Video Time: Properties, Encoders and Evaluation

Amir Ghodrati
Efstratios Gavves
Cees G. M. Snoek
{a.ghodrati,egavves,cgmsnoek}@uva.nl

QUVA Lab
University of Amsterdam
Netherlands

Abstract

Time-aware encoding of frame sequences in a video is a fundamental problem in video understanding. While many attempted to model time in videos, an explicit study on quantifying video time is missing. To fill this lacuna, we aim to evaluate video time explicitly. We describe three properties of video time, namely *a) temporal asymmetry, b) temporal continuity* and *c) temporal causality*. Based on each we formulate a task able to quantify the associated property. This allows assessing the effectiveness of modern video encoders, like C3D and LSTM, in their ability to model time. Our analysis provides insights about existing encoders while also leading us to propose a new video time encoder, which is better suited for the video time recognition tasks than C3D and LSTM. We believe the proposed meta-analysis can provide a reasonable baseline to assess video time encoders on equal grounds on a set of temporal-aware tasks[1].

1 Introduction

The goal of this paper is to investigate and evaluate the manifestation of time in video data. Modeling time in video is crucial for understanding [8, 45], prediction [5, 30] and reasoning [4, 12]. Hence, it received increased attention in the computer vision community lately. Often in the form of advanced video encoders for activity recognition, *e.g.* [2, 7, 31, 33, 37]. However, it is well known that actions may also be recognized by (static) appearance cues in the scene [18], or from characteristic objects [20]. Hence, it is hard to assess the contribution of modeling time using this task. Rather than evaluating time implicitly as part of proxy tasks like activity recognition, we prefer an explicit analysis of video time.

We are encouraged by two recent meta-analysis studies that assess video time as well. In [32] Sigurdsson *et al.* examine datasets, evaluation metrics, algorithms, and potential future directions in the context of action classification. They find that video encoders have to develop a temporal understanding to differentiate temporally similar but semantically different video snippets. Very recently Huang *et al.* [16] measured the effect of reducing motion in C3D video encoders during an ablation analysis on action classification datasets. Instead of removing motion in an action classification task, we prefer to evaluate the effect of video time in encoders like C3D, LSTM and our own proposal, on three time-aware tasks.

This paper makes three contributions to the meta-analysis of time in video. Inspired by three properties of video time, we first present three time-aware video recognition tasks

---

[1]Code is available at https://aghodrati.github.io/videotime.html

allowing to evaluate the temporal modeling abilities of video encoders. Second, we categorize video encoders into two general families. From their analysis, we derive a new video time encoder, specifically designed to capture the temporal characteristics of video. Third, we evaluate our video encoders, C3D and LSTM with respect to their temporal modeling abilities on the three time-aware video recognition tasks and discuss our findings.

2 Related Work

Time in video is an ambiguous concept which is hard to be quantified. [27, 46] strive to observe the time signal within video frames by distinguishing natural ordered frames from reversed frames. Isola *et al*. [19] see this "arrow of time" by discovering the transformations that objects undergo over time. Zhou *et al*. [50] propose the task of predicting the correct temporal ordering of two video snippets. We also consider arrow of time prediction to quantify the temporal asymmetry, but rather than aiming to find the best way to solve the task, we evaluate how capable modern video encoders are in addressing this task.

We also focus on predicting future frames, as being a task that directly connects to temporal continuity. Recently, predicting the future has received significant attention in computer vision. Different forms of future prediction have been proposed, for activities [5, 14, 30, 48], human trajectories [28, 42], body poses [9, 43], visual representations [41], or even to generate the pixel-level reconstructions of future frames [24, 25, 29, 39, 40, 47]. However, generating future pixels is ill-posed: there exist infinite possible futures and generating pixels is a separate machine learning challenge [11, 13, 23]. We, therefore, recast the future frame prediction task as a forward-frame retrieval problem where the goal is to select the correct future frame out of C possible choices. Clearly, casting future frame prediction as retrieval cannot be deployed in practice, since in practice we cannot access future frames. However, it allows for a clear, well-understood and consistent evaluation framework.

Most works for encoding temporal relations are evaluated on action recognition tasks [7, 8, 10, 33, 37, 44, 45]. A video is encoded by learning either the frame order [8], short-term motion patterns [33], frames dependencies [7, 31, 36], spatio-temporal representation [2, 21, 37] or fully connected layers [49]. However, often, recognizing an action can be attributed to several factors, besides temporal modeling, such as the scene type, the appearance of the actors, particular poses, and so on. Alternatively, actions can be defined procedurally, as an ordered set of sub-actions. The advantage is that procedures effectively determine a cause and effect, and, therefore, are temporally causal. For instance, [12] defines actions as templates like *taking something from somewhere* or *putting something next to something*. In such a setting, temporal modeling of detailed sub-actions is required for successful recognition. Thus, we propose to evaluate temporal causality on action templates instead of standard action classes.

3 Properties of Video Time

In this section, we describe three properties of video time and based on each, we formulate a task able to quantify the associated property.

Property I: Temporal Asymmetry. A first important property of time is its asymmetric nature, namely the fact that there is a clear distinction between the forward and the backward
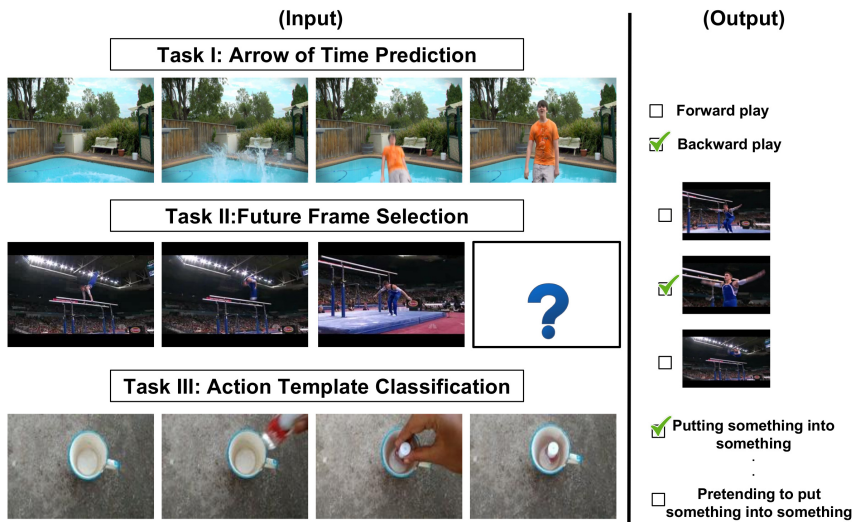
Figure 1: Inspired by three properties of time, detailed in Section 3, we present three time-aware video tasks that allow to evaluate video encoders in light of their temporal modeling abilities.

arrow of time. A question, therefore, is to what extent time asymmetry is also observable and quantifiable by video encoders. Similarly to Pickup *et al.* [27], we adopt the **arrow of time prediction** task, a binary classification task, which aims to predict whether a video is played forwards or backwards. Different from [27], we focus on exploring how capable state-of-the-art video encoders are in addressing this task, instead of finding time-sensitive low-level features to solve this task.

Property II: Temporal Continuity. A second important property of time is continuity. Because of temporal continuity, future observations are expected to be a smooth continuation of past observations, up to a given temporal scale. A time-aware video encoder should be able to predict future frames by its ability to model temporal continuity. In the second task we measure how capable video encoders are in predicting future frames. We introduce the task of **future frame selection**, where we assume we have access to the correct future frame at $T - \tau$ seconds from the currently observed frame x_τ. The goal is to predict the correct future frame among other incorrect ones. By careful sampling of the incorrect future frames we can control the difficulty of the evaluation. This setup allows expressing future frame prediction as a well-defined classification task with a well-defined evaluation procedure.

Property III: Temporal Causality. A third important aspect of time in videos relates to causality. An interesting observation for the traditional action classification task in YouTube-based UCF101 dataset [35] is that it does not necessarily need causal reasoning, meaning that any permutation of the frame order will not change the action classification performance much [49]. For instance, recognizing *playing basketball* is less determined by the precise order of the video frames, and mostly by the presence of the basketball court. However, a good video encoder should capture such causal relations. To formulate this property, we fo-

cus on ***action template classification***. In this setup, classes are formulated as template-based textual descriptions [12], where an action is characterized by temporal relations of detailed intuitive physics. An example of such a template-based action class is *Putting something into something*, which can only be successfully discriminated from *Pretending to put something into something* if the temporal relations are accurately modeled.

We summarize the tasks able to evaluate the properties of video time in Fig 1.

4 Video Time Encoders

We are interested in a video encoder $f(x_1, \ldots, x_t)$ capable of modeling the time signal present in frames $x_1, \ldots, x_t$. To structure our analysis, we first group existing video encoders into two general families namely Sequential Video Time Encoders and Hierarchical Video Time Encoders, and then we describe our proposed video time encoder.

Sequential Video Time Encoders. This family of video encoders views temporal data as infinite time series $x_1, x_2, \ldots, x_t$, where each time step t is associated to a state h_t. Formally, in each step these encoders model the transition between preceding states to the current state by $h_t \propto f(x_1, \ldots, x_t, h_1, \ldots, h_{t-1}; \theta)$, where $f(.)$ is a function parametrized by θ. Recurrent neural networks and variants (LSTM [15], GRU [9]), as well as Hidden Markov Models [1] all belong to the sequential family of video encoders. Particularly, RNNs fit to the video domain due to their long-range temporal recursion. In their simplest form, recurrent neural networks model the hidden state at time step t by

$$h_t = f(x_t, h_{t-1}; \theta) = \sigma_h(W_h \cdot x_t + U_h \cdot h_{t-1} + b_h), \tag{1}$$

where h_{t-1} summarizes all the information about the past behaviour of the frame sequence. The function σ_h is a non-linearity and the parameters $\theta = \{W., U., b.\}$ are shared and learned with backpropagation through time. As such, a single parameter set θ suffices to capture the temporal relations between subsequent variables x_i and x_{i+1}. A special variant of RNNs are LSTMs, which have shown to be particularly strong in learning long-term dependencies and, thus, have been widely used in the vision community for modeling frame sequences [7, 36]. Hence, as a representative sequential encoder, we choose LSTMs in this work.

Hierarchical Video Time Encoders. This family of encoders views video as a hierarchy of temporal finite "patches" of horizon T, where each patch is defined as $H^{(l)} = [h_i^{(l)}, \ldots, h_{i+T}^{(l)}]$. Formally, in level l of the hierarchy, $h^{(l)}$ is defined as

$$h^{(l)} \propto \sigma_h\left(h^{(l-1)} * W^{(l)} + b^{(l)}\right), \tag{2}$$

where $h_i^{(0)} = x_i$ and $*$ denotes the convolution operator. This is similar to the way convolutional neural networks conceive an image as a structure of image patches. Like convolutional neural networks, Hierarchical Video Time Encoders define successive layers (l) of non-linearities. Specifically, at layer (l) a set of temporal templates $(W^{(l)}, b^{(l)})$ (layer parameters) are learned, based on co-occurrence patterns present in the input.Similar to convolutional neural networks for images, different layers have different parameters.

C3D [21, 37] and its variants (like i3D [2] and LTC [38]) belong to the temporal hierarchy family of models, where spatial and temporal filters are learned jointly using 3D

convolutions. As a representative hierarchical encoder, we choose C3D in this paper, which has been widely used in the literature, *e.g.* [21, 37].

Discussion. The main difference between Sequential Video Time Encoders and Hierarchical Video Time Encoders is in their handling of the temporal sequence. Sequential encoders, like LSTMs, view temporal data as time series, and embed all information from the past into a state variable. They then learn a transition function from the current to the next state. By construction, the model converges in the limit to a single, static transition matrix shared between all states in the chain. For video frame sequences, this transition matrix must then be able to model the temporal patterns in the input sequences. Under the noisy and non-stationary reality of video content, learning the transition function is hard. However, sequential encoders focus on learning transitions from one temporal state to the other. Thus they are able, in theory, to model conditional dependencies through time.

In contrast, hierarchical encoders rely on convolutions, which are equivalent to correlations after rotation by π. Thus, hierarchical video time encoders focus on the correlations between input elements, rather than their conditional temporal dependencies. Consequently, hierarchical encoders are well-suited either when temporal correlations suffice for the task, or when correlations coincide with causations for a given task. By viewing sequences as "finite temporal patches" in temporal hierarchies, hierarchical encoders can learn specialized filters to recognize specific temporal patterns of different abstractions. Because they introduce many more learnable filter parameters, however, they are more prone to overfitting.

Proposed Time-Aligned DenseNet. To overcome the limitations of sequential and hierarchical video encoders in time-aware video recognition tasks, we propose the Time-Aligned DenseNet, a neural architecture inspired by DenseNet [17]. Like DenseNet, it has densely connected patterns, *i.e.* each layer is connected to all its preceding layers. Unlike DenseNet, the layers are aligned along the temporal dimension rather than the spatial one. Formally, we describe the Time-Aligned DenseNet by

$$h_t \quad = \quad f(x_t, h_1, \ldots h_{t-1}; \theta_t). \tag{3}$$

The proposed model can be seen as a hybrid between sequential and hierarchical encoders. Similar to RNNs, the Time-Aligned DenseNet also views video frames as time series. Similar to C3D, and unlike RNNs, however, it does not share parameters through time.

Because of the non-shared parameterization Time-Aligned DenseNet enjoys several benefits. First, the encoder has a greater flexibility in modeling temporal transitions compared to LSTM. Second, there is no recursion and standard backpropagation suffices, instead of backpropagation through time. Thus, the encoder is able to exploit all the modern deep learning advances for a better training. This is important, as the shared parameterization can easily lead to chaotic behavior in RNNs [26], forcing RNNs to a more conservative optimization regime. Last, Time-Aligned DenseNet has explicit access to *all* previous hidden states, unlike RNNs that have only implicit access.

The general architecture of the Time-Aligned DenseNet is shown in Fig. 2. Although we have no restrictions on how to represent the state variables h_t, since we focus on video we opt for convolutional feature maps. We start from a sequence of individual frames from a video, encoded using a convolutional neural network, as LSTM. For each time step in the video, we have an associated layer in the network. Specifically, we implement each layer per
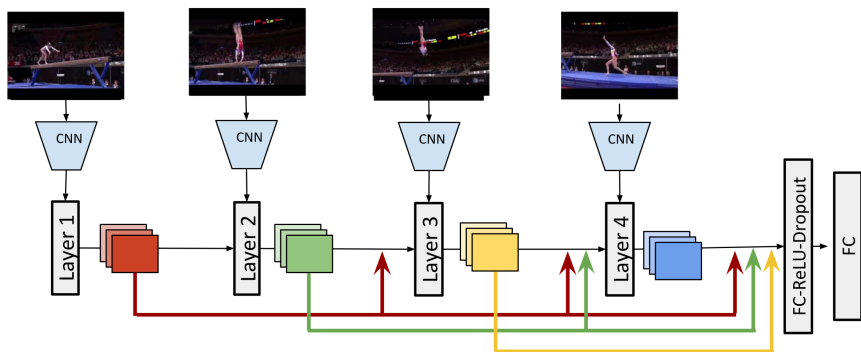
Figure 2: Proposed Time-Aligned DenseNet architecture. Individual frames are encoded using a CNN and then fed to an encoder where each layer represents a time-step. Inspired by [17], each layer is connected to all its preceding layers. At time step T, the encoder outputs $K \times T$ feature maps as representation of the video.

time step as a DenseNet block [17]:

$$f^a = \text{Concat}([h_1, ..., h_{t-1}, x_t])$$
$$h_t = \text{DropOut}(\text{Conv2D}(\text{ReLU}(\text{BatchNorm}(f^a, \theta_t^{bn})); \theta_t^{conv}, K)), \quad (4)$$

where K is the number of output feature maps dedicated to layer t. It is worth noting that such a design makes our architecture fully convolutional. One can see the per time step feature maps h_t as the state of the model at time t. We follow a dense connectivity pattern as [17], where each layer receives feature-maps from all preceding time-steps, generates a set of feature-maps for its own time-step and passes them on to all subsequent layers. These layers are designed to be "temporally causal", in the sense that each layer has access only to past (backward) time steps, not forward steps. After T time steps, the final encoding is a concatenation of all the states $[h_1, \cdots h_T]$ which contains $K \times T$ feature maps in total.

5 Evaluating Video Time

5.1 Arrow of Time Prediction

Datasets. For the first task we focus on two datasets. First, we use the dataset proposed in [7] for evaluating arrow of time recognition, which we refer to as ***Pickup-Arrow***. The dataset contains 155 forward videos and 25 reverse videos, divided in three equal splits. We report the mean accuracy over all three splits. The second dataset is the more complex UCF101 [35], designed for action recognition. We ignore the action labels and use the videos in forward and backward order, both for training and test. We refer to this setup as ***UCF101-Arrow***. We use split-1 of the dataset and again, report the mean accuracy.

Implementation details. For C3D, we use the architecture proposed by [37], pretrained on Sports1M [22], and finetuned on UCF101. For LSTM and Time-Aligned DenseNet we use a VGG16 [34] as the base network, pretrained on ImageNet [6], and then finetuned on

	UCF101-ARROW	PICKUP-ARROW
CHANCE	50.0	50.0
LSTM	67.5	80.0
C3D	57.1	57.1
TIME-ALIGNED DENSENET	**79.4**	**83.3**
PICKUP *et al.* [27]	-	80.6

Table 1: Arrow of time prediction results. Sequential encoders like LSTM and ours are better suited than C3D for this task.

UCF101. The single-layer LSTM uses as input the $fc7$ activations with 512 hidden units, while Time-Aligned DenseNet uses as input the last convolutional layer activations (before max-pooling), with the number of output feature maps in each step set to $K = 12$. The output passes through a final batch normalization, ReLU, max-pooling and fully connected layer with a 2-dim output. In terms of number of parameters, LSTM has 143M parameters (134M for the base VGG16 network, 9M for the LSTM units). Time-Aligned DenseNet has 69M parameters (68M for the base VGG16 network, 1M for the time-step layers). The difference in base network parameters between LSTM and Time-Aligned DenseNet is because LSTM ends with fully connected layers. C3D has in total 78M parameters. All video encoders output a binary classification, indicating whether the video is played forward or backward. The learning rate is set to 0.001 for LSTM and ours and 0.0001 for C3D as the learning was unstable with bigger learning rates. Momentum and weight decay is set to 0.9 and $5e - 4$ for all the encoders. All encoders receive the same 16 frames as input. During training we sample the frames from the whole video using a multinomial distribution, such that encoders see multiple sets of frames of a video. During testing we sample frames from a uniform distribution in order to assure the same input is used for all the encoders.

Results. We report results in Table 1 and plot in Fig. 3 the t-SNE of all three model embeddings, collected from the last layer. On both datasets, the sequential encoder performs better than its hierarchical counterpart, indicating the hierarchical encoders are less good in distinguishing asymmetric temporal patterns. As explained in Section 4, C3D relies on learning correlations between input frames and, therefore, fails to capture conditional dependencies between ordered inputs. From the t-SNE plots we observe that the embeddings from Time-Aligned DenseNet are better clustered, thus allowing for more accurate prediction of the arrow of time. In the Pickup-Arrow dataset, LSTM is able to get close to the accuracy of Time-Aligned DenseNet, while in UCF101-Arrow the gap widens. Despite the rapid progress in deep learning for video understanding, low-level visual features, hand-engineered for the task [27], with an SVM classifier are highly competitive. It appears there is much room for improvement of deep learning of video time representations.

It is noteworthy that the performance is not uniform across all videos. When inspecting videos from particular UCF101 classes, see Fig. 4, we observe that classes that one could claim are temporally causal (*e.g.*, *Billiards*, *Still rings* and *Cliff diving*) appear to be easier for all the methods. That said, all encoders have difficulty for a big portion of the classes (79, 86 and 39 classes have less than 75% accuracy for LSTM, C3D and Time-Aligned DenseNet respectively). Surprisingly, for a few classes, like *Punch*, *Drumming* and *Military parade*, LSTM and C3D even report below chance accuracies.

We conclude that even in ordinary videos, like UCF101, a temporal signal between suc-
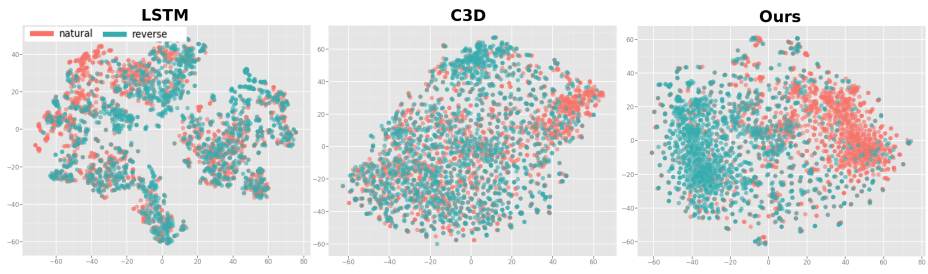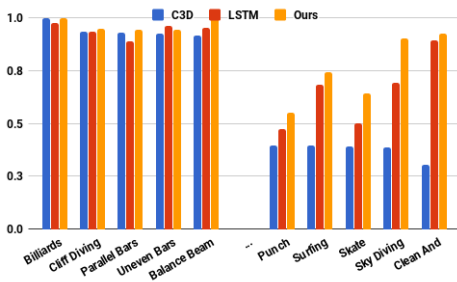
Figure 3: t-SNE visualization of features from the last layer before the classification layer. Note the embeddings from our encoder are better clustered.

Figure 4: Accuracy of C3D, LSTM and ours in predicting the arrow of time for 5 best and worst performing classes in UCF101, sorted by C3D accuracies. Temporally causal actions like *Billiards* appear to be easier than repetitive actions like *Punch*. The class information is only used for illustration purposes.

cessive video states exists. Sequential video encoders including Time-Aligned DenseNet appear to be better for the task of the arrow of time, because they model the temporal conditional dependencies between visual states, instead of their correlations.

5.2 Future Frame Selection

Dataset. For the second task we focus on UCF101, because it is larger than the Pickup dataset. In UCF101 among others there exist action classes that are either static, like *playing flute*, or periodic, like *hula hoop*. Clearly, in such videos temporal continuity is either trivial or purposeless, as in both cases the future frames will likely look nearly identical to (some of the) past frames. As having videos with a recognizable arrow of time also coincides well with videos where temporal continuity can be well evaluated, loosely following [[7]] we choose 24 classes with a distinguishable arrow of time. We coin this dataset ***UCF24-FutureFrame***.

Implementation details. We train the models for this task using the same setup as Section 5.1. During training, this task is formulated as a binary classification problem, where the encoder must predict whether a given frame is the correct future frame or not. During testing, the encoder must select which frame is the correct future frame out of $C = 5$ possible options. As a future frame we define the last frame x_T of the video. To increase the difficulty of the task, the incorrect frames are uniformly sampled from, $x_1, ..., x_{0.8T}$. To make sure the video encoders do not overfit to a specific temporal scale, we consider several τ during training and testing to define the last observed frames and report the mean accuracies[2].

---

[2]We pick τ from $\{0.4, 0.8, 1.3, 1.7, 2.5, 3.3, 4.2, 5.0, 5.8, 6.7, 7.5, 8.3\}$ sec corresponding to the 5-th, 10-th, ...

	UCF24-FUTUREFRAME
CHANCE	25.0
LSTM	46.9
C3D	51.9
TIME-ALIGNED DENSENET	**56.3**
FRAME SIMILARITY	24.2

Table 2: Performance on future frame selection. Our model outperforms others due to the unshared convolutional filters which retain both spatial and temporal correlations.

Results. We show the results in Table 2. Future frame selection is harder than modeling the arrow of time, as it requires not only modeling of the arrow of time but also modeling temporal continuity. We observe that C3D performs better than LSTM. We attribute this to the fact that this task is spatio-temporal and C3D learns spatio-temporal filters, which can specialize un recognizing the most similar future frame. Time-Aligned DenseNet outperforms both LSTM and C3D. Time-Aligned DenseNet learns the temporal conditional dependencies between the spatial embeddings, instead of temporal correlations like C3D. We also compare the encoders with a frame similarity baseline. We compute the cosine similarity between representations (normalized last convolutional layer of VGG16) of the last observed frame and the given future choices; the one with the highest score is considered as the correct answer. All methods perform better than chance and the frame similarity baselines. We conclude that for the temporal modeling task of future frame selection learning both spatial and temporal dependencies are necessary.

5.3 Action Template Classification

Dataset. For the third task we focus on the Something-Something dataset [12], where the action class labels are template-based textual descriptions like *Dropping [something] into [something]*. The Something-Something dataset is crowd-sourced and contains $86,017$ training and $11,522$ validation videos of length $2-6$ sec, across 174 action categories in total. The action classes represent physical actions instead of high-level action semantics (see Fig. 1), such as *Pouring something into something* or *Pouring something until it overflows*. As classes cannot be recognized by indirect cues like object or scene type, the temporal reasoning plays an important role.

Implementation details. For this experiment we follow the setup proposed by [45]. As base network we choose Inception with Batch Normalization (BN-Inception), pretrained on ImageNet. The input feature for LSTM are the activations from the global averaging pooling layer. Since Time-Aligned DenseNet accepts convolutional feature maps, we use the activations from the layer before average pooling, after adding an extra convolutional layer to reduce the number of feature maps from 1024 to 256 for better efficiency. We set the number of time-steps to 4 for all the models, except for C3D, which is pretrained with 16 frames. For all models, but C3D, we use a fully connected layer with 512 units before the classification layer.

frame in the video with fps set to 12 .

	SOMETHING-SOMETHING	
	PREC@1	PREC@5
CHANCE	0.5	3.0
LSTM	15.7	39.5
C3D	28.2	56.3
TIME-ALIGNED DENSENET	**30.4**	**59.3**
WANG *et al.* [45]	16.0	41.1

Table 3: Performance on Task III. C3D and ours are able to learn the temporal dependencies while LSTM can not. Temporal modeling is crucial for this task.

Results. We show results in Table 3. Similar to the second task, LSTM cannot easily learn the temporal conditional dependencies required for the recognition of the physical and time-specific actions. C3D is better than LSTM. Again, Time-Aligned DenseNet outperforms both C3D and LSTM. Time-Aligned DenseNet also outperforms the Temporal Segmentation Network [45] (TSN) by a significant margin. The reason is after splitting a video into segments, TSN discards temporal structure by aggregating video segment representations via average pooling. We also compare with the concurrent work of Zhou *et al.* [49], reporting 29.8 and 58.2 prec@1 and prec@5 respectively, slightly less than Time-Aligned DenseNet for this setting. Their model also aims at learning temporal relations between ordered frames (sets of two frames, three frames, etc.) via fully connected layers, and thus the model cannot be easily adapted for higher order frame relations.

We conclude that for template-based action classification, modelling temporal conditional dependencies is important. Similar to the second task, the model should be able to parameterize the temporal conditional dependencies per time step freely. If not, sharing parameters through time leads to worse results than not modeling the dependencies at all.

6 Conclusions

In this work, we investigated and evaluated the manifestation of time in video data. Our meta analysis quantifies video time with respect to temporal asymmetry, continuity and causality by three corresponding tasks. Moreover, we proposed a new video time encoder and provided in-depth analysis of LSTM, C3D and our proposed encoder in this setting. We observed LSTM is better than C3D in handling our temporal asymmetry task. As the need for joint modeling of spatio-temporal data increases in our tasks measuring temporal continuity and causality, C3D is outperforming LSTM. Our proposed Time-Aligned DenseNet consistently outperforms both C3D and LSTM on all three tasks. An important factor appears to be the unshared parameterization of our proposed encoder in modeling temporal transitions. We believe that more detailed understanding of time holds promise for the next iteration of encoders.

References

[1] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. *arXiv preprint arXiv:1804.02748*, 2018.

[5] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *ECCV*, 2016.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[8] Basura Fernando, Efstratios Gavves, Oramas Mogrovejo, José Antonio, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

[9] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, 2015.

[10] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *PAMI*, 35(11):2782–2795, 2013.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[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*, 2017.

[13] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[14] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. *IJCV*, 107 (2):191–202, 2014.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *CVPR*, 2018.

[17] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017.

[18] Nazli Ikizler-Cinbis and Stan Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*, 2010.

[19] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015.

[20] Mihir Jain, Jan C. van Gemert, and Cees G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015.

[21] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013.

[22] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

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

[24] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *ICLR*, 2017.

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

[26] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

[27] Lyndsey C Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T Freeman. Seeing the arrow of time. In *CVPR*, 2014.

[28] Silvia L Pintea, Jan C van Gemert, and Arnold WM Smeulders. Déja vu:motion prediction in static images. In *ECCV*, 2014.

[29] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[30] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011.

[31] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017.

[32] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017.

[33] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

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

[35] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[36] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[38] Gul Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *PAMI*, 2017.

[39] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. *arXiv preprint arXiv:1704.05831*, 2017.

[40] Carl Vondrick and Antonio Torralba. Generating the future with adversarial transformers. In *CVPR*, 2017.

[41] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016.

[42] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016.

[43] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017.

[44] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[45] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[46] Donglai Wei, Jospeh Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018.

[47] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*, 2016.

[48] Kuo-Hao Zeng, William B Shen, De-An Huang, Min Sun, and Juan Carlos Niebles. Visual forecasting by imitating dynamics in natural sequences. In *ICCV*, 2017.

[49] Bolei Zhou, Alex Andonian, and Antonio Torralba. Temporal relational reasoning in videos. *arXiv preprint arXiv:1711.08496*, 2017.

[50] Yipin Zhou and Tamara L Berg. Temporal perception and prediction in ego-centric video. In *ICCV*, 2015.