Learning Sequential Human-Robot Interaction Tasks from Demonstrations: The Role of Temporal Reasoning

Estuardo Carpio, Madison Clark-Turner and Momotaz Begum

***Abstract*— There are many human-robot interaction (HRI) tasks that are highly structured and follow a certain temporal sequence. Learning such tasks from demonstrations requires understanding the underlying rules governing the interactions. This involves identifying and generalizing the key spatial and temporal features of the task and capturing the high-level relationships among them. Despite its crucial role in sequential task learning, temporal reasoning is often ignored in existing learning from demonstration (LfD) research. This paper proposes a holistic LfD framework that learns the underlying temporal structure of sequential HRI tasks. The proposed Temporal-Reasoning-based LfD (TR-LfD) framework relies on an automated spatial reasoning layer to identify and generalize relevant spatial features, and a temporal reasoning layer to analyze and learn the high-level temporal structure of a HRI task. We evaluate the performance of this framework by learning a well-explored task in HRI research: robot-mediated autism intervention. The source code for this implementation is available at https://github.com/AssistiveRoboticsUNH/TR-LfD.**

I. Introduction

Learning from Demonstration (LfD) is a popular robot learning paradigm in which the goal is to develop a policy for performing a task based on a set of demonstrations provided by a human teacher [1], [2]. LfD has been employed to teach robotic systems low-level tasks such as generalizing the motion trajectories needed to perform obstacle avoidance [3], pick-and-place operations [4], or furniture assembly [5], [6]. Similarly, LfD has been employed to learn policies for high-level tasks such as object sorting [7] and domestic activities [8], [9]. The majority of LfD frameworks derive policies by focusing on the key spatial features of the task, disregarding the temporal structure. Many HRI tasks can be defined by their rigid temporal structure. A prominent example is the highly explored HRI domain of robot-mediated intervention (RMI) for autism spectrum disorder [10]. Each RMI follows a set of activities to be performed by a robot subjected to student responses, with each activity-response pair being fully pre-defined. Although almost all RMI in the existing literature are either tele-opearted or pre-coded, if we want to learn an arbitrary intervention solely from observations, the temporal structure of the RMI sessions will act as the major discriminatory feature for policy learning. Our previous work reported in [11], the first attempt to learn an RMI from video observations, highlighted the necessity of incorporating temporal information along with spatial features for learning sequential HRI tasks. In this paper we propose the Temporal-Reasoning-based Learning from Demonstration framework (TR-LfD), which is a holistic framework for learning sequential HRI tasks (such as RMI) from observations.

The proposed TR-LfD is a layered architecture where a Temporal Reasoning Layer (TRL) learns interval temporal relations (ITRs) among discriminative spatial features extracted by a Spatial Reasoning Layer (SRL). ITRs are defined according to Allen's interval temporal Algebra [12]. Learning the temporal structure of a task from demonstration data helps to narrow down the action-space while selecting policy during the real-time execution of a task. We validate the performance of the TR-LfD in learning a RMI for teaching social skills from video observations and show that the policy learning performance improves when the temporal information is leveraged. To the best of our knowledge, this is the first work that employs interval temporal Algebra to incorporate temporal reasoning for learning LfD policy. A layered architecture integrating fully automated spatial and temporal reasoning is another novel aspect of the work proposed in this paper.

II. Related Work

This section reports low- and high-level LfD research that learned policies of HRI tasks while considering spatial and/or temporal features. The work described in [13] introduces an end to end deep reinforcement learning approach that was employed to learn a basic social interaction from raw demonstration data. This model, however, was designed to learn an interaction with simple temporal dynamics in which policy selection could be executed without performing temporal reasoning. In [11] the authors introduce a deep reinforcement learning framework capable of learning a structured human-robot interaction. This framework, although proficient in learning spatial reasoning, failed to learn the underlying temporal rules that govern the interaction.

Several frameworks have been proposed in the low-level LfD literature to incorporate temporal information in the policy derivation process. In [14] a Hidden Markov Model (HMM) was employed to construct skill trees that capture the sequence in which events take place in a task. Similarly, the LfD framework described in [15] learns finite state machines that model the temporal relationships between the events of a task. Meanwhile, the work in [8] introduces influence graphs to model the sequence of events needed to complete a task. None of these works, however, performs spatial and temporal reasoning in an integrated manner. Also, no existing work

This work was supported in part by the National Science Foundation (IIS 1664554)

Authors are with the Cognitive Assistive Robotics Lab, University of New Hampshire, NH, USA {erp48,mbc2004,mbegum}@cs.unh.edu

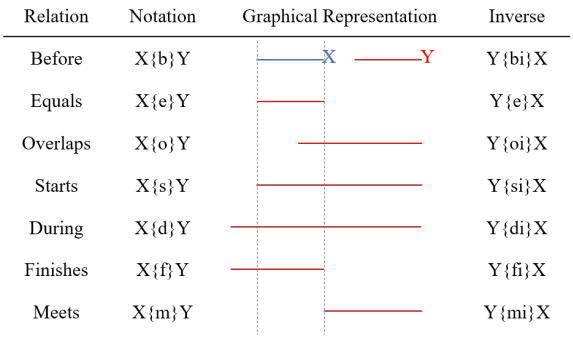


Fig. 1. Set of interval temporal relations that can exist between two events.

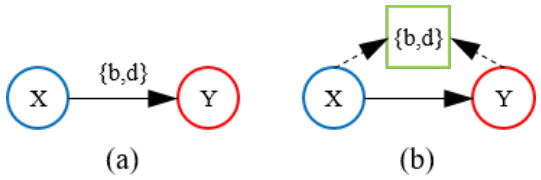


Fig. 2. (a) ITBN model for an activity in which X can happen before or during Y ($X\{b,d\}Y$). (b) BN representation of the ITBN shown in (a).

has considered interval Algebra to model the wide range of temporal relations that typically exist in HRI tasks.

III. Preliminaries

A. Interval Algebra

Complex activities are composed of several events, each defined by a start and a stop time. During the execution of an activity, events can happen simultaneously or in a sequential manner, creating temporal relations and constraints between the events. Allen and Ferguson [12] proposed a set of 13 atomic interval temporal relations that can exist between a pair of events and limit the order in which events can take place in an activity (Fig. 1).

B. Interval Temporal Bayesian Networks

Interval Temporal Bayesian Networks (ITBN) are probabilistic graphical models designed to model the interval temporal relations that exist between the individual events that constitute a complex activity [16]. This is accomplished by combining Bayesian Networks (BN) with Interval Algebra. BNs are graphical models capable of capturing conditional dependencies among random variables using a directed acyclic graph. In an ITBN, each event of an activity is represented by a node in the graph. Meanwhile, each of the edges represents the existence of a temporal relationship between the two events it connects. In an edge that goes from event X to Y, X is the temporal reference of Y, meaning that Y has a temporal dependency on X (Fig. 2).

Zhang et al. [16] proposed algorithms to perform structure and parameter learning on ITBNs. They implemented an ITBN as a BN by introducing a new set of nodes to represent the temporal relationships between two event nodes (Fig 2). This approach allows ITBNs to perform inference using existing BN algorithms.

Structure Learning: This process learns a graphical model that captures the spatio-temporal dynamics of an activity using a training dataset. First, the interval temporal relationships that exist between all the events of an activity are learned using the concept of temporal distance:

$$d(\Omega_Y, \Omega_X) = \left(s_Y - s_X, e_Y - e_X, s_Y - e_X, e_Y - s_X\right) \quad (1)$$

where X is the temporal reference of Y and Ω represents a tuple $[s, e]$ containing the start (s) and end (e) times of an event. The temporal distances for every possible pair of events are then mapped to the atomic temporal relations listed in Fig. 1. Afterwards, an iterative local search procedure [17] is used to generate new candidate networks. These structures are evaluated using the Bayesian Information Criterion (BIC) [18] to select the one that best fits the training data.

Parameter Learning: This process involves finding a maximum likelihood estimate for the parameters of a model from the training data. The algorithm is analogous to the parameter learning process of a BN, with the exception that along with learning the conditional probability for each event node, it is necessary to learn the conditional probability for the temporal relation nodes of the model [16].

IV. Temporal-Reasoning-based Learning from Demonstration (TR-LfD)

TR-LfD is proposed specifically to learn sequential HRI tasks. To better understand the reality of such tasks, let us take as example a standard RMI [19], [20]. Here a robot executes a certain action a (typically through tele-operation) and waits for responses (r) from the child. A positive response from the child causes the robot to trigger a follow-up action (typically a reward, R), while a negative response triggers a different follow-up action (typically a feedback, F) before the robot formally ends the interaction (e). Thus, the most primitive structure of this sequential HRI task can be described as follows: $a \rightarrow r \rightarrow R$ or $F \rightarrow e$. This primitive structure can be used to learn any arbitrary RMI or other ritualistic HRI tasks while using task-specific definitions for a, r, R, F and e. More complicated interactions can also be defined using this primitive structure. The objective of TR-LfD is to learn this primitive interaction structure from video based observations.

Despite its simplistic appearance, learning this structure from video observations is a complex problem for various reasons. For example, the discriminatory state features may vary due to its human component: different people may execute the same interaction in slightly different ways, and positive/negative responses may be manifested in different ways among different participants, etc. These perceptual uncertainties directly affect state identification and, in turn, policy learning [2]. Using hand-picked features or hard-coding every interaction, therefore, is inefficient, if not impossible.

A TR-LfD uses deep convolutional neural networks (CNN) and the temporal information of observed events to learn state features and employs an ITBN-based model to

robustly learn the task's policy (i.e. the state-action pairs). These components are integrated using a layered architecture. A diagram showing the TR-LfD is shown in Fig. 5(b).

A. Demonstration Data Processing

A number of pre-processing steps on the demonstration data are required for policy learning through TR-LfD. A demonstration set for the HRI task to be learned needs to include start- and end-times of all atomic actions in the task. Here atomic actions refer to meaningful events that constitute the entire task. For example, in the contect of RMI, every action and response of the robot and the human is an example of an atomic action. Segmenting a demonstration into a set of atomic actions performed by the robot and identifying their start-end times are relatively trivial steps when the data is collected through tele-operation. Identifying the exact start-end times for responses/actions performed by humans however, is non-trivial and requires special application-specific algorithms. Similarly, when demonstrations are not collected through tele-operation, autonomous segmentation algorithms can be applied to extract atomic actions and their start-end times. For example, the research in [21] and [22] present segmentation techniques for low- and high-level LfD tasks, respectively.

B. Spatial Reasoning Layer (SRL)

The SRL layer of the TR-LfD consists of a set of multi-class classifiers that learn to classify video frames into three perceptual classes: actions performed by robots (ROBOT), responses/events triggered by humans (HUMAN), and events/action that are neither triggered by humans nor performed by the robot (NULL). The atomic actions that constitute the HRI task therefore fall under one of these three classes.

We built on our previous work [23] to propose a convolutional neural network (CNN)-based SRL. The proposed SRL has multiple independent CNNs, each dedicated to process one input modality, e.g. audio, RGB frames, optical flow, depth, etc. Independently operating CNNs allows the SRL to learn relevant features from each modality without the potential of over-fitting to one specific input source. Fig. 3 shows an example SRL with two CNN architectures designed to process spectrogram (generated from audio data) and optical flow inputs. Each CNN architecture consists of standard convolutional layers, a long-short term memory (LSTM) layer and a fully connected (FC) layer. The LSTM layer helps to learn patterns that may be present in the input and that can improve the classification performance. State-of-the-art parameter tuning techniques can be used to choose the values of different parameters (such as filter size, stride, number of filter, the number of output classes, etc), depending on the type of input being used. The CNNs in Fig. 3 show standard choices for different parameter values.

The CNNs in a SRL use a sliding window approach to process the input frames. In this approach, as shown in Fig. 4, the window size indicates the number of frames included in a window, and the frame stride is the number of frames skipped between adjacent windows. The label for each window is determined based on the ITRs it may have with any of the atomic actions observed so far in the interaction. Equation 1 is used to calculate the temporal distance between a given window and other atomic actions present in the task, to identify the presence of ITRs. A window is defined to belong to a class a *during*, *overlaps*, *starts*, *finishes* or *equals* ITR existed between the window and an atomic actions in that class (Fig. 4). The HUMAN class is given priority in this process when a window can belong to more than one of the classes.

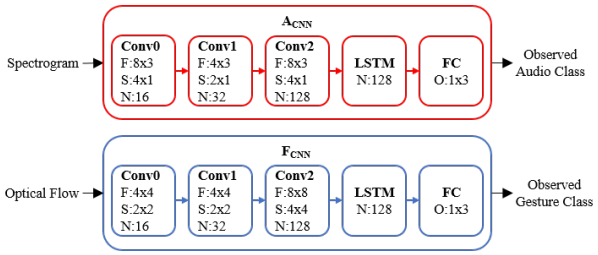


Fig. 3. CNN-based SRL to process image and spectrogram (from audio) data. The size of the filters (F), stride (S), number of filters (N) and output size (O) are design parameters defined based on the type of inputs. Some standard parameter choices are shown here.

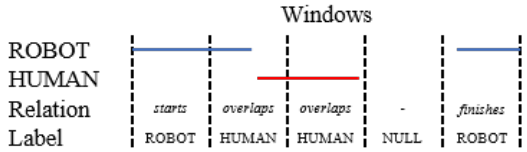


Fig. 4. Labels are assigned to each window depending on the ITRs that exist between them and the different atomic actions (red and blue in the timeline) belonging to a class.

C. Temporal Reasoning Layer (TRL)

During training, the TRL of the TR-LfD learns and encodes the underlying temporal dynamics among the different atomic actions, along with their approximate start-end times. During policy execution, the TRL leverages the learned temporal structure to analyze the observations (i.e. the class label) determined by the SRL and, when needed, select the action that will be executed next. TRL triggers actions under one of the two conditions: a pre-learned waiting period has been elapsed or a change of state has been detected in the environment.

The core of the TRL is an ITBN reasoning model (ITBN-RM). The ITBN-RM expands the capabilities of a standard ITBN by capturing the waiting period between two events and maintaining an open list to facilitate the inference process. In this context, an open list refers to a list of states that can be reached from the current state. The ITBN-RM is formally defined as

$$\text{ITBN} - \text{RM}, G = \langle S_n, T_n, S_t \rangle \tag{2}$$

Where S_n is the set of starting nodes and T_n is the set of terminal nodes. Meanwhile, S_t is the current state of the task

Algorithm 1 Policy Selection in ITBN-RM

Input: obs_{srl}, W_s, W_e
Output: a_{t+1}
initialize: $O_a \leftarrow \emptyset$, $T_a \leftarrow \emptyset$, $a_{t-1} \leftarrow \emptyset$
1: **if** $a_{t-1} = \emptyset$ **then**
2: $a_{t+1} \leftarrow S_n$
3: **else**
4: $obs \leftarrow$ processObservation(obs_{srl})
5: **if** $obs =$ NULL **and** $w = 0$ **then**
6: $obs \leftarrow$ ROBOT
7: **end if**
8: $a_{t+1}, w \leftarrow$ inferNext(obs, O_a, T_a)
9: **end if**
10: **if** $a_{t-1} \neq a_{t+1}$ **then**
11: $O_a =$ updateOpenList(a_{t+1})
12: $T_a \leftarrow T_a \cup (a_{t+1}, W_s, W_e)$
13: $a_{t-1} \leftarrow a_{t+1}$
14: **else**
15: updateEndTime(T_a, a_{t-1}, W_e)
16: $w \leftarrow w - 1$
17: **end if**
18: **return** a_{t+1}

and is defined as

$$S_t = \langle O_a, T_a, a_{t-1}, w \rangle \tag{3}$$

where O_a is an open list of atomic actions that can happen in the remainder of the interaction, T_a are the times at which different atomic actions have taken place, a_{t-1} is the last atomic action that was observed in the interaction, and w is the time (in seconds) that the model will wait before performing inference to select the next action to execute. The graphical structure and parameters of the ITBN-RM are learned from the temporal information of different atomic actions available in the demonstration set. The learning mechanisms for ITBN have been reported in Section III. As a part of the structure learning phase, a set of nodes labeled 'ROBOT observation' are attached to all non-terminal atomic actions related to the robot agent and another set of nodes labeled 'HUMAN observation' are attached to actions/responses related to the human participant. This labeling simplifies policy selection as discussed below and will be further explained in Section V.

During execution, the ITBN-RM is used to select the action that will be performed next in the intervention. Algorithm 1 outlines this policy selection process. This algorithm takes as input the observations generated by the SRL (obs_{srl}) and the start-times (W_s) and end-times (W_e) of the input window that generated those observations. The value of the observation nodes (ROBOT observation/HUMAN observation) for the ITBN-RM (obs) is decided based on obs_{srl} (line 4). The ITBN-RM leverages the temporal information contained in S_t to analyze the values of obs and perform policy selection (line 8). The event times included in T_a are used to calculate the interval temporal relations between past events and an event that has been detected by the SRL. The temporal relation, along with O_a, are then used in the Bayesian inference process of the ITBN-RM. The ITBN-RM also learned the duration of the delays that may exist between different atomic actions from the demonstration set. This information is used during the inference process to decide when to execute the next action in an automated intervention (line 5). To select the next action, the ITBN-RM triggers the ROBOT observation node (line 6) and then infers the appropriate action to execute based on the current state of the intervention (line 10). The ITBN-RM is implemented using the Python package for graphical models, pgmpy [24].

V. Experiments

A. Evaluation Domain

We evaluated the performance of the TR-LfD in learning an applied behavior analysis (ABA)-style RMI to teach a basic social skill (of responding to a greeting in a socially acceptable manner). ABA, a proven methodology for designing behavioral intervention to teach elemental skills to children with autism, suggests a very structured interaction between a child and the teacher which is advantageous from an LfD perspective. Our previous work evaluated the clinically-oriented effectiveness of this particular intervention to teach this social skill to children with autism through a tele-operated robot [25]. In this work, we evaluate the performance of the proposed TR-LfD in learning this intervention from video observations and reproduce it autonomously with different participants.

During the intervention a teacher and a child learner go through a series of structured interactions with the purpose of teaching the child how to respond to a greeting in a socially acceptable manner. The intervention begins with the teacher delivering a discriminative stimuli (SD) where the teacher greets the child by saying "hello" and waving at him/her. The child may respond (RESPONSE) verbally and/or wave his/her hand. If the child does not provide an appropriate response, the teacher proceeds by delivering a prompt (PROMPT), which shows the child how to respond in a socially acceptable manner, e.g. "John, say hi to me". If the intervention is failing to be productive, the teacher can decide to abort the session (ABORT). If the child provides an appropriate response, the teacher concludes the session by giving a verbal reward (REWARD) to the child, such as "Great job!". In the context of the primitive interaction structure discussed in Section IV, a: SD, r: RESPONSE, R: REWARD, F: PROMPT, and e: ABORT.

B. Demonstration Data and Pre-processing

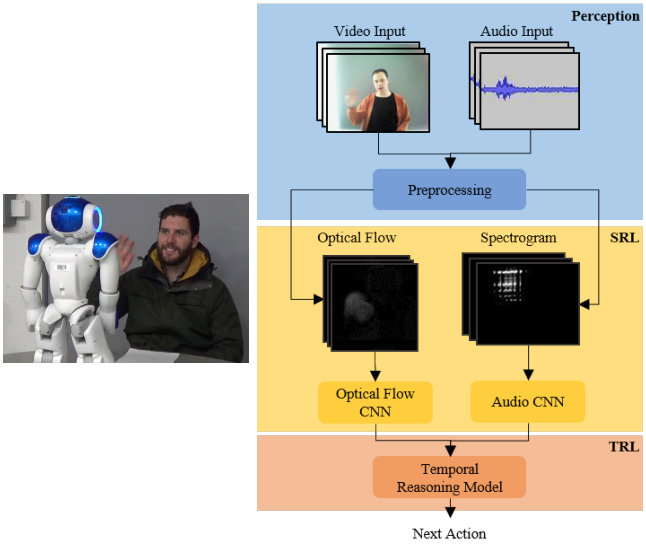
An IRB-approved user study was organized to collect demonstration data. In the user study, a NAO humanoid robot was tele-operated to deliver the ABA-based intervention described above. The setup used during the data collection sessions can be observed in Fig. 5(a). The robot performed the following four actions in the role of a therapist: SD, PROMPT, REWARD and ABORT. Six college students (4 male, 2 female) without autism participated in the study. Before starting the study, participants were made aware

that the robot was being tele-operated. Each participant was requested to complete a set of 18 interactions with the tele-operated robot emulating a compliant or non-compliant state. In 12 of the sessions the participants emulated a compliant state by providing an appropriate response to the robot, thus ending the session successfully and receiving a verbal reward from the robot. In the remaining 6 interventions, participants emulated a non-compliant state, meaning they did not provide a valid response to the robot. The participant responses consisted of different combinations of gaze, gestures, and audio, as defined below:

- Gaze: maintaining eye contact with the robot (Responses consisting of only gaze were not considered valid in this user study.)
- Gesture: responding to the robot with a waving gesture.
- Audio: acknowledging the robot with a verbal response, e.g. "hello".

The different combinations of valid responses were kept balanced by indicating the participants which response they should use when emulating a compliant state. The demonstration data also included the temporal information (start and end times) of the atomic actions. The temporal data for the SD, PROMPT, REWARD and ABORT actions was available from the tele-operation logs of each session. Meanwhile, the timing information for the participant's responses were hand-labeled by the first author of this work. The labeling process consisted of analyzing the video and audio that preceded a REWARD action, to find the start and end times of a response. Timing information for auditory responses was initially obtained by processing the data set with speech recognition software. These times were identified manually in instances where the verbal response was not recognized by the software. The collected data set had a total of 189 demonstrations. All the sessions included the SD action, but only 133, 112, and 77 included the PROMPT, REWARD and ABORT actions, respectively. From the successful interactions, 74 contained gestural responses and 75 had auditory responses. An evaluation data set was created by randomly selecting 25% of the demonstration videos.

All video and audio data were recorded using the camera and microphones available on the NAO robot. The image feed is recorded with the robot's main camera, which provides 640×480 images at a rate of 15 frames per second. These images are cropped to be 299×299 pixels in size and are centered on the participant's face using a Haar Cascade classifier trained on human faces. Frames in which a face cannot be detected are cropped using the center of the original image as a reference point. The resulting images are then resized to 64×64 and converted to gray-scale. Finally, an optical flow image for each frame is generated using the change detection method described in [26]. The frames of the video are then collected into an array F. Audio data is pre-processed using a combination of spectral subtraction and FIR filters to reduce the audio signal's background noise. The smoothed data is subsequently converted to a Mel-Spectrogram in order to provide a two-dimensional



(a) (b)

Fig. 5. (a) Physcial setup for user studies, (b) TR-LfD framework for learning social greeting intervention.

representation of the data [27]. Finally, the resulting image is split into an array of frames (A) equal in length to the number of frames in F. Each of the frames in A has dimensions 128×8 and contains columns from the previous frame in its first two columns and columns from the next frame in its last three columns, to generate a better representation of the entire audio signal and capture relevant patterns.

C. Spatial Reasoning Layer (SRL)

We used two independent CNN to define the SRL, one to process F and the other to process A. The two CNNs shown in Fig. 3 have been used to define the SRL and are discussed below with greater details.

Audio CNN: The input of the model for the auditory network (A_{CNN}) is A, the visual representation of the recorded audio. For the training process, a grid search approach was used to find the window size and frame stride that maximized the number of windows that contain an entire audio response from across the entire training dataset, without impacting the training time of the CNN. As a result, the window size parameter was set to 20 frames and the frame stride had a value of 7. To simplify the model, the SD, PROMPT, REWARD, and ABORT actions were grouped into the ROBOT class. The number of training examples of each class was balanced by omitting excess windows belonging to the more common classes (ROBOT and NULL). The omitted windows were randomly selected.

Optical Flow CNN: The optical flow CNN (F_{CNN}) uses F as its input. During training, the window size and frame stride parameters were set to 45 and 20 frames respectively. As in the case of A_{CNN}, the values of these parameters were selected using a grid search approach. The ROBOT class captured the ambient movement caused by the waving motion of the robot when performing the SD and PROMPT

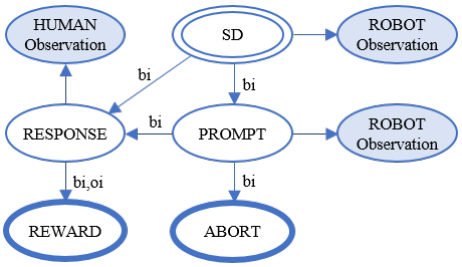


Fig. 6. ITBN-RM learned from demonstrations. The start node of the model is marked with a double line, terminal nodes are marked with a thick line and observation nodes are shaded.

actions. Meanwhile, the REWARD and ABORT actions were excluded from this CNN as they did not trigger any motion from either of the participants in the intervention. The same approach used in the training of A_{CNN} for the balancing of training examples was used for F_{CNN}.

D. Temporal Reasoning Layer (TRL)

Fig. 5(b) shows the entire TR-LfD framework for the experiment. The ITBN-RM learned from the demonstration data is shown in Fig. 6. The interval temporal relations in this model can be read as follows: dependent node, followed by the temporal relation, and then the reference node. For example, the relationship '*prompt* $\rightarrow_{bi}$ *response*' can be read as "a response follows a prompt". The ROBOT observation nodes are triggered when the A_{CNN} classifies a window as a ROBOT action. Meanwhile, the HUMAN observation node is triggered if either the A_{CNN} or the F_{CNN} classify the window as a HUMAN action. A_{CNN} is given priority in this process because the REWARD and ABORT actions do not trigger the F_{CNN}. To reduce the number of false positives during execution, the ITBN-RM was configured to require two consecutive observations to accept the detection of a HUMAN class.

A brief example of the policy selection process using the ITBN-RM of Fig. 6 is illustrated in Fig. 7. In this example, the first window, w_1 (red frame), consists of the last 20 frames of video corresponding to an SD atomic action. Both A_{CNN} and F_{CNN} classify w_1 as a ROBOT action, triggering the ROBOT observation nodes of the ITBN-RM. In the graphical representation of the ITBN-RM for w_1, the current atomic action is shaded with green, actions that can still happen in the intervention are white and the inferred atomic action for the current window is marked with a red line. After 7 more frames are received from the robot, a second window, w_2 (green frame), is processed. In this case, the F_{CNN} classifies the window as a HUMAN action, so the HUMAN observation node is triggered in the ITBN-RM. Using this observation, the ITBN-RM infers that the window belongs to a RESPONSE action and updates the state of the model. Now, atomic actions that have already concluded are shaded with yellow and atomic actions that cannot happen in the session are shaded with gray. Afterwards, a set of HUMAN action windows are received and processed by the ITBN-RM, for simplicity these are not included in the figure. Finally, 3 seconds after the RESPONSE action ends, window w_n (blue frame) is processed. In this window, the ITBN-RM performs inference and selects REWARD as the next atomic action to execute, thus ending the interaction.

VI. Results

A. Simulation

The learning capabilities of the framework were tested in simulation using the training and evaluation data sets. The window size and frame stride parameters of the CNN models were set to 20 and 7 frames, respectively, using the values used during training as reference.

The evaluation dataset was used to execute two different simulated experiments on the framework. The first one, aimed to evaluate the individual performance of each CNN when used to classify the windows of the evaluation set. The results, which are shown on Fig. 8, confirm that both models of the SRL were able to generalize and use the learned features to classify novel windows with an accuracy of over 92%. Moreover, even though the classification rate of A_{CNN} for the HUMAN class was only 83.72%, a more thorough inspection revealed that all the false negatives were late detections. This means that the first HUMAN action window of an event was not classified correctly, but the subsequent windows were. Therefore, during the delivery of an intervention, the event would be recognized with a negligible delay or, in most cases, no delay at all.

In the second experiment, the TRL was used to infer which atomic action was taking place in each window given the observations provided by the SRL and the current state of the ITBN-RM. Since in this use case the SD atomic action is the only start node, the first atomic action by the robot recognized by the SRL was assumed to be the SD. First, the whole framework was evaluated on the 139 training demonstrations, achieving a performance of 98.56% at the session level (two failed sessions). At the atomic action level, the model achieved a perfect performance for the SD, PROMPT, and ABORT atomic actions and a performance of 97.59% on the RESPONSE and REWARD atomic actions. Then, the experiment was repeated using the evaluation set, with a performance of 97.48% (2 of 50 sessions failed). Once again, the model was able to achieve a performance of 100% on the SD, PROMPT, and ABORT atomic actions. The two failed sessions were caused by false positives reported by the A_{CNN}. From the results of this experiment it is important to highlight that, even though the CNNs misclassified a total of 184 windows on the first experiment described above, the TRL was able to use its knowledge of the temporal dynamics to reduce the number of failed sessions to only two.

B. Experiments with Human Participants

To fully evaluate the learning capabilities of the framework, a new IRB-approved user study was conducted. The setup and structure of the intervention were the same described in section V-A. In this study, however, the behavioral intervention was delivered autonomously by the robot. Six college students (5 male, 1 female) without ASD participated in the study and were made aware that the robot was acting

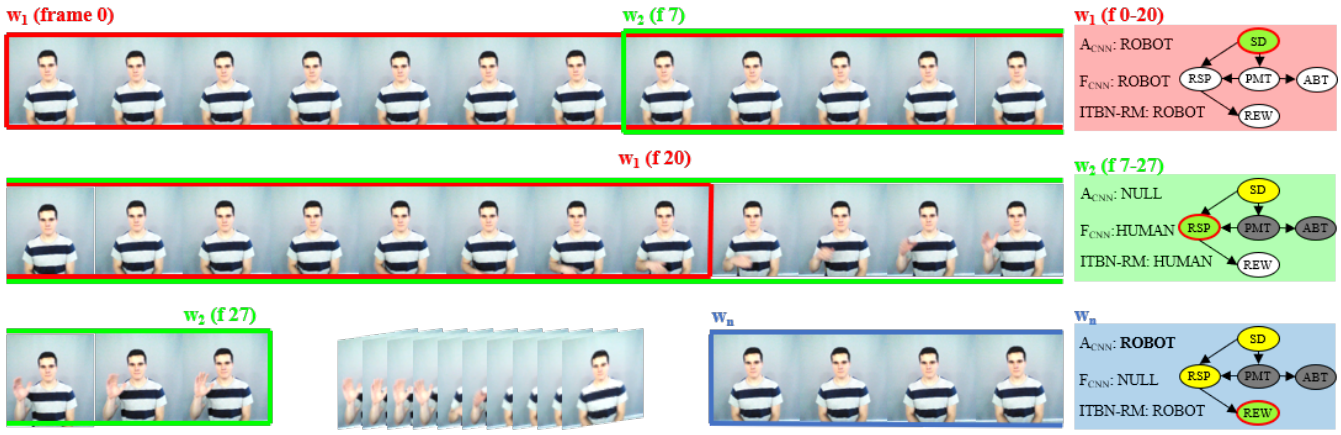


Fig. 7. Graphical description of the policy selection process executed by the ITBN-RM. In the graphical representation of the state the current event is marked with green, possible future events are white, events that can no longer take place are gray, events that have already concluded are yellow and the event inferred for the current window is marked with a red line.

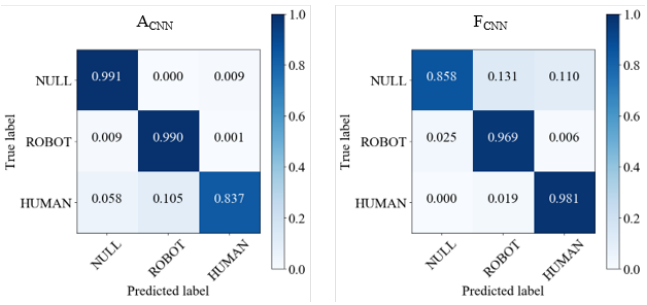


Fig. 8. CNN performances on validation data set.

TABLE I
ACCURACY ON AUTOMATED INTERVENTIONS

Responses				
Gaze	**Gestural**	**Auditory**	**TR-LfD**	**DR-LfD**
No	No	No	94.4%	**95.8%**
No	No	Yes	**100%**	75.0%
No	Yes	No	**91.7%**	25.0%
No	Yes	Yes	**83.3%**	68.8%
Yes	No	No	**94.4%**	87.5%
Yes	No	Yes	**91.7%**	81.3%
Yes	Yes	No	**100%**	6.3%
Yes	Yes	Yes	**100%**	37.5%
Total			**94.4%**	**67.8%**

autonomously. None of the participants had taken part of the data collection study. Each participant completed a set of 18 interactions that included different combinations of audio and gesture response, as well as instances in which no response was provided.

This experiment was executed using the policy selection routine described in Algorithm 1. An automated intervention started with the robot executing the SD action (line 4), as it is the only start action in the ITBN-RM. Then the robot would act according to the participants reaction. If no appropriate response was given, the robot would wait before selecting the next action, following the behavior observed in the training set. Similarly, if a valid response was received, the robot waited for a short period before triggering the next action. These wait periods were learned from the temporal information of the data set. The interventions continued until a terminal action (REWARD or ABORT) was executed by the robot.

In this experiment, 84.26% of the automated interventions were successful (91 successful, 17 failures). An intervention was considered successful if the model allowed the robot to react in accordance to the actions of the human participant and the state of the intervention until the REWARD or ABORT actions were executed. A total of 11 failures were caused by the misclassification of a ROBOT action by the A_{CNN}. Table I shows the performance of the framework at selecting the correct action after different combinations of human responses. The accuracy in this table represents the percentage of responses that were followed by the correct action by the robot. In addition, this table also includes the results obtained by our previous Deep Reinforcement LfD (DR-LfD) approach [23] on the same ABA intervention. A demonstration video of the performance of the system can be found at https://goo.gl/veo9Ht.

A questionnaire was completed by the participants of the evaluation user study. The questions asked them to grade the performance of the automated robot and their overall experience using a Likert scale with 5 meaning "strongly agree" and 1 "strongly disagree". The scores reflect that the robot learned to react correctly according to the actions of the participants (4.5 ± 0.5) and that, even though the automated intervention did not feel natural (3.5 ± 1.0), interacting with the robot was easy (4.8 ± 0.4) and enjoyable (4.3 ± 0.5). Nevertheless, we believe that the low score associated with the naturalness of the interaction is a product of the highly structured nature of the intervention.

VII. DISCUSSION

The performance of the TR-LfD framework on the evaluation set shows that the proposed approach is capable of

learning sequential HRIs from a relatively smaller number of training demonstrations. This is in part because the use of a TRL allows the simplification of the models in the SRL. The results illustrate the capacity of the TR-LfD to leverage the underlying temporal dynamics of the task to perform temporal reasoning, even in cases when the SRL provided incorrect observation values. This can be observed by comparing the results of the two simulated experiments. In the first of these experiments, close to 5% of the windows were misclassified. However, by using the learned temporal rules and constraints of the interaction, the TRL can minimize the effect of the misclassified windows, achieving a performance of 98%. Lastly, it is relevant to point out that the proposed approach outperforms the accuracy of the DR-LfD by 26.6%. This difference further confirms the advantages of leveraging temporal reasoning in a LfD framework.

VIII. Conclusion

This paper presents the TR-LfD, a novel Temporal-Reasoning-based LfD framework that has been proven capable of learning a sequential human-robot interaction from demonstrations. The framework relies on an SRL that extracts the discriminative features of the different states of a task and a TRL that derives and leverages the temporal dynamics of the task. The framework was evaluated with a use case consisting of a robot-mediated intervention. The results confirm that the framework is capable of learning and automating sequential human-robot interactions, even when trained with a small number of demonstrations. These results suggest that temporal reasoning will be key when deploying automated robotic agents on applications that require human-robot interactions.

Future work will explore the use of more complex temporal reasoning approaches, capable of modeling recursive events in an interaction. Additionally, the possibility of implementing video segmenting techniques to replace the hand-labeling stage of the data collection process will be researched. These improvements will allow the framework to be fully automated and more generalizable.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] S. Chernova and A. L. Thomaz, "Robot learning from human teachers," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 8, no. 3, pp. 1–121, 2014.

[3] S. R. Ahmadzadeh, R. Kaushik, and S. Chernova, "Trajectory learning from demonstration with canal surfaces: A parameter-free approach," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 544–549.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer handbook of robotics*. Springer, 2008, pp. 1371–1394.

[5] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, "Learning grounded finite-state representations from unstructured demonstrations," *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015.

[6] C. Paxton, F. Jonathan, M. Kobilarov, and G. D. Hager, "Do what i want, not what i did: Imitation of skills by planning sequences of actions," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 3778–3785.

[7] R. Cubek, W. Ertel, and G. Palm, "High-level learning from demonstration with conceptual spaces and subspace clustering," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2592–2597.

[8] N. Koenig and M. J. Matarić, "Robot life-long task learning from human demonstrations: a bayesian approach," *Autonomous Robots*, vol. 41, no. 5, pp. 1173–1188, 2017.

[9] K. Bullard, B. Akgun, S. Chernova, and A. L. Thomaz, "Grounding action parameters from demonstration," in *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on*. IEEE, 2016, pp. 253–260.

[10] M. Begum, R. W. Serna, and H. A. Yanco, "Are robots ready to deliver autism interventions? a comprehensive review," *International Journal of Social Robotics*, vol. 8, no. 2, pp. 157–181, 2016.

[11] M. Clark-Turner and M. Begum, "Deep reinforcement learning of abstract reasoning from demonstrations," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2018, pp. 160–168.

[12] J. F. Allen and G. Ferguson, "Actions and events in interval temporal logic," *Journal of logic and computation*, vol. 4, no. 5, pp. 531–579, 1994.

[13] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Robot gains social intelligence through multimodal deep reinforcement learning," in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 745–751.

[14] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[15] S. Niekum, S. Chitta, A. G. Barto, B. Marthi, and S. Osentoski, "Incremental semantically grounded learning from demonstration." in *Robotics: Science and Systems*, vol. 9. Berlin, Germany, 2013.

[16] Y. Zhang, Y. Zhang, E. Swears, N. Larios, Z. Wang, and Q. Ji, "Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 10, pp. 2468–2483, 2013.

[17] C. P. d. Campos and Q. Ji, "Efficient structure learning of bayesian networks using constraints," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 663–689, 2011.

[18] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[19] D. David, S. Matu, and O. A. David, "Robot-based psychotherapy:concepts development, state of the art, and new directions," *International Journal of Cognitive Therapy*, vol. 7, no. 2, p. 192210, 2014.

[20] M. A. Goodrich, M. Colton, B. Brinton, M. Fujiki, J. A. Atherton, D. Ricks, M. H. Maxfield, and A. Acerson, "Incorporating a robot into an autism therapy team," *IEEE Intelligent Systems Magazine*, vol. 27, no. 2, pp. 52–59, 2012.

[21] N. Figueroa and A. Billard, "Learning complex manipulation tasks from heterogeneous and unstructured demonstrations," in *IROS Workshop on Synergies between Learning and Interaction*, 2017.

[22] A. Murali, A. Garg, S. Krishnan, F. T. Pokorny, P. Abbeel, T. Darrell, and K. Goldberg, "Tsc-dl: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4150–4157.

[23] M. Clark-Turner and M. Begum, "Deep reinforcement learning of abstract reasoning from demonstration," in *HRI '18: 2018 ACM/IEEE International Conference on Human-Robot Interaction*, March 2018.

[24] A. Ankan and A. Panda, "pgmpy: Probabilistic graphical models using python," 2015.

[25] M. Begum, R. W. Serna, D. Kontak, J. Allspaw, J. Kuczynski, H. A. Yanco, and J. Suarez, "Measuring the efficacy of robots in autism therapy: How informative are standard HRI metrics'," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. ACM, 2015, pp. 335–342.

[26] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[27] L.-C. Yang, S.-Y. Chou, J.-Y. Liu, Y.-H. Yang, and Y.-A. Chen, "Revisiting the problem of audio-based hit song prediction using convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 621–625.