[48] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. Modeling interaction via the principle of maximum causal entropy. 2010.

[49] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.