Understanding Temporal Relations from Video: A Pathway to Learning Sequential Tasks from Visual Demonstrations

Madison Clark-Turner and Momotaz Begum

Abstract— **Sequential tasks, such as many activities of daily living, typically have innate temporal structures. Understanding these temporal structures can greatly benefit the learning of these tasks from visual demonstrations. Learning temporal relations from un-engineered video however is a challenging frontier that is largely under explored both in computer vision and vision-based learning from demonstration (LfD) research. This paper proposes *Deep Interval Temporal Relationship Learner (D-ITR-L)*, a pipeline that extracts rich temporal relations among visual features in the video. D-ITR-L acts as a wrapper building on the spatial features learned by any standard convolutional neural network (CNN)-based video inference architecture. We use a Graph Convolutional Network (GCN) in concert with D-ITR-L to infer discriminatory temporal relations from video. We evaluate the effectiveness of D-ITR-L learned features in two contexts: vision-based policy learning of a block-stacking task by a robot and activity recognition from two benchmark video datasets namely furniture construction [14], [30] and recipe following [20]. Our code is available at [5].**

I. INTRODUCTION

Temporal relationships that exist among spatial visual features, hereafter referred to as temporal features, are key to understanding sequential tasks. In the context of a sequential task, such as making tea, temporal features can describe the order of events (water is boiled *before* it is poured), when and how spatial features overlap (the teabag and spoon were visible *simultaneously*), and repeated spatial patterns (sugar was added to the cup *twice*). Temporal features are useful in settings where spatial features are no longer discriminatory due to their abundance in the environment. By relating how spatial feature expression evolves over time, instead of just spatial feature presence, we build an information-rich representation of the world that directly benefits inference from visual demonstrations. In this paper, we are interested in learning temporal features from videos and leveraging them to conduct inference on sequential tasks.

Convolutional neural networks (CNN), because of their unparalleled success in identifying visual features from images and videos [10], are the standard for autonomous visual feature identification. CNNs, therefore, lie at the heart of many end-to-end frameworks for vision-based activity recognition [32], [35] and LfD policy learning [24], [25]. However, contemporary CNN architectures are ill-equipped to robustly recognize temporal features [38], [40]. Deep learning approaches such as recurrent neural networks (RNN) and 3D convolutional filters, known for their temporal learning ability, describe temporal relationships among visual features in a duration-specific manner [40], [42]. Subsequently, these approaches can scale poorly to long videos and have difficulty generalizing to novel data [17]. This paper focuses on two issues that make end-to-end learning of temporal features from videos challenging:

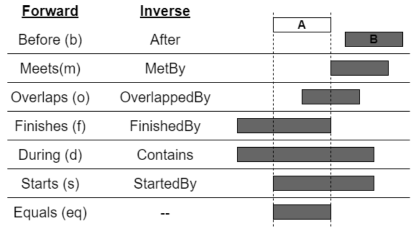


Fig. 1: The 13 Interval Temporal Relationships

1) *Variations in Duration*: The onset of a spatial feature in a video and the duration over which that feature is expressed are rarely consistent between examples. For instance, the action ‘pouring from a pitcher’ may occur over 3 seconds or 30, depending on the persons demonstrating the task. Most CNN-based architectures are duration-specific and use multiple representations to capture this activity (discussed in Section II). If the temporal information were, instead, learned in a duration invariant manner, the same representation would capture both observations. For example, if “X” is a set of spatial features that shows the start of the ‘pouring’ event and “Y” is a set of spatial features corresponding to the end of the same event, an abstract description such as “X” *before* “Y” can characterize both observations (3s and 30s), without mentioning their duration. Allen’s Interval Algebra describes 13 such abstract interval temporal relationships (ITRs) (Fig. 1). Use of abstract temporal relationships among spatial features is a great way to achieve a duration invariant representation of time. This approach has yet to be explored in CNN-based video inference applications.

2) *Video-scale features*: Videos of sequential tasks can last upwards of several seconds throughout which significant, and sometimes cyclical, features may be scattered. Boiling water for tea occurs in several steps: adding water to the kettle, boiling the water, and pouring the water into the teapot. Ensuring that each event is expressed in order confirms that the task was completed correctly. Similarly, the same activity could repeat a specific number of times, such as adding spoonfuls of sugar to a teacup. Recognizing these examples requires parsing the entire video as opposed to a part of it. Many CNN-based methods claim to capture video-scale features, but these approaches do so at the cost of fidelity [42] or computational resources [41]. The natural inclination to isolate these patterns via segmentation of the video is naive, requiring substantial annotation of the training

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

dataset or expert domain knowledge [3]. Instead, video-scale features are best characterized through the use of a graphical structures that directly relate distant features.

The state-of-the-art in both vision-based LfD and computer vision research has lacks a holistic solution to these issues. Contemporary video inference architectures were designed with an innate spatial-bias [37] and have been evaluated on benchmarks that reflect this bias [2]. Subsequently, the importance of temporal features and the methods for extrapolating them have not yet been adequately investigated. To that end, we propose *Deep Interval Temporal Relationship Learner* (D-ITR-L), a framework for identifying the temporal features present in video. D-ITR-L is a wrapper that uses CNN-learned spatial features in concert with Allen's Interval Algebra [1] to capture and describe the temporal features present in a video. Our approach combines abstract temporal relationships with a graphical structure to capture video-scale features in a duration invariant manner. The Graph Convolution Network (GCN) is a natural approach to modelling this data and we use it to make high-level inferences about video observations. In this paper we show that D-ITR-L is a highly effective approach to modelling human-led demonstrations of sequential tasks that can lead to improvements in activity recognition and policy learning.

II. RELATED WORK

We discuss how vision-based LfD policy learners and CNN-based computer vision methods fail to address the challenges associated with learning temporal features: *variations in duration* and *video-scale features*.

A. Policy Learning

Contemporary vision-based LfD architectures leverage the feature learning properties of CNNs in their design. Many of these models are designed for low-level control using single frames or short video clips [23], [24], [25], [36]. They do not encounter *variations in duration* or issues representing *video-scale features*. High-level policy learners using vision typically rely on simplifying assumptions [7], [13] or expert knowledge [11], [29] when implementing perception. The exception is our earlier work [6] which used a CNN to teach a robot a behavioral therapy from full-length video.

B. Computer Vision

Understanding temporal features is a goal of video-recognition research. Cao *et al.* [2] discuss the spatial focus of older video datasets and the interest in representing temporal features in newer datasets. However, real-life sequential tasks have a greater spatial and short-term focus than the most temporally-focused datasets by computer vision standards ([12], [22]).

Integrated	**Interleaved**	**Separate-CNN**
MML [18]	TPN [38]	TCN [19]
SlowFast [9]	TSM [21]	Multiscale TRN [41]
ECO [42]	STM [15]	**Separate-RNN**
I3D [4]	TrajectoryNet [16]	ConvGRU [8]
2-stream [27]	R(2+1)D [31]	CNN-LSTM [34]

TABLE I: Popular CNN Models for Video Inference.

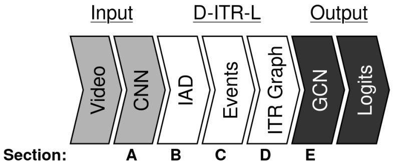


Fig. 2: The D-ITR-L Pipeline. References are for Section III.

CNN-based approaches to representing time generally fall into one of three categories (Table I): *integrated* models learn spatio-temporal features with 3D-convolutions, *interleaved* models alternate spatial and temporal feature learning, and *separate* methods learn temporal features after spatial features (through either convolutions (CNN) or recurrent models (RNN)). These architectures fail to address the two issues of temporal feature learning. They do not develop duration invariant temporal representations. Integrated, interleaved, and separated-CNN models represent time explicitly in terms of elapsed duration. Separated-RNN methods can potentially generate duration invariant representations by aggregating adjacent frames together if feature expression is constant. But, this is unrealistic and RNNs overfit to noisy data [40], [42]. These methods also fail to generate representations that scale to the length of full videos. Integrated models have high computational demands limiting inference to a handful of frames. These models generally infer over parts of a video and lack the scope to capture distant temporal relationships. Interleaved methods are less demanding but use frame-skipping to constrain the duration of a video which require expensive ensembles to find the best temporal stride at which to sample frames. Separate models are most likely to perform frame-by-frame analysis without an ensemble. However, CNN-based models require unreasonably deep architectures to develop a representation that spans the duration of a full video. Also RNN-based approaches generate temporal features by a sequential aggregation of frames demanding that their temporal representation consider every frame between related spatial features [42], [19].

III. D-ITR-L: Learning from Temporal Features

Deep Interval Temporal Relationship Learner (D-ITR-L) is a pipeline (Fig. 2) that leverages the spatial features identified by a CNN backbone model to develop temporal features (in the format of an ITR graph, a novel graphical representation of connectivity among events which will be discussed in Section III-D) which are used as the basis for training a GCN for state estimation. The backbone and graphical inference methods of D-ITR-L are user-defined and interchangable with contemporary alternatives. Our novel contribution is the extraction of temporal features from this data and its presentation (as an ITR graph) and the benefits achieved by this approach.

A. Spatial Feature Extraction

The first step in the D-ITR-L pipeline is identifying a set of informative spatial features from which to build complex temporal relationships. We obtain these spatial features from a pre-trained, user-selected CNN backbone. We assume that the CNN backbone has been fine-tuned to recognize spatial features present in the dataset that D-ITR-L is expected to operate upon. The method used to prepare backbone CNN models in this work is discussed in Section IV-B.2.

B. *Formatting Interval Algebra Descriptors*

By leveraging the spatial features extracted from a video we can determine when a specific feature is expressed in time. This is accomplished through a novel data structure we term the Interval Algebra Descriptor (IAD). The output of a CNN backbone when operating on video data is a four-dimensional tensor ($F \times T \times H \times W$) that captures the relative expression of the learned features in the space (height (H) and width (W)) and time (T) of a video. This information is used to generate our two-dimensional representation ($F \times T$), the IAD, by collapsing the spatial dimensions of the tensor using the maximum operation. Fig. 3a shows an example IAD developed from a video lasting 195 frames.

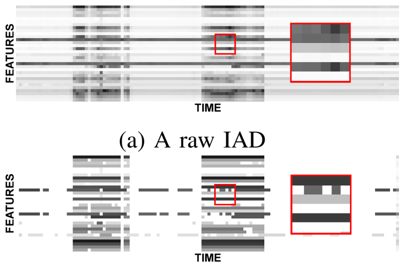
Expression of features in an IAD is currently continuous. To explicitly determine when a spatial feature is being expressed or not we apply a threshold to each feature in the IAD (Φ_f), with values above the Φ_f indicating moments when a feature is actively expressed. The value of Φ_f is selected for each feature ($f \in F$) in the IAD. We use the mean value of that feature's expression across the entire dataset (D) as the criteria for this threshold ($\Phi_f = \frac{1}{|D||T|} \sum_{d \in D} \sum_{t \in T} IAD_{d,t,f}$). This requires a single pass over the training dataset to identify this value and a subsequent pass to threshold the data. We investigated more complex threshold values by varying the mean according to the standard deviation of the feature expression, but none were as effective as the one described. Alternative threshold values exist and identifying if a better value exists is a potential future research direction.

C. *Event Detection*

From the IAD we can perform event detection to determine the explicit start and stop times when a spatial feature is actively expressed. We define each event (the regions above Φ_f) with a four-tuple $< t_s, t_e, f_i, f_{mx} >$ denoting the timestamps at which the event started (t_s) and ended (t_e), and a description of the content of the event using the feature label (f_i) and the maximum expression of that feature across the event (f_{mx}). Fig. 3b depicts the events present in the IAD shown in Fig. 3a. The intensity of the shaded regions matches the value of f_{mx}.

D. *Interval Temporal Relationship Identification*

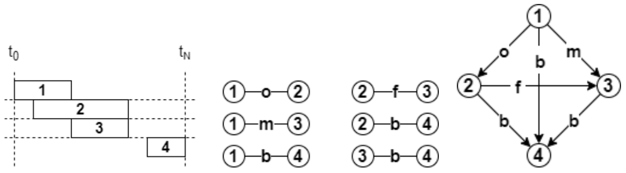
The events identified in Fig. 3b are the basis for recognizing the presence of ITRs among the events in the input video. To understand this process, Fig. 4a shows four example events identified from a thresholded IAD of a video. We investigate only *forward* ITRs (listed in Fig. 1) to avoid redundant calculations. We adopted the approach proposed in our earlier work [3] for rapidly identifying forward ITRs: sorting the events in ascending order of t_s and t_e, and then iterating through the events in a pairwise manner to find the ITR that relates each pair of events. Fig. 4b provides a demonstration of this, where we have related event 1 to event 2 by an "overlaps" ITR, event 1 and event 3 by a "meets" ITR, so on and so forth. This list of ITRs is subsequently assembled into an ITR graph. Events become the nodes (labelled with f_i, and weighted by f_{mx}) and the ITRs become the edges. Fig. 4c depicts an ITR graph. The ITR graph is the collection of all the temporal features in an input video.



(a) A raw IAD

(b) Events expressed in the IAD

Fig. 3: Event detection using IAD. The raw IAD (a) contains 32 features expressed over 195 frames. Black intensity denotes greater feature expression. The thresholded version of the same IAD (b) explicitly defines when and to what degree events are expressed. We show zoomed regions of each IAD for clarity (in red).



(a) IAD Example (b) ITR List (c) ITR Graph

Fig. 4: Transformation from IAD (a) to a list of ITRs (b) to an ITR Graph (c). ITR denotation matches Fig. 1.

E. *Learning From Temporal Features*

We use a GCN to perform inference using the ITR graph. GCNs have been used to model unprincipled spatial [40] and action relationships [37] in earlier works. ITRs are a principled logic and their use as input to a GCN is a novel attempt. The relational GCN (R-GCN) described in [26] learns the discriminatory relations of a graph whose edge-labels are relationships. This offers a natural integration of our data. Convolutions in a GCN inference extend a representation from a node along edges to neighboring nodes. A sufficiently deep GCN can create a network of ITRs that describe complex dependencies among several temporal features. In the context of Fig. 4c, the combined ITRs that connect events 1, 2, and 3 might be discriminatory compared to other relationships in the graph. The output of a GCN is a vector of values that represent the contents of the video (logits). These logits can be used in concert with any deep learning-based video inference application. In this work they are used for activity recognition (using a softmax layer) and state estimation when learning the policy of sequential tasks

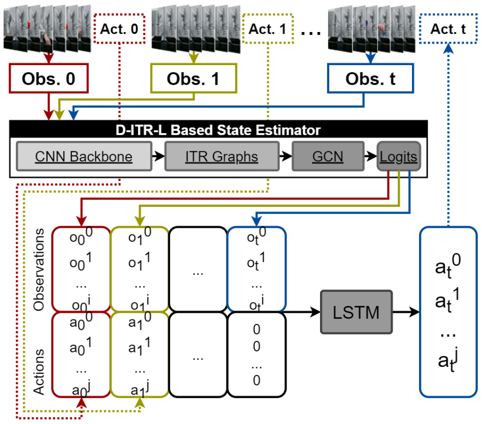

Act. 0
Act. 1
...
Act. t
Obs. 0
Obs. 1
Obs. t
D-ITR-L Based State Estimator
CNN Backbone
ITR Graphs
GCN
Logits
Observations
Actions
o_0^0 o_0^1 ... o_0^i
a_0^0 a_0^1 ... a_0^j
o_1^0 o_1^1 ... o_1^i
a_1^0 a_1^1 ... a_1^j
...
o_t^0 o_t^1 ... o_t^i
0 0 ... 0
LSTM
a_t^0 a_t^1 ... a_t^j

Fig. 5: The D-ITR-L policy learning pipeline.

from visual demonstrations.

Our policy learning architecture operates as a pipeline (Fig. 5). A video-based observation of arbitrary duration taken at time (o_t) is fed as input into D-ITR-L to generate an I length vector of logits (o_t^i where $i \in I$). The length of I is user-defined and should be large enough to capture all of the potential observation states needed to define the policy. These logits are combined with logits generated by observations in previous time steps and a one hot encoding of prior actions (a_t^j; of length J, where $j \in J$). The length of J is defined by the number of actions in the task. Zeros are used to represent the action to be inferred (a_t). The resulting matrix is an estimation of the state (S) and is fed sequentially into an LSTM layer. The LSTM generates values for each of the policy's actions and the action with the highest value is performed in the subsequent time step (a_t).

IV. EXPERIMENTS

We establish that D-ITR-L is a highly effective method for capturing and inferring the temporal features present in video through policy learning and activity recognition tasks.

A. Datasets

The strength of D-ITR-L at learning temporal features is best demonstrated through tasks that capture *variations in duration* and *video-scale features*. Existing benchmark datasets fail to investigate these concerns [2]. To that end, we designed a dataset and re-purpose two other benchmark video datasets to evaluate D-ITR-L. All videos within these datasets are taken at 30fps and down sampled to 10fps.

1) Block Stacking Task: In this task, a human moves colored blocks between two opaque containers while following any of these rules at each step: move no blocks (n); move one red (r), blue(b), or green(g) block; move a blue block followed by a green block (bg) or vice versa (gb); or move two or three red blocks (rr and rrr respectively). These last two are examples of *video-scale features*. The use of opaque containers focuses learning on temporal features, preventing a single frame of the video from fully defining the observation. The goal of the experiment is for a robot to stack colored blocks in an order matching the pattern demonstrated by the human (Fig. 6). The robot selects one action during each phase of the interaction to either stack a single colored block (R, B, or G) or pass (N).

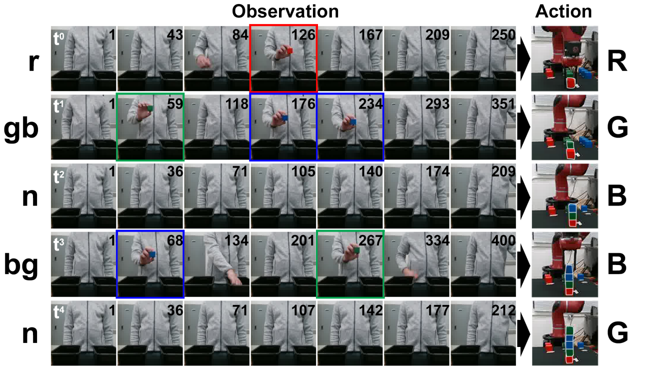


Fig. 6: The Block Stacking Experiment. Frames of observations where a colored block are visible are highlighted with the block color. Frames are sampled uniformly from source video and tagged with the frame number.

Expert demonstrations are collected with a single human demonstrator and a tele-operated Sawyer robot. We recorded ten RGB videos in which the human moved blocks according to the eight observations for a total of 80 videos. We set three videos from each observation aside for evaluation and used the rest for training. To investigate the influence of *variations in duration*, we compared two variants of the dataset: one where the timing of the movement of blocks was **fixed** according to a metronome and another where movements were executed **variably** according to the whims of the user.

For efficiency, multi-step demonstrations were generated in a procedural manner by shuffling one example from each of the eight observations together and choosing the appropriate actions given the observation sequence. Observations depicting no action (n) are used to pad the sequence of observations to match the length of the sequence of actions. A total of 100 traces were generated of which 90 were used for training and the remainder were used for evaluation. Trajectory learning is beyond the scope of this work and it is assumed that the robot knows where the blocks are located and how to manipulate the blocks. The focus of this experiment is to generate a strong representation of the state using the latent temporal information present in the video observations.

2) Activity Recognition: We investigate two activity recognition datasets re-purposed from other computer vision applications namely, action prediction and multiple person tracking. The first is a furniture construction dataset composed of 101 videos capturing actors as they perform the 6 steps to construct and deconstruct a table (Fig. 7(left)) [14], [30]. The second is a recipe following dataset consisting of 53 videos of chefs preparing meals according to 6 possible recipes [20] (each composed from 9 composite actions (Fig. 7(right))). We develop two variations of this second dataset: one focuses on the composite actions and the other on the recipes. Videos among all three datasets are subject to variations in duration and capture video-scale features.

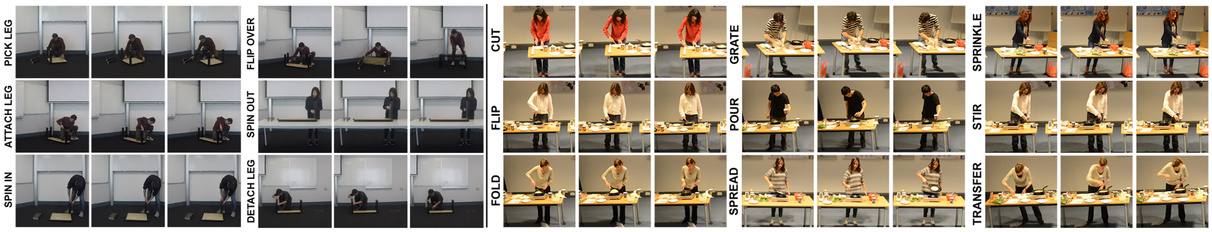


Fig. 7: The Furniture Construction Dataset (left) and the component actions from the Recipe Following Dataset (right).

B. Training

D-ITR-L is compared against other CNN-based temporal inference models.

1) Pre-processing: Videos in these experiments are resized to be compliant with the backbone models they are fed into. Videos from the block stacking and recipe following datasets were also subjected to Gaussian blur and background subtraction [33] to emphasize spatial features that move. These steps were not applied to the furniture construction dataset which possessed many subtle movements (e.g. 'screw in' and 'screw out') that could be occluded by these methods.

2) Spatial Feature Extractors: We contrast four CNN-backbone structures in this work: Two image inference (VGG-16 (abbr. VGG) [28] and Wide ResNet (abbr. WRN) [39]) and two video inference architectures (I3D [4] and TSM [21]). We fine-tune each backbone network to recognize spatial features present in the datasets. Unfortunately, the sparse expression of spatial features in these videos impedes learning. We address this concern by applying a max pool operation over the temporal dimension of the model, thereby reducing sparsity and focusing exclusively on spatial feature presence. Once trained, the weights of the backbone model are fixed and the pooling layer is discarded allowing for inference over the temporal dimension. For consistency all architectures in this work (fine-tuning and inference) are trained over 50 epochs with an Adam optimizer utilizing a learning rate of $1e-3$ and a cross entropy loss.

The ITR Graph possess an exponential number of edges for each spatial feature being investigated. When using D-ITR-L we constrain the number of features output by a CNN backbone by introducing a bottleneck (BN) between the CNN-backbone and the temporal inference layer. The number of features we reduce to is user-defined. For each dataset and model we performed a grid-search to find the best BN size for each model from values of 8, 16, 32, and 64. BN values varied for the block stacking task and are depicted in Table II. On all other datasets and backbone models the best performing BN value was 64.

3) Temporal Inference Architectures: D-ITR-L is compared against three other *separated* approaches that infer temporal features from extracted spatial features: *linear* inference (which conducts only spatial inference), *LSTM* (a RNN), and *TCN* (a CNN). LSTM and TCN are the most classical representations of their respective architectures. These temporal inference architectures each use the same set of spatial features extracted from the CNN-backbones.

V. Results

The results show that D-ITR-L can learn video-scale temporal features in a duration invariant fashion and leverage them for policy learning and activity recognition.

A. Block Stacking

We compare D-ITR-L against baseline temporal inference models when overcoming aforementioned challenges.

1) Variations in Duration: Variations in duration are present in all observations and we measure the accuracy (as a percentage of correct action predictions) across the entire dataset as opposed to a specific observation (Table II). When trained on the dataset with fixed timing, D-ITR-L performed comparably with the LSTM and TCN models. This is expected given the duration-specific nature of these baseline architectures and the dataset. Among the CNN backbones, the video-based architectures (I3D and TSM) performed worse than the image-based architectures (WRN and VGG). We attribute this to the increased challenge present in generalizing spatio-temporal features to video when compared to just spatial features.

D-ITR-L dominated all other approaches when applied on the variably timed data. We attribute this to D-ITR-L's use of duration invariant feature representation. Curiously, the D-ITR-L-driven policy learner generally performed better on variably timed data than it would with the easier to model fixed time data. Movement of blocks in the fixed time dataset was uniformly scheduled over several seconds and expression of features was not as easily delineated from these videos as it was in the variably timed data where block movements was often delayed.

2) Video-Scale Features: Representation of video-scale features is assessed by how well a model can distinguish between the *bg* and *gb* observations (ordering) and the *r*, *rr*, and *rrr* observations (cycles). This experiment was conducted using the variably timed version of the block stacking dataset and the VGG CNN-backbone (Table II). Variably timed movements were collected with fewer constraints and better represent real world data. VGG performed the best of the backbone models investigated on this dataset. Accuracy is a measure of the model's ability to correctly select the next three actions following an observation (Table IV). Three examples were used for each observation.

The linear model was able to recognize the presence of spatial features, but could not learn the temporal content of the observation. This model would default to a single pattern of actions for all observations (i.e. moving the green

	Block Stacking: Fixed Timing					Block Stacking: Variable Timing				
CNN	**BN**	**Linear**	**LSTM**	**TCN**	**D-ITR-L**	**BN**	**Linear**	**LSTM**	**TCN**	**D-ITR-L**
I3D	16	53.3%	53.3%	46.7%	**60.0%**	8	38.0%	38.0%	30.0%	**40.0%**
TSM	16	46.6%	70.0%	**90.0%**	**90.0%**	16	40.0%	82.0%	73.3%	**94.0%**
WRN	16	46.6%	76.6%	**90.0%**	**90.0%**	16	40.0%	65.8%	90.0%	**96.7%**
VGG	32	66.6%	**90.0%**	86.6%	**90.0%**	32	56.0%	72.0%	73.3%	**98.0%**

TABLE II: Accuracy of Policy Learning Approaches Trained on the Block Stacking Datasets

	Furniture Construction				Recipe Following: Component Actions				Recipe Following: Full Recipe			
CNN	**Linear**	**LSTM**	**TCN**	**D-ITR-L**	**Linear**	**LSTM**	**TCN**	**D-ITR-L**	**Linear**	**LSTM**	**TCN**	**D-ITR-L**
I3D	53.20%	56.49%	56.49%	**58.87%**	61.67%	67.07%	70.64%	**73.05%**	10.53%	10.53%	21.05%	**36.84%**
TSM	24.86%	27.06%	25.59%	**48.63%**	57.49%	67.07%	53.29%	**72.46%**	15.79%	15.79%	15.79%	**26.32%**
WRN	17.92%	21.94%	21.39%	**42.60%**	29.98%	52.10%	61.67%	**70.06%**	15.79%	15.79%	15.79%	**21.05%**
VGG	43.69%	50.63%	51.00%	**65.27%**	58.68%	56.28%	59.28%	**66.46%**	15.79%	21.05%	21.05%	**31.58%**

TABLE III: Accuracy on the Recipe Following Dataset

Obs.	**Linear**	**LSTM**	**TCN**	**D-ITR-L**
gb	**100.0%**	**100.0%**	66.7%	**100.0%**
bg	0.0%	**100.0%**	66.7%	**100.0%**
r	0.0%	0.0%	**100.0%**	**100.0%**
rr	0.0%	0.0%	33.3%	**66.7%**
rrr	**100.0%**	**100.0%**	33.3%	**100.0%**

TABLE IV: Accuracy of State Estimators given Specific Observations from the Block Stacking Task

block followed by the blue block for both *bg* and *gb*). The LSTM architecture learned the difference between the ordering of blocks, but could not distinguish between the different cyclical actions. Identification of temporal patterns is easier to accomplish when using dissimilar spatial features than it is with similar features, which is the case with the cyclical activities. The TCN model was able to capture both video-scale features, but did so poorly. TCN is duration-specific and matching a representation to the data is challenging given the possible variations in the duration of spatial events. D-ITR-L was able to distinguish both video-scale features effectively. The singular exception was an instance of *rr* incorrectly identified as *rrr*. The CNN-backbone was inconsistent for this observation and the expression of the red block was noisy. This discrepancy cascaded through the D-ITR-L pipeline resulting in inaccurate event detection and subsequently incorrect temporal feature identification. The quality of our temporal features are fundamentally tied to the quality of the spatial features present in the backbone.

B. Furniture Construction

D-ITR-L outperformed all other temporal inference mechanisms on the furniture construction dataset (Table III) regardless of CNN-backbones. The linear model performed the worst in all cases. This is expected given that the linear model lacks an ability to represent temporal information in an informed way. TCN and LSTM performed similarly to each other. These architectures are duration specific and ill-suited to modelling video-scale features, two properties expressed in the furniture construction dataset. In all cases D-ITR-L showed an improvement over the baseline models, with margins of between 2.38% in I3D and 21.57% in TSM. The highest accuracy achieved on the dataset was 65.27% when D-ITR-L leveraged the spatial features of VGG.

C. Recipe Following

The recipe following dataset was evaluated at two scales: the component actions that compose the recipes and the recipes themselves (Table III). Beginning with the component actions, our results generally aligned with those in Sections V-A and V-B: the linear model in most cases performed the worst, the TCN and LSTM approaches were comparable, and D-ITR-L shows improvement over the nearest baseline video architecture regardless of the type of backbone model used (between 2.4% and 8.39% using I3D and WRN).

When comparing the accuracy given the full recipes, D-ITR-L again outperformed earlier models. However, the accuracy of these models is more muted in comparison to the results of earlier experiments. Many of the baseline models in this dataset failed to generalize to the data and achieved a random accuracy (15.79%) or over fit to a single class label that was under-represented in the validation dataset (10.53%). In contrast, D-ITR-L creates a more robust representation of the data and is able to recognize some of the demonstrated recipes, specifically those that possessed the ‘sprinkle’ component action. This action can be distinguished from the ‘transfer’ action by a cyclical video-scale feature: the number of times the person picks up and scatters ingredients. D-ITR-L’s ability to effectively capture video-scale features provided it the leverage to recognize the more complex videos where other models simply failed to learn.

VI. CONCLUSIONS

Adequate representation of temporal features is a useful first step towards teaching robots to autonomously perform sequential tasks. Regrettably, the current state-of-the-art is inadequate when it comes to modelling this data. We desire a holistic architecture capable of capturing video-scale temporal features in a duration invariant manner. D-ITR-L addresses this niche by building temporal features directly on top of the spatial features learned by popular CNN-backbone models. We have demonstrated, through several experiments, that the temporal features extracted by D-ITR-L (when learned by a GCN) are a better alternative to those of contemporary deep learning models.

ACKNOWLEDGMENT

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

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] 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.

[3] Estuardo Carpio, Madison Clark-Turner, Paul Gesel, and Momotaz Begum. Leveraging temporal reasoning for policy selection in learning from demonstration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7798–7804. IEEE, 2019.

[4] 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.

[5] Madison Clark-Turner. Temporal feature lfd, 2021. https://github.com/AssistiveRoboticsUNH/temporal_feature_lfd.

[6] 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.

[7] Richard Cubek, Wolfgang Ertel, and Günther Palm. High-level learning from demonstration with conceptual spaces and subspace clustering. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2592–2597. IEEE, 2015.

[8] Debidatta Dwibedi, Pierre Sermanet, and Jonathan Tompson. Temporal reasoning in videos using convolutional gated recurrent units. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1111–1116, 2018.

[9] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[10] Jiangfan Feng, Yuanyuan Liu, and Lin Wu. Bag of visual words model with deep spatial features for geographical scene classification. *Computational intelligence and neuroscience*, 2017, 2017.

[11] 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.

[12] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, volume 1, page 3, 2017.

[13] Daniel H Grollman and Odest Chadwicke Jenkins. Incremental learning of subtasks from unsegmented demonstration. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 261–266. IEEE, 2010.

[14] Tengda Han, Jue Wang, Anoop Cherian, and Stephen Gould. Human action forecasting by learning task grammars. *arXiv:1709.06391*, 2017.

[15] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2000–2009, 2019.

[16] Xiang Jiang, Erico N de Souza, Ahmad Pesaranghader, Baifan Hu, Daniel L Silver, and Stan Matwin. Trajectorynet: An embedded gps trajectory representation for point-based classification using recurrent neural networks. *arXiv preprint arXiv:1705.02636*, 2017.

[17] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151:107398, 2021.

[18] Stepan Komkov, Maksim Dzabraev, and Aleksandr Petiushko. Mutual modality learning for video action classification. *arXiv preprint arXiv:2011.02543*, 2020.

[19] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[20] Kyuhwa Lee, Dimitri Ognibene, Hyung Jin Chang, Tae-Kyun Kim, and Yiannis Demiris. Stare: Spatio-temporal attention relocation for multiple structured activities detection. *IEEE Transactions on Image Processing*, 24(12):5916–5927, 2015.

[21] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019.

[22] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[23] Anahita Mohseni-Kabir, Charles Rich, Sonia Chernova, Candace L Sidner, and Daniel Miller. Interactive hierarchical task learning from a single demonstration. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 205–212, 2015.

[24] Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2146–2153. IEEE, 2017.

[25] Ilija Radosavovic, Xiaolong Wang, Lerrel Pinto, and Jitendra Malik. State-only imitation learning for dexterous manipulation. *arXiv preprint arXiv:2004.04650*, 2020.

[26] 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.

[27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

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

[29] Radoslav Skoviera, Karla Stepanova, Michael Tesar, Gabriela Sejnova, Jiri Sedlar, Michal Vavrecka, Robert Babuska, and Josef Sivic. Teaching robots to imitate a human with no on-teacher sensors. what are the key challenges? *arXiv preprint arXiv:1901.08335*, 2019.

[30] Sam Toyer, Anoop Cherian, Tengda Han, and Stephen Gould. Human pose forecasting via deep Markov models. In *DICTA*, 2017.

[31] 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.

[32] Amin Ullah, Khan Muhammad, Weiping Ding, Vasile Palade, Ijaz Ul Haq, and Sung Wook Baik. Efficient activity recognition using lightweight cnn and ds-gru network for surveillance applications. *Applied Soft Computing*, 103:107102, 2021.

[33] Antoine Vacavant, Thierry Chateau, Alexis Wilhelm, and Laurent Lequievre. A benchmark dataset for outdoor foreground/background extraction. In *Asian Conference on Computer Vision*, pages 291–300. Springer, 2012.

[34] 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.

[35] Kun Xia, Jianguang Huang, and Hanyu Wang. Lstm-cnn architecture for human activity recognition. *IEEE Access*, 8:56855–56866, 2020.

[36] Danfei Xu, Suraj Nair, Yuke Zhu, Julian Gao, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Neural task programming: Learning to generalize across hierarchical tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3795–3802. IEEE, 2018.

[37] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020.

[38] 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.

[39] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[40] 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.

[41] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the*

European Conference on Computer Vision (ECCV), pages 803–818, 2018.

[42] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 695–712, 2018.