

Temporal Relational Reasoning in Videos

Bolei Zhou, Alex Andonian, Aude Oliva, Antonio Torralba

MIT CSAIL

{bzhou,aandonia,oliva,torralba}@csail.mit.edu

Abstract. Temporal relational reasoning, the ability to link meaningful transformations of objects or entities over time, is a fundamental property of intelligent species. In this paper, we introduce an effective and interpretable network module, the Temporal Relation Network (TRN), designed to learn and reason about temporal dependencies between video frames at multiple time scales. We evaluate TRN-equipped networks on activity recognition tasks using three recent video datasets - Something-Something, Jester, and Charades - which fundamentally depend on temporal relational reasoning. Our results demonstrate that the proposed TRN gives convolutional neural networks a remarkable capacity to discover temporal relations in videos. Through only sparsely sampled video frames, TRN-equipped networks can accurately predict human-object interactions in the Something-Something dataset and identify various human gestures on the Jester dataset with very competitive performance. TRN-equipped networks also outperform two-stream networks and 3D convolution networks in recognizing daily activities in the Charades dataset. Further analyses show that the models learn intuitive and interpretable visual common sense knowledge in videos¹.

1 Introduction

The ability to reason about the relations between entities over time is crucial for intelligent decision-making. Temporal relational reasoning allows intelligent species to analyze the current situation relative to the past and formulate hypotheses on what may happen next. For example (Fig.1), given two observations of an event, people can easily recognize the temporal relation between two states of the visual world and deduce what has happened between the two frames of a video².

Temporal relational reasoning is critical for activity recognition, forming the building blocks for describing the steps of an event. A single activity can consist of several temporal relations at both short-term and long-term timescales. For example, the activity of *sprinting* contains the long-term temporal relations of crouching at the starting blocks, running on track, and finishing at the end line, while it also includes the short-term temporal relations of periodic hands and feet movement.

¹ Code and models are available at <http://relation.csail.mit.edu/>.

² Answer: a) Poking a stack of cans so it collapses; b) Stack something; c) Tidying up a closet; d) Thumb up.



Fig. 1: What takes place between two observations? (see answer below the first page). Humans can easily infer the temporal relations and transformations between these observations, but this task remains difficult for neural networks.

Activity recognition in videos has been one of the core topics in computer vision. However, it remains difficult due to the ambiguity of describing activities at appropriate timescales [1]. Many video datasets, such as UCF101 [2], Sport1M [3], and THUMOS [4], include many activities that can be recognized without reasoning about the long-term temporal relations: still frames and optical flow are sufficient to identify many of the labeled activities. Indeed, the classical two-stream Convolutional Neural Network [5] and the recent I3D Network [6], both based on frames and optical flow, perform activity recognition very well on these datasets.

However, convolutional neural networks still struggle in situations where data and observations are limited, or where the underlying structure is characterized by transformations and temporal relations, rather than the appearance of certain entities [7,8]. It remains remarkably challenging for convolutional neural networks to reason about temporal relations and to anticipate what transformations are happening to the observations. Fig.1 shows such examples. The networks are required to discover visual common sense knowledge over time beyond the appearance of objects in the frames and the optical flow.

In this work, we propose a simple and interpretable network module called Temporal Relation Network (TRN) that enables temporal relational reasoning in neural networks. This module is inspired by the relational network proposed in [7], but instead of modeling the spatial relations, TRN aims to describe the temporal relations between observations in videos. Thus, TRN can learn and discover possible temporal relations at multiple time scales. TRN is a general and extensible module that can be used in a plug-and-play fashion with any existing CNN architecture. We apply TRN-equipped networks on three recent video datasets (Something-Something [9], Jester [10], and Charades [11]), which are constructed for recognizing different types of activities such as human-object interactions and hand gestures, but all depend on temporal relational reasoning. The TRN-equipped networks achieve very competitive results even given only discrete RGB frames, bringing significant improvements over baselines. Thus TRN provides a practical solution for standard neural networks to solve activity recognition tasks using temporal relational reasoning.

1.1 Related Work

Convolutional Neural Networks for Activity Recognition. Activity recognition in videos is a core problem in computer vision. With the rise of deep convolutional neural networks (CNNs) which achieve state-of-the-art performance on image recognition tasks [12,13], many works have looked into designing effective deep convolutional neural networks for activity recognition [3,5,14,15,16,6]. For instance, various approaches of fusing RGB frames over the temporal dimension are explored on the Sport1M dataset [3]. Two stream CNNs with one stream of static images and the other stream of optical flows are proposed to fuse the information of object appearance and short-term motions [5]. 3D convolutional networks [15] use 3D convolution kernels to extract features from a sequence of dense RGB frames. Temporal Segment Networks sample frames and optical flow on different time segments to extract information for activity recognition [16]. A CNN+LSTM model, which uses a CNN to extract frame features and an LSTM to integrate features over time, is also used to recognize activities in videos [14]. Recently, I3D networks [6] use two stream CNNs with inflated 3D convolutions on both dense RGB and optical flow sequences to achieve state of the art performance on the Kinetics dataset [17]. There are several important issues with existing CNNs for action recognition: 1) The dependency on beforehand extraction of optical flow lowers the efficiency of the recognition system; 2) The 3D convolutions on sequences of dense frames are computationally expensive, given the redundancy in consecutive frames; 3) Since sequences of frames fed into the network are usually limited to 20 to 30 frames, it is difficult for the networks to learn long-term temporal relations among frames. To address these issues, the proposed Temporal Relation Network sparsely samples individual frames and then learns their causal relations, which is much more efficient than sampling dense frames and convolving them. We show that TRN-equipped networks can efficiently capture temporal relations at multiple time scales and outperform dense frame-based networks using only sparsely sampled video frames.

Temporal Information in Activity Recognition. For activity recognition on many existing video datasets such as UCF101 [2], Sport1M [3], THUMOS [4], and Kinetics [17], the appearance of still frames and short-term motion such as optical flow are the most important information to identify the activities. Thus, activity recognition networks such as Two Stream network [5] and the I3D network [6] are tailored to capture these short-term dynamics of dense frames. Therefore, existing networks don't need to build temporal relational reasoning abilities. On the other hand, recently there have been various video datasets collected via crowd-sourcing, which focus on sequential activity recognition: Something-Something dataset [9] is collected for generic human-object interaction. It has video classes such as 'Dropping something into something', 'Pushing something with something', and even 'Pretending to open something without actually opening it'. Jester dataset [10] is another recent video dataset for gesture recognition. Videos are recorded by crowd-source workers performing 27 kinds of gestures such as 'Thumbing up', 'Swiping Left', and 'Turning hand counterclockwise'. Charades dataset is also a high-level human activity dataset

that collects videos by asking crowd workers to perform a series of home activities and then record themselves [11]. For recognizing the complex activities in these three datasets, it is crucial to integrate temporal relational reasoning into the networks. Besides, many previous works model the temporal structures of videos for action recognition and detection using bag of words, motion atoms, or action grammar [18,19,20,21,22]. Instead of designing temporal structures manually, we use a more generic structure to learn the temporal relations in end-to-end training. One relevant work on modeling the cause-effect in videos is [23]. [23] uses a two-stream siamese network to learn the transformation matrix between two frames, then uses brute force search to infer the action category. Thus the computation cost is high. Our TRN much more efficiently integrates the multiple frames information, both in training and testing.

Relational Reasoning and Intuitive Physics. Recently, relational reasoning module has been proposed for visual question answering with super-human performance [7]. Our work is inspired by that work, but we focus on modeling the multi-scale temporal relations in videos. In the domain of robot self-supervised learning, many models have been proposed to learn the intuitive physics among frames. Given an initial state and a goal state, the inverse dynamics model with reinforcement learning is used to infer the transformation between the object states [24]. Physical interaction and observations are also used to train deep neural networks [25]. Time contrast networks are used for self-supervised imitation learning of object manipulation from third-person video observation [26]. Our work aims to learn various temporal relations in videos in a supervised learning setting. The proposed TRN can be extended to self-supervised learning for robot object manipulation.

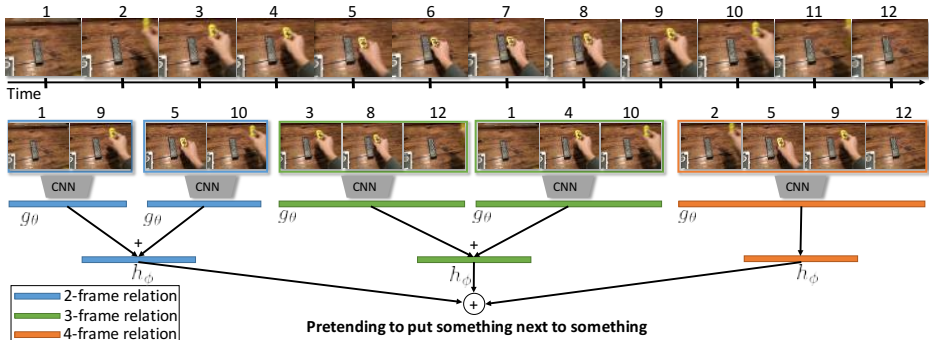


Fig. 2: The illustration of Temporal Relation Networks. Representative frames of a video (shown above) are sampled and fed into different frame relation modules. Only a subset of the 2-frame, 3-frame, and 4-frame relations are shown, as there are higher frame relations included.

2 Temporal Relation Networks

In this section, we introduce the framework of Temporal Relation Networks. It is simple and can be easily plugged into any existing convolutional neural network

architecture to enable temporal relational reasoning. In later experiments, we show that TRN-equipped networks discover interpretable visual common sense knowledge to recognize activities in videos.

2.1 Defining Temporal Relations

Inspired by the relational reasoning module for visual question answering [7], we define the pairwise temporal relation as a composite function below:

$$T_2(V) = h_\phi \left(\sum_{i < j} g_\theta(f_i, f_j) \right) \quad (1)$$

where the input is the video V with n selected ordered frames as $V = \{f_1, f_2, \dots, f_n\}$, where f_i is a representation of the i^{th} frame of the video, e.g., the output activation from some standard CNN. The functions h_ϕ and g_θ fuse features of different ordered frames. Here we simply use multilayer perceptrons (MLP) with parameters ϕ and θ respectively. For efficient computation, rather than adding all the combination pairs, we uniformly sample frames i and j and sort each pair.

We further extend the composite function of the 2-frame temporal relations to higher frame relations such as the 3-frame relation function below:

$$T_3(V) = h'_\phi \left(\sum_{i < j < k} g'_\theta(f_i, f_j, f_k) \right) \quad (2)$$

where the sum is again over sets of frames i, j, k that have been uniformly sampled and sorted.

2.2 Multi-Scale Temporal Relations

To capture temporal relations at multiple time scales, we use the following composite function to accumulate frame relations at different scales:

$$MT_N(V) = T_2(V) + T_3(V) \dots + T_N(V) \quad (3)$$

Each relation term T_d captures temporal relationships between d ordered frames. Each T_d has its own separate $h_\phi^{(d)}$ and $g_\theta^{(d)}$. Notice that for any given sample of d frames for each T_d , all the temporal relation functions are end-to-end differentiable, so they can all be trained together with the base CNN used to extract features for each video frame. The overall network framework is illustrated in Fig.2.

2.3 Efficient Training and Testing

When training a multi-scale temporal network, we could sample the sums by selecting different sets of d frames for each T_d term for a video. However, we use a sampling scheme that reduces computation significantly. First, we uniformly

sample a set of N frames from the N segments of the video, $V_N^* \subset V$, and we use V_N^* to calculate $T_N(V)$. Then, for each $d < N$, we choose k random subsamples of d frames $V_{kd}^* \subset V_N^*$. These are used to compute the d -frame relations for each $T_d(V)$. This allows kN temporal relations to be sampled while run the base CNN on only N frames, while all the parts are end-to-end trained together.

At testing time, we can combine the TRN-equipped network with a queue to process streaming video very efficiently. A queue is used to cache the extracted CNN features of the equidistant frames sampled from the video, then those features are further combined into different relation tuples which are further summed up to predict the activity. The CNN feature is extracted from incoming key frame only once then enqueued, thus TRN-equipped networks is able to run in real-time on a desktop to processing streaming video from a webcam.

3 Experiments

We evaluate the TRN-equipped networks on a variety of activity recognition tasks. For recognizing activities that depend on temporal relational reasoning, TRN-equipped networks outperform a baseline network without a TRN by a large margin. We achieve highly competitive results on the Something-Something dataset for human-interaction recognition [9] and on the Jester dataset for hand gesture recognition [10]. The TRN-equipped networks also obtain competitive results on activity classification in the Charades dataset [11], outperforming the Flow+RGB ensemble models [27,11] using only sparsely sampled RGB frames.

The statistics of the three datasets Something-Something dataset (Something-V1 [9] and Something-V2 [28] where the Something-V2 is the 2nd release of the dataset in early July 2018) [9,28], Jester dataset [10], and Charades dataset [11] are listed in Table 1. All three datasets are crowd-sourced, in which the videos are collected by asking the crowd-source workers to record themselves performing instructed activities. Unlike the Youtube-type videos in UCF101 and Kinetics, there is usually a clear start and end of each activity in the crowd-sourced video, emphasizing the importance of temporal relational reasoning.

Table 1: Statistics of the datasets used in evaluating the TRNs.

| Dataset | Classes | Videos | Type |
|--------------|---------|---------|--------------------------|
| Something-V1 | 174 | 108,499 | human-object interaction |
| Something-V2 | 174 | 220,847 | human-object interaction |
| Jester | 27 | 148,092 | human hand gesture |
| Charades | 157 | 9,848 | daily indoor activity |

3.1 Network Architectures and Training

The networks used for extracting image features play an important factor in visual recognition tasks [29]. Features from deeper networks such as ResNet [30] usually perform better. Our goal here is to evaluate the effectiveness of the TRN module for temporal relational reasoning in videos. Thus, we fix the base network architecture to be the same throughout all the experiments and

compare the performance of the CNN model with and without the proposed TRN modules.

We adopt Inception with Batch Normalization (BN-Inception) pretrained on ImageNet used in [31] because of its balance between accuracy and efficiency. We follow the training strategies of partial BN (freezing all the batch normalization layers except the first one) and dropout after global pooling as used in [16]. We keep the network architecture of the MultiScale TRN module and the training hyper-parameters the same for training models on all the three datasets. We set $k = 3$ in the experiments as the number of accumulated relation triples in each relation module. g_ϕ is simply a two-layer MLP with 256 units per layer, while h_ϕ is a one-layer MLP with the unit number matching the class number. The CNN features for a given frame is the activation from the BN-Inception’s global average pooling layer (before the final classification layer). Given the BN-Inception as the base CNN, the training can be finished in less than 24 hours for 100 training epochs on a single Nvidia Titan Xp GPU. In the Multi-Scale TRN, we include all the TRN modules from 2-frame TRN up to 8-frame TRN (thus $N = 8$ in Eq.3), as including higher frame TRNs brings marginal improvement and lowers the efficiency.

3.2 Results on Something-Something Dataset

Something-Something is a recent video dataset for human-object interaction recognition. There are 174 classes, some of the ambiguous activity categories are challenging, such as ‘Tearing Something into two pieces’ versus ‘Tearing Something just a little bit’, ‘Turn something upside down’ versus ‘Pretending to turn something upside down’. We can see that the temporal relations and transformations of objects rather than the appearance of the objects characterize the activities in the dataset.

The results on the validation set and test set of Something-V1 and Something-V2 datasets are listed in Table 2a. The baseline is the base network trained on single frames randomly selected from each video. Networks with TRNs outperform the single frame baseline by a large margin. We construct the 2-stream TRN by simply averaging the predicted probabilities from the the two streams for any given video). The 2-stream TRN further improves the accuracy on the validation set of Something-v1 and Something-v2 to **42.01%** and **55.52%** respectively. Note that we found that the optical stream with average pooling of frames used in TSN [16] achieves better score than the one with the proposed temporal relational pooling so we use 8-frame TSN on optical flow stream, which gets 31.63% and 46.41% on the validation set of Something-V1 and Something-V2 respectively. We further submit MultiScale TRN and 2-stream TRN predictions on the test set, the results are shown in Table 2.a

We compare the TRN with TSN [16], to verify the importance of temporal orders. Instead of concatenating the features of temporal frames, TSN simply averages the deep features so that the model only captures the co-occurrence rather than the temporal ordering of patterns in the features. We keep all the training conditions the same, and vary the number of frames used by two models.

As shown in Table 2b, our models outperform TSNs by a large margin. This result shows the importance of frame order for temporal relation reasoning. We also see that additional frames included in the relation bring further significant improvements to TRN.

| | Something-V1 | | Something-V2 | | TRN TSN | | |
|----------------|--------------|-------|--------------|-------------|---------|-------|-------|
| | Val | Test | Val | Test | 2-fr. | 22.23 | 16.72 |
| Baseline | 11.41 | - | - | - | 3-fr. | 26.22 | 17.30 |
| MultiScale TRN | 34.44 | 33.60 | 48.80/77.64 | 50.85/79.33 | 5-fr. | 30.39 | 18.11 |
| 2-Stream TRN | 42.01 | 40.71 | 55.52/83.06 | 56.24/83.15 | 7-fr. | 31.01 | 18.48 |

(a)
(b)

Table 2: (a) Results on the validation set and test set of the Something-V1 Dataset (Top1 Accuracy) and Something-V2 Dataset (Both Top1 and Top5 accuracy are reported). (b) Comparison of TRN and TSN as the number of frames (fr.) varies on the validation set of the Something-V1. TRN outperforms TSN in a large margin as the number of frames increases, showing the importance of temporal order.

3.3 Results on Jester and Charades

We further evaluate the TRN-equipped networks on the Jester dataset, which is a video dataset for hand gesture recognition with 27 classes. The results on the validation set of the Jester dataset are listed in Table 3a. The result on the test set and comparison with the top methods are listed in Table 3b. MultiScale TRN again achieves competitive performance as close to 95% Top1 accuracy.

| Val | | Test | |
|----------------|--------------|-----------------------|--------------|
| Baseline | 63.60 | 20BN Jester System | 82.34 |
| 2-frame TRN | 75.65 | VideoLSTM | 85.86 |
| 3-frame TRN | 81.45 | Guillaume Berger | 93.87 |
| 4-frame TRN | 89.38 | Ford’s Gesture System | 94.11 |
| 5-frame TRN | 91.40 | Besnet | 94.23 |
| MultiScale TRN | 95.31 | MultiScale TRN | 94.78 |

(a)
(b)

Table 3: Jester Dataset Results on (a) the validation set and (b) the test set.

We evaluate the MultiScale TRN on the recent Charades dataset for daily activity recognition. The results are listed in Table 4. Our method outperforms various methods such as 2-stream networks and C3D [11], and the recent Asynchronous Temporal Field (TempField) method [27].

The qualitative prediction results of the Multi-Scale TRN on the three datasets are shown in Figure 3. The examples in Figure 3 demonstrate that the TRN model is capable of correctly identifying actions for which the overall temporal ordering of frames is essential for a successful prediction. For example, the turning hand counterclockwise category would assume a different class label when

models trained to recognize the activity? To obtain a sequence of representative frames for each TRN, we first compute the features of the equidistant frames from a video, then randomly combine them to generate different frame relation tuples and pass them into the TRNs. Finally we rank the relation tuples using the responses of different TRNs. Figure 4 shows the top representative frames voted by different TRNs to recognize an activity in the same video. We can see that the TRNs learn the temporal relations that characterize an activity. For comparatively simple actions, a single frame is sufficient to establish some degree of confidence in the correct action, but is vulnerable to mistakes when a transformation is present. 2-frame TRN picks up the two frames that best describe the transformation. Meanwhile, for more difficult activity categories such as ‘Pretending to poke something’, two frames are not sufficient information for even a human observer to differentiate. Similarly, the network needs additional frames in the TRNs to correctly recognize the behavior.

Thus the progression of representative frames and their corresponding class predictions inform us about how temporal relations may help the model reason about more complex behavior. One particular example is the last video in Figure 4: The action’s context given by a single frame - a hand close to a book - is enough to narrow down the top prediction to a qualitatively plausible action, unfolding something. A similar, two-frame relation marginally increases the probability the initial prediction, although these two frames would not be sufficient for even human observers to make the correct prediction. Now, the three frame-relation begins to highlight a pattern characteristic to Something-Somethings set of *pretending* categories: the initial frames closely resemble a certain action, but the later frames are inconsistent with the completion of that action as if it never happened. This relation helps the model to adjust its prediction to the correct class. Finally, the upward motion of the individuals hand in the third frame of the 4-frame relation further increases the discordance between the *anticipated* and *observed* final state of the scene; a motion resembling the action appeared to take place with no effect on the object, thus, solidifying confidence in the correct class prediction.

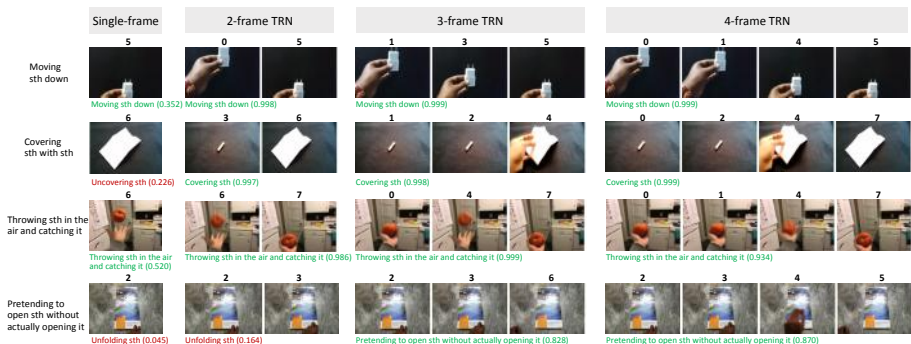


Fig. 4: The top representative frames determined by single frame baseline network, the 2-frame TRN, 3-frame TRN, and 4-frame TRN. TRNs learn to capture the essence of an activity only given a limited number of frames. Videos are from the validation set of the Something-Something dataset

Temporal Alignment of Videos. The observation that the representative frames identified by the TRN are consistent across instances of an action category suggests that the TRN is well suited for the task of temporally aligning videos with one another. Here, we wish to synchronize actions across multiple videos by establishing a correspondence between their frame sequences. Given several video instances of the same action, we first select the most representative frames for each video and use their frame indices as “landmark”, temporal anchor points. Then, we alter the frame rate of video segments between two consecutive anchor points such that all of the individual videos arrive at the anchor points at the same time. Fig.5 shows the samples from the aligned videos. We can see different stages of an action are captured by the temporal relation. The temporal alignment is also an exclusive application of our TRN model, which cannot be done by previous video networks 3D convNet or two-stream networks.

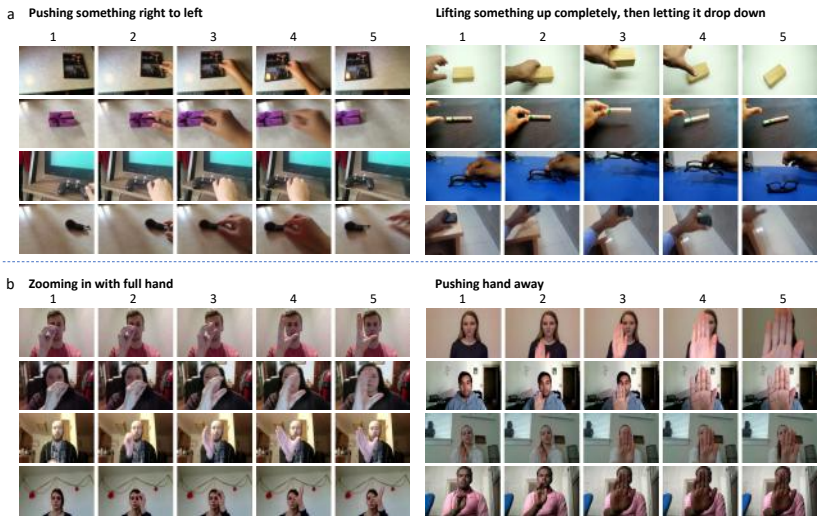


Fig. 5: Temporal alignment of videos from the (a) Something-Something and (b) Jester datasets using the most representative frames as temporal anchor points. For each action, 4 different videos are aligned using 5 temporal anchor points.

Importance of temporal order for activity recognition. To verify the importance of the temporal order of frames for activity recognition, we conduct an experiment to compare the scenario with input frames in temporal order and in shuffled order when training the TRNs, as shown in Figure 6a. For training the shuffled TRNs, we randomly shuffle the frames in the relation modules. The significant difference on the Something-Something dataset shows the importance of the temporal order in the activity recognition. More interestingly, we repeat the same experiment on the UCF101 dataset [2] and observe no difference between the ordered frames and shuffled frames. That shows activity recognition for the Youtube-type videos in UCF101 doesn’t necessarily require the temporal reasoning ability since there are not so many casual relations associated with an already on-going activity.

To further investigate how temporal ordering influences activity recognition in TRN, we examine and plot the categories that show the largest differences in the class accuracy between ordered and shuffled inputs drawn from the Something-Something dataset, in Figure 6b. In general, actions with strong ‘directionality and large, one-way movements, such as ‘Moving something down’, appear to benefit the most from preserving the correct temporal ordering. This observation aligns with the idea that the disruption of continuous motion and a potential consequence of shuffling video frames, would likely confuse a human observer, as it would go against our intuitive notions of physics.

Interestingly, the penalty for shuffling frames of relatively static actions is less severe if penalizing at all in some cases, with several categories marginally benefiting from shuffled inputs, as observed with the category ‘putting something that can’t roll onto a slanted surface so it stays where it is’. Here, simply learning the coincidence of frames rather than temporal transformations may be sufficient for the model to differentiate between similar activities and make the correct prediction. Particularly in challenging ambiguous cases, for example ‘Pretending to throw something’ where the release point is partially or completely obscured from view, disrupting a strong ‘sense of motion’ may bias model predictions away from the likely alternative, ‘throwing something’, frequently but incorrectly selected by the ordered model, thus giving rise to a curious difference in accuracy for that action.

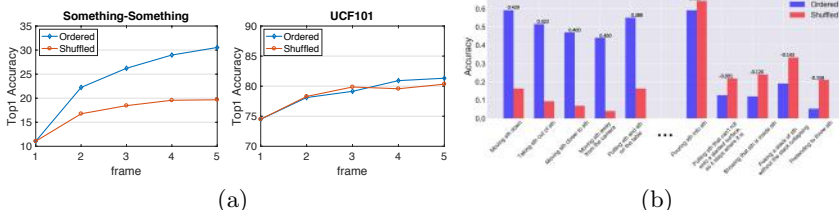


Fig.6: (a) Accuracy obtained using ordered frames and shuffled frames, on Something-Something and UCF101 dataset respectively. On Something-Something, the temporal order is critical for recognizing the activity. But recognizing activities in UCF101 does not necessarily require temporal relational reasoning. (b) The top 5 action categories that exhibited the largest gain and the least gain (negative) respectively between ordered and shuffled frames as inputs. Actions with directional motion appear to suffer most from shuffled inputs.

The difference between TSN and TRN is at using different frame feature pooling strategies, where TRN using Temporal Relation(TR) pool emphasizes on capturing the temporal dependency of frames while TSN simply uses average pool to ignore the temporal order. We evaluate the two pool strategies in detail as shown in Table 5. The difference in the performance using average pool and TR pool actually reflects the importance of temporal orders in a video dataset. The tested datasets are categorized by the video source, where the first three are Youtube videos, the other three are videos crowdsourced from AMT. The base CNN is BNInception. Both of the models use 8 frames. Interestingly, the

models with average pool and TR pool achieve similar accuracy on Youtube videos, thus recognizing Youtube videos doesn’t require much temporal order reasoning, which might be due to that activity in the randomly trimmed Youtube videos doesn’t usually have a clear action start or end. On the other hand, the crowdsourced video has just one activity with clearly start and end, thus temporal relation pool brings significant improvement.

| Dataset | Youtube videos | | | Crowdsourced videos | | |
|--------------|----------------|--------------|--------------|---------------------|--------------|--------------|
| | UCF | Kinetics | Moments | Something | Jester | Charades |
| Num.Classes | 101 | 200 | 339 | 174 | 27 | 157 |
| Average Pool | 82.69 | 63.34 | 24.11 | 19.53 | 85.41 | 11.32 |
| TR Pool | 83.83 | 63.18 | 25.94 | 34.44 | 95.31 | 25.20 |

Table 5: Accuracy on six video datasets for models with two pool strategies.

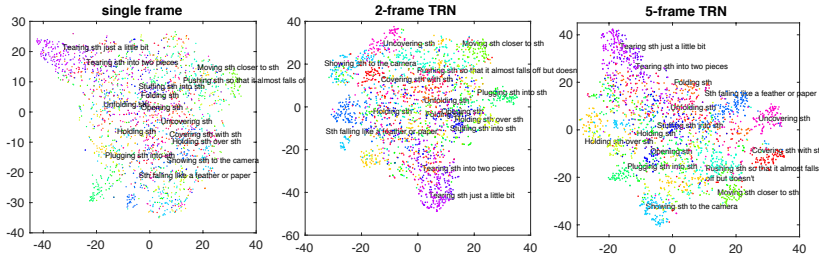


Fig. 7: t-SNE plot of the video samples of the 15 classes using the deep features from the single-frame baseline, 2-frame TRN, and 5-frame TRN. Higher frame TRN can better differentiate activities in Something-Something dataset.

t-SNE visualization of activity similarity. Figure 7 shows the t-SNE visualization for embedding the high-level features from the single frame baseline, the 3-frame TRN, and the 5-frame TRN, for the videos of the 15 most frequent activity classes in the validation set. We can see that the features from 2-frame and 5-frame TRNs can better differentiate activity categories. We also observe the similarity among categories in the visualization map. For example, ‘Tearing something into two pieces’ is very similar to ‘Tearing something just a little bit’, and the categories ‘Folding something’, ‘Unfolding something’, ‘Holding something’, ‘Holding something over something’ are clustered together.

Table 6: Early activity recognition using the MultiScale TRN on Something-Something and Jester dataset. Only the first 25% and 50% of frames are given to the TRN to predict activities. Baseline is the model trained on single frames.

| Frames | Something | | Jester | |
|-----------|-----------|-------|----------|-------|
| | baseline | TRN | baseline | TRN |
| first 25% | 9.08 | 11.14 | 27.25 | 34.23 |
| first 50% | 10.10 | 19.10 | 41.43 | 78.42 |
| full | 11.41 | 33.01 | 63.60 | 93.70 |

Early Activity Recognition. Recognizing activities early or even anticipating and forecasting activities before they happen or fully happen is a chal-

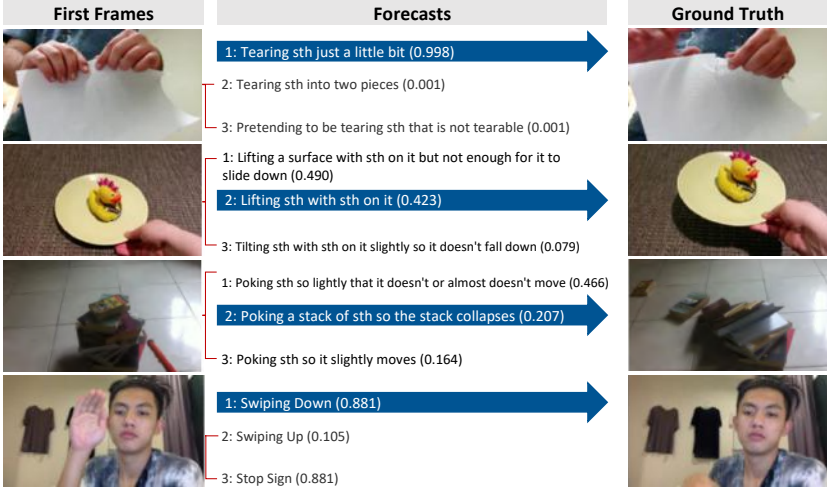


Fig. 8: Early recognition of activity when only given the first 25% frames. The first 25% of each video, represented by the first frame shown in the left column, is used to generate the top 3 anticipated forecasts and corresponding probabilities listed in the middle column. The ground truth label is highlighted by a blue arrow which points to the last frame of the video on the right.

lenging yet less explored problem in activity recognition. Here we evaluate our TRN model on early recognition of activity when given only the first 25% and 50% of the frames in each validation video. Results are shown in Table 6. For comparison, we also include the single frame baseline, which is trained on randomly sampled individual frames from a video. We see that TRN can use the learned temporal relations to anticipate activity. The performance increases as more ordered frames are received. Figure 8 shows some examples of anticipating activities using only first 25% and 50% frames of a video. A qualitative review of these examples reveals that model predictions on only initial frames do serve as very reasonable forecasts despite being given task with a high degree of uncertainty even for human observers.

4 Conclusion

We proposed a simple and interpretable network module called Temporal Relation Network (TRN) to enable temporal relational reasoning in neural networks for videos. We evaluated the proposed TRN on several recent datasets and established competitive results using only discrete frames. Finally, we have shown that TRN modules discover visual common sense knowledge in videos.

Acknowledgement: This work was partially funded by DARPA XAI program No. FA8750-18-C-0004, NSF Grant No. 1524817, and Samsung to A.T.; the Vannevar Bush Faculty Fellowship program funded by the ONR grant No. N00014-16-1-3116 to A.O.. It is also supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00341. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

1. Sigurdsson, G.A., Russakovsky, O., Gupta, A.: What actions are needed for understanding human actions in videos? arXiv preprint arXiv:1708.02696 (2017)
2. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. *Proc. CVPR* (2012)
3. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: *Proc. CVPR*. (2014)
4. Gorban, A., Idrees, H., Jiang, Y., Zamir, A.R., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes. In: *CVPR workshop*. (2015)
5. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: *In Advances in Neural Information Processing Systems*. (2014) 568–576
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. arXiv preprint arXiv:1705.07750 (2017)
7. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. arXiv preprint arXiv:1706.01427 (2017)
8. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. *Behavioral and Brain Sciences* (2016) 1–101
9. Goyal, R., Kahou, S., Michalski, V., Materzyńska, J., Westphal, S., Kim, H., Haelen, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The” something something” video database for learning and evaluating visual common sense. *Proc. ICCV* (2017)
10. : Twentybn jester dataset: a hand gesture dataset. <https://www.twentybn.com/datasets/jester> (2017)
11. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: *European Conference on Computer Vision*, Springer (2016) 510–526
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. (2012) 1097–1105
13. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: *Advances in neural information processing systems*. (2014) 487–495
14. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2015) 2625–2634
15. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: *Proc. CVPR*. (2015)
16. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: *Proc. ECCV*. (2016)
17. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)

18. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization of actions with atoms. *IEEE transactions on pattern analysis and machine intelligence* **35**(11) (2013) 2782–2795
19. Pirsiavash, H., Ramanan, D.: Parsing videos of actions with segmental grammars. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2014) 612–619
20. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: *Proc. ICCV*. (2013) 3551–3558
21. Gaidon, A., Harchaoui, Z., Schmid, C.: Activity representation with motion hierarchies. *International journal of computer vision* **107**(3) (2014) 219–238
22. Wang, L., Qiao, Y., Tang, X.: Mofap: A multi-level representation for action recognition. *International Journal of Computer Vision* **119**(3) (2016) 254–271
23. Wang, X., Farhadi, A., Gupta, A.: Actions~ transformations. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. (2016) 2658–2667
24. Agrawal, P., Nair, A.V., Abbeel, P., Malik, J., Levine, S.: Learning to poke by poking: Experiential learning of intuitive physics. In: *Advances in Neural Information Processing Systems*. (2016) 5074–5082
25. Pinto, L., Gandhi, D., Han, Y., Park, Y.L., Gupta, A.: The curious robot: Learning visual representations via physical interactions. In: *European Conference on Computer Vision*, Springer (2016) 3–18
26. Sermanet, P., Lynch, C., Hsu, J., Levine, S.: Time-contrastive networks: Self-supervised learning from multi-view observation. *arXiv preprint arXiv:1704.06888* (2017)
27. Sigurdsson, G.A., Divvala, S., Farhadi, A., Gupta, A.: Asynchronous temporal fields for action recognition. (2017)
28. Mahdisoltani, F., Berger, G., Gharbieh, W., Fleet, D., Memisevic, R.: Fine-grained video classification and captioning. *arXiv preprint arXiv:1804.09235* (2018)
29. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. (2014)
30. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2016) 770–778
31. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*. (2015) 448–456