Recur, Attend or Convolve? On Whether Temporal Modeling Matters for Cross-Domain Robustness in Action Recognition

Sofia Broomé[1]   Ernest Pokropek[1]   Boyu Li[1]   Hedvig Kjellström[1,2]

[1] KTH, Sweden `sbroome,pokropek,boyul,hedvig@kth.se`   [2] Silo AI, Sweden

Abstract

Most action recognition models today are highly parameterized, and evaluated on datasets with appearance-wise distinct classes. It has also been shown that 2D Convolutional Neural Networks (CNNs) tend to be biased toward texture rather than shape in still image recognition tasks [19], in contrast to humans. Taken together, this raises suspicion that large video models partly learn spurious spatial texture correlations rather than to track relevant shapes over time to infer generalizable semantics from their movement. A natural way to avoid parameter explosion when learning visual patterns over time is to make use of recurrence. Biological vision consists of abundant recurrent circuitry, and is superior to computer vision in terms of domain shift generalization. In this article, we empirically study whether the choice of low-level temporal modeling has consequences for texture bias and cross-domain robustness. In order to enable a light-weight and systematic assessment of the ability to capture temporal structure, not revealed from single frames, we provide the Temporal Shape (TS) dataset, as well as modified domains of Diving48 allowing for the investigation of spatial texture bias in video models. The combined results of our experiments indicate that sound physical inductive bias such as recurrence in temporal modeling may be advantageous when robustness to domain shift is important for the task.

1. Introduction

One of the most fundamental questions when it comes to video understanding is how to model the dependency between frames in such a way that temporal relationships relevant to the activity in the video can be learned. A robust action recognition system should be able to figure out how frames relate to each other, and which shapes and objects have changed or persisted over time. With this knowledge, it can start to infer relationships at a higher level, such as object-object or agent-object relationships.

Three principally different approaches to frame dependency are 3D convolutions, self-attention and recurrence.

These methods model the world (the visual sequence) in principally different manners: linearly, non-linearly, and non-linearly with a time-causal direction, meaning that they each use different inductive biases for the temporal modeling (**Fig. 1 of the supplemental**). In spite of its essentiality, the frame dependency question has almost disappeared from action recognition articles, possibly in the race to improve on the classification benchmarks. Emphasis is instead placed on other aspects of deep video models, such as advanced architectural superstructures, regularization or training schemes. A shift toward attention-based video models has recently taken place, but without a discussion of the physical interpretation of its underlying temporal model.

Humans are still significantly stronger at generalization than artificial neural networks in vision tasks [20, 46]. Recurrent models are critical in the only visual system that has been 'solved' to date – biological vision [1, 10, 12, 13, 15, 32, 34, 44, 52]. Based on the observation that feedback connections are abundant in biological but not computer vision [32, 56], in this article, we hypothesize that the lack of recurrence when learning spatiotemporal features may be one reason for this discrepancy. We therefore investigate the following research question empirically in extensive and systematic experiments: does the principally different mathematical natures of 3D convolutions, self-attention and recurrence affect cross-domain robustness in video models – and in particular, does recurrence bring about an advantage?

Video models lack robustness to domain shift [8, 61, 62], and it has been repeatedly shown [8, 25, 35, 60] that the datasets most frequently cited during the 2010s (UCF-101 [51], HMDB [33], Kinetics [29]) exhibit significant spatial biases. This is a plausible reason for the poor cross-domain robustness in action recognition, since overly relying on spatial rather than motion cues intuitively results in overfitting to one domain (e.g., certain backgrounds, viewpoints or similar actor appearances).

Contemporary state-of-the-art approaches to action recognition are predominantly either fully convolutional [5, 17, 18, 21, 60], combine convolutions with temporal sampling and fusion strategies [58, 59, 63], or, more recently, attention-based Video Transformers (VTs) [4, 23,
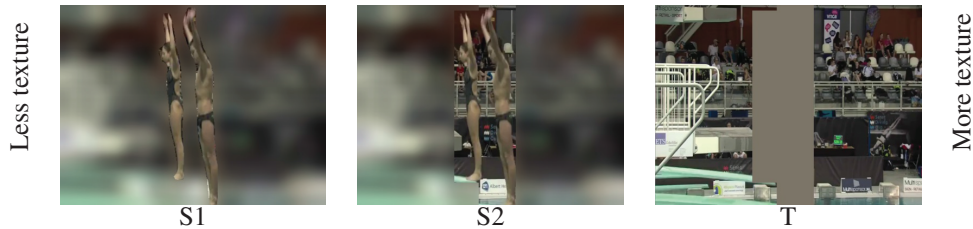
Figure 1. **Animated figure, displayed on click in Adobe Reader.** Example clip showing our three modified domains of Diving48 enabling the investigation of texture bias in video models: S1 (segmented divers over a blurred background), S2, (cropped bounding boxes of divers over a blurred background), and T (masked boxes of divers, and the original background).

31, 45, 53]. The sheer size of the models, typically more than 50M trainable parameters, gives them a strong capacity to learn in-domain patterns. As models grow larger, ever more resources are spent to train them. State-of-the-art models should display competitive benchmarking numbers on large-scale datasets, such as Kinetics-400 and Kinetics-600. It is questionable whether these benchmarks are suitable for temporal modeling, or rather for how large amounts of YouTube clips efficiently can be stored as weight representations. At the same time, the reciprocal dependency between the hardware and software of standard graphics processing units (GPUs), on the one hand, and models requiring massive parallel computation for their training, on the other hand, is becoming ever more intertwined [30, 41]. The question looms whether we have cornered ourselves in action recognition, in the expectancy to work on ever larger models, in industry as well as in academia.

Theoretical works [2, 39, 50] have indicated that overparametrization helps generalization, in that local minima of the loss landscape for such models often are thought to be global. These studies are made on held-out data, but never on data with significant domain shift, to the best of our knowledge.

Although less efficient to train on GPUs, recurrent video models have a more parameter-efficient approach per timestep, which may hinder over-reliance on texture cues, and promote learning the temporally relevant motion cues. The need to be economical with the use of trainable parameters, we hypothesize, creates incitement to learn better shape representations instead of texture representations. In turn, this allows for better generalization across datasets and in the wild. For contour detection, it was found that a model with recurrent dynamics was more sample-efficient and generalized better than a feed-forward model [37, 38].

The primary contributions of our paper are as follows:

- We present the first empirical results from systematic experiments on how the choice of frame dependency modeling in action recognition can affect cross-domain robustness.

- We introduce a lightweight dataset allowing for investigation of both temporal shape modeling ability

and domain generalization, called the Temporal Shape dataset.

- We provide the first discussion and experiments on shape vs. texture bias (following Geirhos et al. [19]) in deep video models.

- We make segmentation-based shape and texture versions of the Diving48 dataset public (as well as 303 instance-segmented frames), allowing studies on whether a video model has learned to rely more on (temporal) shape or on texture.

2. Related Work

Domain shift in action recognition. In [7, 61], cross-domain datasets are introduced to study methods for video domain adaptation. [7] proposes to align the temporal features where the domain shift is most notable, whereas [61] proposes to improve the generalizability of so-called local features instead of global features, and use a novel augmentation scheme. Strikingly, however, all experiments in [7, 61] are based on features extracted frame-by-frame, by a 2D ResNet [26], and aggregated after-the-fact, meaning that they in effect do not handle spatiotemporal features. Using frame-wise features saves large amounts of time and computation, but it avoids an essential aspect of video modeling. Different from the field of Domain adaptation, we are not proposing methods on top of base architectures to reduce domain shift, but rather study empirically which types of fundamental video models inherently seem to be more robust to it. In an important work by Yi et al. [62], benchmarks are introduced to study robustness against common video corruptions, evaluated for spatiotemporal attention- and convolution-based models. Different from our work, the domain shift is restricted to data corruptions rather than the same classification task in a new domain, and recurrent models are not evaluated.

Emphasis on temporality in action recognition. Many works emphasize the importance of temporal modeling, as the field of video understanding is growing, e.g., [14, 22, 40, 43, 47, 49, 60, 63]. [22] and [40] compare temporal model-

ing abilities between principally different architectures, but without explicitly investigating domain shift generalization.

[49] examines video architectures and datasets for human activity understanding on a number of qualitative attributes such as pose variability, brevity and density of the actions. [28] investigates how much the motion contributes to the classification performance of the C3D architecture [54]. Both [6] and [55] perform large-scale studies of the features of different variants of 2D and 3D CNNs in action recognition. Last, we are connected to [47], which discusses the risk that models with strong image modeling abilities may prioritize those cues over the temporal modeling cues. Reminiscent of the findings of [19], the authors of [47] find that inflated convolutions tend to learn classes better where motion is less important, and that generalization can be helped by training on more temporally focused data (in analogy to training on shape-based data in [19]). Different from our work, however, only fully convolutional models are studied and the focus is not on comparing models with fundamentally different approaches to frame dependency modeling.

3. Experiment design

In this section, we describe the experiment design for the two datasets: Temporal Shape and Diving48.

Main idea. In all experiments, we begin by training on a specific domain, and validating on a held-out dataset from the same domain. We save the model checkpoint which performed the best on the validation set, and then proceed to evaluate it on other domains that are different in some respects but share the same task. Following [7], the domain we train on will be referred to as the source, and the unseen domains that we evaluate on as the target. To measure cross-domain robustness, we define the **robustness ratio (rr.)** as the ratio between a model's accuracy on a target domain and its best validation accuracy on the source domain. When the target task corresponds to the source task, this number should ideally be close to one (higher is better). It can be noted that the rr. is a heuristic metric, which builds on the assumption that the performance on the in-domain validation set typically is higher than on other domains. If the performance on the validation set is poor to begin with, the rr. is less informative.

Method common to all experiments. In our study, we are purposefully comparing the basic functionality of models. No pre-training, dropout, or data augmentation is applied in our experiments, except for 50% chance of horizontal flipping of the clips on Diving48. Sequences are uniformly sub-sampled into equal length (a fixed input size is required for the input to both 3D CNNs and attention-based models). There are non-uniform frame sampling methods, which can be used as augmentation, or as informed priors

(e.g., the TimeSformer only samples the middle frames during inference in [4]); these are thus not used in our study, in order to study the bare bones of the models. Code related to neural networks was written in PyTorch [42] using Lightning [16]. Further implementation details and code can be found in the corresponding repositories (Temporal Shape experiments, Diving48 experiments and diver segmentation). The datasets are available for download on Harvard Dataverse and linked to from the repositories.

3.1. Models

We will compare ConvLSTMs, 3D CNNs and VTs, since these present three principally different temporal modeling approaches with varying types and degrees of inductive bias. As VT, we will use the TimeSformer [4], because it recently achieved state-of-the-art results on a number of action recognition benchmarks.

It is a challenging task to compare neural network models which have principally different architectures. In our work, we decided on controlling for three different factors: the performance on a particular dataset, the number of trainable parameter and the layer structure (i.e., the number and expressivity of hierarchical abstractions). The experiments were designed prior to running them, to keep the process as unbiased as possible. The experiments are further completely reproducible as they were run on five fixed random seeds throughout the study.

Convolutional LSTMs. The ConvLSTM [48] layer functions like an LSTM layer [27], but with matrix multiplication replaced with 2D convolutions. This crucially means that they allow for the input to maintain its spatial structure, contrary to classical recurrent layers which require a flattened input. Frame dependency is modeled using recurrence, which introduces non-linearities between timesteps. Further, time can only flow in the causal direction. A ConvLSTM video model, in this work, is a model fully based on these types of layers, with a classification head on top.

TimeSformer. The TimeSformer (hereon, TimeSf) [4] is a VT, relying entirely on self-attention mechanisms to model frame dependency. As in [11], each frame is first divided into patches, which are flattened. We use the TimeSformer-PyTorch library [57], mainly with the standard settings unless otherwise specified (divided space-time attention). Self-attention is applied both among the patches of one frame (spatial attention) and across patches located in the same positions across the temporal axis (temporal attention). Two variants are used in the TS experiments, with the number of heads $\mathcal{A}$ set to either 1 or 8 (TimeSf-1 and TimeSf-8). TimeSf-1 is closer to the ConvLSTM and 3D CNN in terms of parameter count, whereas TimeSf-8 is the standard setting. We note again that in order to study the

fundamental behavior of the models, we do not use pre-training, advanced data augmentation, nor averaging over multiple predictions. This results in a lower performance on Diving48 for TimeSf than its state-of-the-art results. In order to control for layer structure or number of parameters which requires architectural modifications, it is not possible to use a pre-trained checkpoint. It is well-known that VTs, or Vision Transformers (ViTs) in general, require a lot of training data due to their minimal inductive bias. We therefore stress that we are not questioning the overall performance of these models – a pre-trained version would have performed better on the Diving48 task than in our experiments, but we are investigating the fundamental behavior of models in our experiments, prior to more advanced or large-scale training schemes.

3D CNNs. In a 3D CNN, time is treated as space, and thus the input video as a volume, across which we convolve local filter volumes. Convolution is a linear operation, meaning that the order of frames that the 3D filter traverses does not matter. Instead, all non-linearities are applied hierarchically, between layers, which is how this model still can learn the arrow of time. Its layer structure is typically similar to a 2D CNN, including batch normalization and pooling. This is also the case for the instances used in our study.

3.2. Experiments on the Temporal Shape dataset

Our proposed TS dataset is a synthetically created dataset for classification of short clips showing either a square dot or a random MNIST digit tracing shapes with their trajectories over time (Fig. 2). The dataset has five different trajectory classes (i.e., *temporal shapes*): circle, line, arc, spiral and rectangle. The task is to recognize which class was drawn by the moving entity across the frames of the sequence. The spatial appearance of the moving object is not correlated with the class, and can thus not be employed in the recognition. In the 2Dot, 5Dot and MNIST domains, the background is black, and in MNIST-bg, the background contains white Perlin noise. The Perlin noise can be more or less fine-grained; scale is regulated by a random parameter $\sigma \in [1, 10]$. The dataset can be thought of as a heavily scaled-down version of an action template dataset, such as 20BN-Something-something-v2 [24], stripped of appearance cues.

The sequences consist of 20 64x64 frames, in grey scale. Each of the five classes has different amounts of possible variation in their states. The shapes can have varying starting positions, starting angles, direction, size and speeds. In the experiments, 4000 clips were used for training and 1000 for validation (model selection), and 500 clips for evaluation only. The classes are sampled so as to obtain class balance.

Since the dataset is small, we use lightweight models. We control for layer structure by letting the compared mod-

els have three layers each of analogous blocks with the same number of hidden units in each. One block for the ConvL-STM and 3D CNN consists of a model-specific layer, max pooling, followed by batch normalization. These two models used the same convolutional kernel sizes in all three layers (3×3). For the TimeSformer, we used one TimeSformer layer as one block, and the latent dimension for each attention head, D_h, as the number of hidden units, since these were similar in scale.

We run experiments for different numbers of hidden units per layer, $h \in \{2, 4, 6, 8, 10, 12, 16, 24, 32, 48\}$. For each of the ten experiments of varying model sizes, we train the models on the five-class task on the source domain for 100 epochs, with ten epochs of early stopping patience, repeated under five different random seeds set from the beginning of the study. For TimeSf-1 and TimeSf-8, the maximum number of epochs is 300 (100 for early stopping) because they demand more epochs to converge than the other two types of models, due to their minimal inductive bias. We then evaluate the best model checkpoint from the source domain on different target domains with the same classification task. Experiments were conducted in two 'directions', training on 2Dot and evaluating on the other domains, or training on MNIST-bg and evaluating on the other domains, since these represent two extremes on the continuum of less to more spatial noise.

Training on the TS data is light-weight compared to real video data, and runs fast (in the minutes-range, up to an hour, for the model sizes we evaluated) on one GPU card. We train with a batch size of 64 in all TS experiments.

3.3. Experiments on Diving48

Diving48 [35] is a well-known dataset for fine-grained and time-critical action recognition. It consists of 18k short clips with dives from 48 classes. Successfully classifying these dives requires temporal modeling, since one needs to keep track of the movements of the divers and their order. The dataset is varied appearance-wise, in terms of both backgrounds and viewpoints, which may contain unknown biases. The same competition sites can be present in both the training and test split, "to avoid biases for certain competitions", according to the authors [35]. Instead, in our view, this in fact increases the risk for bias, since the ability to recognize a dive at an unseen site is never tested. It would have been preferable to separate competition locations entirely between training and test set. Thus, even though the dataset presents a very challenging classification task from a temporal modeling perspective, it is likely not free from spatial biases (as will be demonstrated by our experiments).

Modified domains of Diving48. We always train on the original dataset, but evaluate our trained models on slightly modified domains of the original test set. We modify the test set into three new domains: two based on shape and

Figure 2. **The videos can be displayed on click in Adobe Reader.** Example clip showing the four domains of the TS dataset, for the class *circle*. In 2Dot and 5Dot, the circle is drawn with a square of width 2 and 5 pixels. In MNIST and MNIST-bg, the circle is drawn with a MNIST digit, w/ and w/o a Perlin noise background.

one based on texture (following Geirhos et al. [19], Fig. 1). To do this, we extend the concepts of shape and texture bias from [19] to the temporal domain in the following way. In the shape domains, we blur the background and only maintain the segmented diver(s) (S1), or the divers and their bounding boxes (S2). In the texture domain (T), we conversely mask bounding boxes where the diver(s) are in each frame, and only keep the background. The masked boxes are filled with the average Imagenet [9] pixel value, following [8]. The class evidence should lie only in the divers' movement; hence, the texture version should not contain any relevant signal, and the accuracy should ideally drop to random performance. In this way, we can study how different models drop in score when tested on the shape or texture domain, indicating both cross-domain robustness (for S1 and S2) and texture bias (for T).

Instance Segmentation of Diving48. The segmentation of divers in Diving48 is detailed in the supplemental. We release 303 manually labeled frames with instance segmentation (single or double dives), since off-the-shelf COCO-trained [36] networks fail at this task for the class Person, presumably because of the unusual shapes assumed in the air by the diver, or include people in the audience.

Training. Just as for TS, we deliberately avoid bells and whistles when training models on Diving48, to study their fundamental behavior. All three models are trained with the same SGD optimizer, cross-entropy loss, and a constant learning rate of 0.001. Each model is trained for 500 epochs maximally, with an early stopping patience of 30 epochs if the validation performance does not improve. The only data augmentation used is horizontal flipping of 50% probability for the entire clip. The models are trained using PyTorch Lightning's *ddp* parallelization scheme across eight A100 GPUs, with a batch size of 8 and a clip length of 32 uniformly sampled frames, at 224×224.

Given that the purpose of our experiments is *not* to optimize classification performance, we evaluate the models at different levels of performance, ranging from 30% to 50% accuracy. Some of the advanced state-of-the-art methods today, including pre-training and heavy data augmentation, obtain up to 80% performance on Diving48, but when the dataset was introduced in 2018, and standard video methods were tested off-the-shelf on it, the best result was 27% ac-

curacy [35]. Thus, the range of 30-50% is reasonably well-performing, and well above random (which is at 2.1%).

Experiments. We conduct three different kinds of experiments on Diving48, namely control for: layer structure and performance (a-c), performance of the best performing variants (d), and number of parameters and performance (e-h). ConvLSTM has four blocks of 128 hidden ConvLSTM units each (14.3M params.) in all experiments.

a-c. *Controlling for layer structure and performance.* In this experiment, we let the models have four layers, with $h = 128$ in each. We again treat D_h as the hidden unit analogy for TimeSf. We evaluate model checkpoints at different performance levels: 30%, 35%, and 38.3% accuracy. The last accuracy, was chosen because it was the limiting, highest performance by the 3D CNN in this experiment. Having the same layer structure gives rise to a varying number of parameters for each type of model. Here, the 3D CNN has 10.6M params., and TimeSf 85M.

d. *Controlling for performance only.* Here, we compare models at their best performance, after hyperparameter search. Since it was not possible to train TimeSf to a higher accuracy than 39.7% in all variants we tried, this experiment was only conducted with the 3D CNN and ConvLSTM. [1] The 3D CNN was an 11-layer VGG-style model (23.3M params.). The checkpoints used were both at exactly 50.07% validation accuracy.

e-h. *Controlling for number of parameters and performance.* Here, we have chosen models with a similar amount of trainable parameters, in this case 14M. To arrive at this number of parameters for TimeSf, its depth was reduced from 12 to 11, and D_h and D were halved, to 32 and 256, respectively, relative to the default model. The 3D CNN has six blocks with 128 units in each.

4. Results and discussion

Having presented the experimental design for both datasets, next, we discuss our empirical findings, first on TS, and then on Diving48 and its modified domains.

---

[1] A list of the variants we attempted with TimeSf is in the supplemental.

a) Training on 2Dot

Legend: 3D CNN, ConvLSTM, TimeSf-8, TimeSf-1
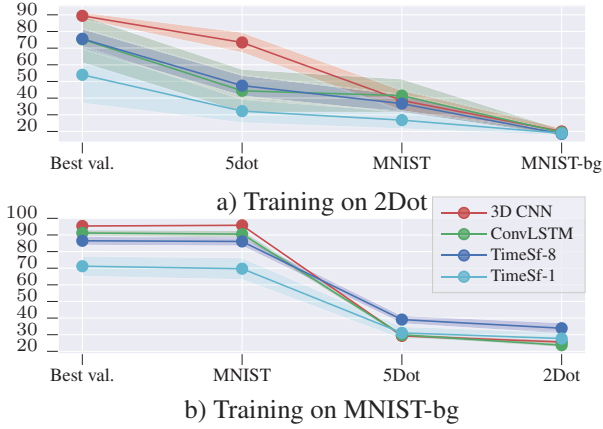
b) Training on MNIST-bg

Figure 3. Average results (% acc.) across ten trials with varying numbers of hidden units per layer, repeated five times each (thus in total, 50 runs per model). Plots corresponding to each model size can be found in the supplemental.

4.1. Temporal Shape

Condensed results: **TimeSf and ConvLSTM are more cross-domain robust than the 3D CNN in the absence of spatial texture bias.**

Training on 2Dot. Fig. 3a shows that although the 3D CNN generally obtains higher results on the source validation set and the nearby 5Dot domain, the ConvLSTM and TimeSf drop less compared to their original results when tested on MNIST (further from the source domain). ConvL-STM in fact outperforms the 3D CNN in absolute numbers on the MNIST domain. The inductive bias of a 3D CNN is highly local in space and time, which might impede learning of these temporal shapes. Generalization to the MNIST-bg domain proves too challenging for all three models.

Robustness ratio vs. model size. In Fig. 4, we have plotted the rr. for the three target domains when training on 2Dot. For 5Dot, the rr. for ConvLSTM decreases slightly with model size, whereas the 3D CNN and TimeSf, in contrast, increase the rr. with increased model size. For MNIST, which is further from the validation domain, the upward trend for the 3D CNN is broken, and less pronounced for TimeSf. For the most challenging domain, MNIST-bg, the rr. becomes very low for all three models with increased size. The trends in Figs. 4 a-c point to how a larger model size with promising performance in a nearby domain can potentially be an obstacle in domains that are further from the source for TimeSf and the 3D CNN.

Training on MNIST-bg. In this experiment, TimeSf-8 and TimeSf-1 were the most robust (Fig. 3 b). A VT is an excellent model when it comes to learning sparse, long-term dependencies in space and time. We hypothesize that this allowed TimeSf to to fully disregard the Perlin noise (which is highly stochastic and demanding to model) and learn the true temporal shapes, and that this, in turn, allowed it to be

unbiased in the other domains, since the training data was designed to exclude spatial bias. In real-world data, however, there will always be biases, and it is therefore best to construct models which inherently encode as little bias as possible, regardless of the training data.

4.2. Diving48: Sensitivity to shape and texture

Condensed results: **ConvLSTM exhibits less texture bias and is more cross-domain robust than TimeSf and 3D CNN.**

Model	S1/V ↑	S2/V ↑	T/V ↓	T/S1 ↓
3D CNN	**0.255** ± 0.026	0.260 ±0.034	0.221 ±0.028	0.878 ±0.15
ConvLSTM	0.230 ±0.028	**0.266** ± 0.026	**0.185** ± 0.042	**0.807** ± 0.16
TimeSformer	0.175 ±0.028	0.176 ±0.026	0.190 ±0.037	1.10 ±0.17

Table 1. Average results for experiments a-h.

Table 1 shows the average results for the Diving48 experiments. We note that ConvLSTM drops the most for T, both relative to the validation (T/V) and to the S1 (T/S1) accuracies. ConvLSTM is also most robust to the S2 domain, whereas the 3D CNN is most robust to the S1 domain.

Model	S1/V ↑	S2/V ↑	T/V ↓	T/S1 ↓
3D CNN	**0.253**	0.245	0.257	1.01
ConvLSTM	0.247	**0.281**	**0.238**	**0.965**
TimeSformer	0.198	0.203	0.250	1.27

Table 2. Experiment a: 4x128, 30% validation accuracy.

Experiments a-d. In experiments a-c (Fig. 5), where we vary the validation accuracy on the source domain between 30% and 38.3%, both TimeSf and the 3D CNN perform better on T than on S1 and S2, even if only the two latter contain class evidence. This suggests that spatial bias is indeed present in Diving48, and that these models are more prone to encode it than ConvLSTM. Tables 2-4 show that T/S1 > 1 for these two models, also visible in Fig. 5 a-c. [2]

Model	S1/V ↑	S2/V ↑	T/V ↓	T/S1 ↓
3D CNN	**0.230**	0.233	0.243	1.06
ConvLSTM	0.217	**0.271**	**0.199**	**0.919**
TimeSformer	0.154	0.152	0.190*	1.24

Table 3. Results for experiment b: 4x128, 35% validation accuracy. * when a low T/V result is not accompanied by T/S1 < 1.

Model	S1/V ↑	S2/V ↑	T/V ↓	T/S1 ↓
3D CNN	0.219	0.219	0.237	1.09
ConvLSTM	**0.279**	**0.308**	**0.187**	**0.669**
TimeSformer	0.155	0.155	0.175*	1.13

Table 4. Results for experiment c: 4x128, 38.3% validation accuracy. * when a low T/V result is not accompanied by T/S1 < 1.

In contrast, ConvLSTM clearly drops for T. TimeSf is large here, at 85M params., whereas the 3D CNN is interestingly quite small at 10.6M params. This suggests that

---

[2]In Tables 3-4, TimeSf drops the most for T relative to the validation set (T/V). This can be explained by its overall large drops, rather than being robust to texture bias, most clearly visible in Fig. 5 a-c. For T/V to be a meaningful metric, T/S1 should be < 1. Therefore, we have put asterisks on the lowest T/V results which are not accompanied by T/S1 < 1.
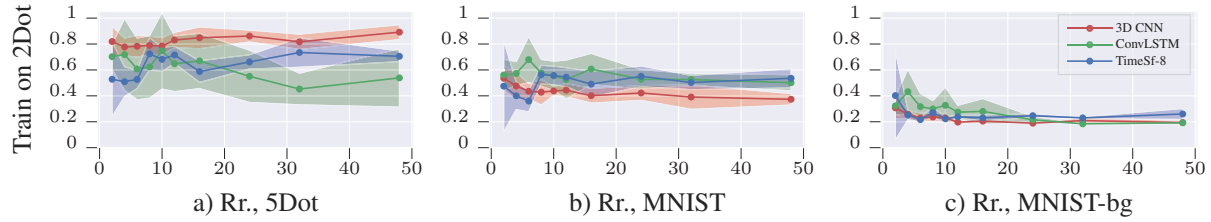
Figure 4. Robustness ratio (rr.) (↑) when training on 2Dot, vs. number of hidden units per layer. The target domain is progressively further away from the source in subplots a-c. TimeSf-1 is excluded here due to its near random validation accuracy for small model sizes.
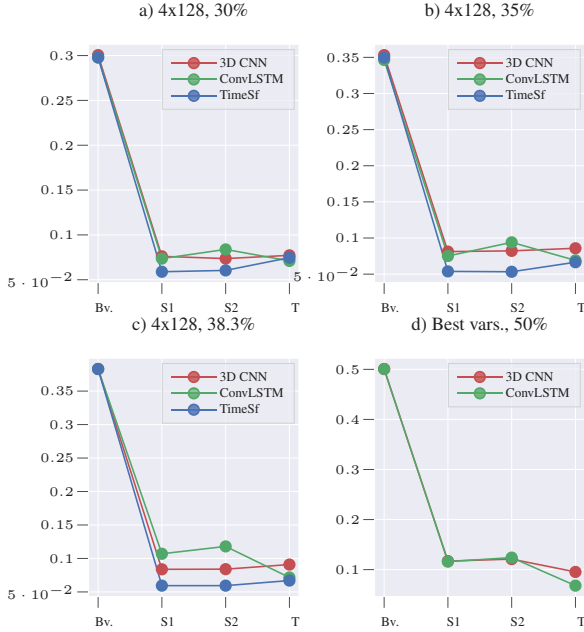

Figure 5. Diving48 accuracy drops, from source to target, experiments a-d. Bv. is best result in the validation domain. Note how ConvLSTM drops for T, in contrast to the 3D CNN and TimeSf.

not only the parameter count causes susceptibility for over-fitting, but that there may be innate tendencies to overfitting in the choice of spatiotemporal modeling. A recurrent model necessarily takes each timestep into account as it traverses the sequence in the time-causal direction, since each timestep is non-linearly registered in the hidden state. We hypothesize that this enables it to register motion changes over time more in detail, and count these as salient, when that is the case (as it should be for Diving48).

Model	S1/V ↑	S2/V ↑	T/V ↓	T/S1 ↓
3D CNN	**0.234**	0.242	0.191	0.815
ConvLSTM	0.232	**0.248**	**0.136**	**0.586**

Table 5. Experiment d: best variants, 50.07% val. accuracy.

In experiment d, where we compare a ConvLSTM and a 3D CNN at 50.07% validation accuracy – the best results on Diving48, the 3D CNN does not longer improve on the texture dataset relative to S1 and S2, but the drop on T is

Model	S1		S2		T	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ConvLSTM	0.4	1.0	0.4	1.0	0.0	0.6
3D CNN	0.4	0.8	0.4	0.6	0.0	0.8
TimeSf	0.4	0.4	0.2	0.4	0.0	0.2

Table 6. Qualitative example with predictions on five random clips from class **34**, made by the model instances from experiment c.

markedly larger for ConvLSTM (Table 5).

Qualitative examples and diving attributes. Table 6 shows a breakdown of the models' predictions on five randomly selected clips from a randomly chosen class (34). The models instances used here are from experiment c (38.3% acc.). Top-1 acc. for these five clips being equal for all models at 0.4, we note that ConvLSTM has 100% top-5 acc. for both S1 and S2, whereas the 3D CNN has 80% and 60% (40% and 40% for TimeSf). As for the texture (T) results, the top-5 acc. of the 3D CNN remains at 80% relative to S1 and even improves from 60% to 80% relative to S2, whereas ConvLSTM drops by 40% and TimeSf drops by 50%. Thus, so far ConvLSTM and TimeSf display sound dropping on T. Next, we study the predictions made by the models in detail to observe that there is a qualitative difference between the predictions of ConvLSTM and TimeSf.

Each label of Diving48 has four attributes: takeoff, somersault, twist and flight position. Among the top-1 predictions for both S1 and S2 (Table 7), we study how many attributes are correct in the misclassifications for each model. Class 34 has the attribute values *inward* takeoff, *2.5* somersault, *no* twist and *tuck* flight position. For ConvLSTM, the misclassifications of class 34 are 8, 20, 35 and 44, where 8, 35 and 44 all contain 3/4 correct attributes, and 20 contains 1/4 correct attributes (no twist). For the 3D CNN, only two predictions (32, 35) obtain three correct attributes, and for TimeSf, the best misclassification has only two correct attributes. This suggests that the 3D CNN and TimeSf have modeled the classes in terms of the true attributes to a lesser extent than ConvLSTM, i.e., ConvLSTM has learned more relevant temporal patterns, at the same global validation performance. Observing the three lower sections of Table 7 for further randomly selected classes 12, 22 and 45, the ConvLSTM still achieves the largest proportion of correct attributes in the misclassifications. Just as for class 34, the

| Model | Top-1 predictions for five random clips from class **34** | | Misclassifications (set) | Correct attr. |
	S1	S2		
ConvLSTM	[34, 34, 35, 8, 20]	[34, 34, 44, 8, 20]	8, 20, 35, 44	**10/16**
3D CNN	[34, 19, 21, 35, 34]	[34, 32, 21, 21, 34]	19, 21, 32, 35	8/16
TimeSf	[34, 12, 34, 47, 20]	[31, 12, 34, 47, 20]	12, 20, 31, 47	5/16
	Top-1 predictions for five random clips from class **12**			
ConvLSTM	[35, 26, 45, 26, 21]	[27, 26, 45, 14, 21]	14, 21, 26, 27, 35, 45	**14/24**
3D CNN	[3, 20, 12, 5, 44]	[3, 20, 12, 5, 34]	3, 5, 20, 34, 44	8/20
TimeSf	[22, 33, 12, 31, 14]	[22, 33, 12, 31, 14]	14, 22, 31, 33	5/16
	Top-1 predictions for five random clips from class **22**			
ConvLSTM	[26, 26, 35, 22, 21]	[26, 26, 35, 22, 21]	21, 26, 35	5/12
3D CNN	[29, 7, 26, 28, 0]	[29, 22, 26, 26, 0]	0, 7, 26, 28, 29	7/20
TimeSf	[15, 27, 46, 44, 34]	[15, 27, 46, 44, 34]	15, 27, 34, 44, 46	5/20
	Top-1 predictions for five random clips from class **45**			
ConvLSTM	[26, 21, 12, 35, 27]	[26, 21, 12, 35, 44]	12, 21, 26, 27, 35, 44	14/24
3D CNN	[46, 20, 35, 35, 34]	[34, 20, 12, 35, 34]	12, 20, 31, 34, 35, 46	12/24
TimeSf	[15, 31, 44, 12, 18]	[42, 31, 44, 12, 8]	8, 12, 15, 18, 31, 42, 44	11/28

Table 7. Examples of predictions and misclassifications. Each class has four attributes, and the Correct attr. column shows how many attributes were correct among the set of misclassifications.
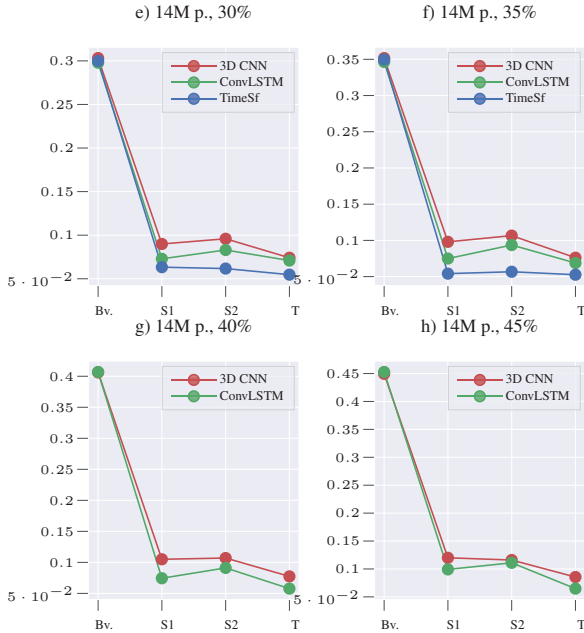

Figure 6. Experiments e-h on Diving48, reads as Fig. 5.

3D CNN comes second, and TimeSf last.[3]

Experiments e-h. The results for experiments e-h, where the number of trainable parameters and performance are fixed, are shown in Fig. 6 (tabulated results in the supplemental). Here, the 3D CNN is the most robust out of the three, although ConvLSTM approaches the 3D CNN and drops more steeply for T in g-h, where the performance is higher (40% and 45% acc.). In these experiments, although least robust, TimeSf does not improve on T relative to S1 and S2 any more. This suggests that TimeSf is more likely to display texture bias when it has a larger amount of parameters, as it does in experiments a-c.

---

[3]Tables containing the corresponding top-1 and top-5 accuracy for these additional clips are in the supplemental.

5. Conclusions and discussion

We have studied cross-domain robustness for three models that are principally different in terms of temporal modeling, in their bare-bones settings. A 3D CNN treats frames as a linear volume, a VT lets frames have non-linear but time-symmetric relationships, and a ConvLSTM models frame dependency non-linearly in a time-causal direction. Recently, a discrepancy in terms of feedback connections between biological and computer vision has been discussed [32, 56], and in our work we have hypothesized that the lack of feedback connections is one reason for the similarly lacking generalization abilities in computer vision.

Our experiments were carried out on two very different datasets, one synthetic, without bias, and one with natural data, thus with more noise and potential spatial bias. The combined results (Figs. 3-5, Tables 1 and 7) on these datasets indicated that convolutional-recurrent temporal modeling is more robust to domain shift than self-attention and 3D convolutions in terms of bare-bones behavior, presumably owing to its lesser encoding of texture bias. Our results are fully reproducible with public seeds, code and data. The fact that our observations regarding texture bias are made for a fine-grained dataset such as Diving48, constructed to contain as little bias as possible, suggests that the issue may be worse when it comes to more spatially biased datasets such as Kinetics, which is left for future work. It is furthermore left for future work whether ImageNet pre-trained VTs display more or less texture bias than their trained-from-scratch counterparts. Another observation from our study is that when the parameter count was kept equal (experiments e-h), these trends were less pronounced.

Moreover, qualitative random examples consistently showed that the ConvLSTM learned more relevant diving patterns than the two others, when scrutinizing the three models' misclassifications – which emphasizes the texture bias tendency of TimeSf and the 3D CNN. Sharing parameters across timesteps, as recurrent models do, narrows the parameter space, possibly incentivizing these models to prioritize which patterns to learn. Another reason to use smaller models is that they require less data to train, which is ethically desirable, both in that the data can be inspected more easily, and from a sustainability perspective [3].

Our study indicates that sound physical inductive bias such as recurrence in temporal modeling may be advantageous when robustness to domain shift is important for the task. In action recognition, benchmarking has thus far mainly been conducted for in-domain-tasks where large models perform well. We encourage the video understanding community to increasingly conduct evaluation on tasks involving domain shift. We hope that our proposed datasets and framework for evaluation can help such future domain shift robustness investigations of spatiotemporal features.

References

[1] Alessandra Angelucci and Paul C. Bressloff. Contribution of feedforward, lateral and feedback connections to the classical receptive field center and extra-classical receptive field surround of primate V1 neurons. *Progress in Brain Research*, 154:93–120, 2006.

[2] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[3] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021.

[5] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[6] Chun-Fu Richard Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of CNN-based spatio-temporal representations for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[7] Min-Hung Chen, Zsolt Kira, Ghassan Al-Regib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *2019 IEEE International Conference on Computer Vision*, 2019.

[8] Jinwoo Choi, Chen Gao, Joseph C.E. Messou, and Jia-Bin Huang. Why can't I dance in the mall? Learning to mitigate scene bias in action recognition. In *Advances in Neural Information Processing Systems*, 2019.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[10] Vincent di Lollo, James T. Enns, and Ronald A. Rensink. Competition for consciousness among visual events: the psychophysics of reentrant visual processes. *Journal of Experimental Psychology. General*, 129 4:481–507, 2000.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[12] R Douglas, Christof Koch, Misha A. Mahowald, KA Martin, and H. E. Ortiz Suarez. Recurrent excitation in neocortical circuits. *Science*, 269:981 – 985, 1995.

[13] Rodney J. Douglas and Kevan A. C. Martin. Recurrent neuronal circuits in the neocortex. *Current Biology*, 17:R496–R500, 2007.

[14] Debidatta Dwibedi, Pierre Sermanet, and Jonathan Tompson. Temporal reasoning in videos using convolutional gated recurrent units. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[15] Johannes Jacobus Fahrenfort, H. Steven Scholte, and Victor A. F. Lamme. Masking disrupts reentrant processing in human visual cortex. *Journal of Cognitive Neuroscience*, 19:1488–1497, 2007.

[16] William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*, 2020.

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *IEEE International Conference on Computer Vision*, 2019.

[18] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross B. Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[19] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[20] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, 2018.

[21] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Kumar Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[22] A. Ghodrati, E. Gavves, and C. G. M. Snoek. Video time: Properties, encoders and evaluation. In *British Machine Vision Conference*, 2018.

[23] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[24] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter N. Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *IEEE International Conference on Computer Vision*, 2017.

[25] Kensho Hara, Yuchi Ishikawa, and Hirokatsu Kataoka. Rethinking training data for mitigating representation biases in action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

[26] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

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

[28] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[29] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[30] Nicole Kobie. NVIDIA and the battle for the future of AI chips. *Wired*, 2021.

[31] Raivo Koot and Haiping Lu. Videolightformer: Lightweight action recognition using transformers. *arXiv preprint arXiv:2107.00451*, 2021.

[32] G. Kreiman and Thomas Serre. Beyond the feedforward sweep: feedback computations in the visual cortex. *Annals of the New York Academy of Sciences*, 1464, 2020.

[33] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In *IEEE International Conference on Computer Vision*, 2011.

[34] Victor A. F. Lamme and Pieter R. Roelfsema. The distinct modes of vision offered by feedforward and recurrent processing. *Trends in Neurosciences*, 23:571–579, 2000.

[35] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards action recognition without representation bias. In *European Conference on Computer Vision*, 2018.

[36] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*, 2014.

[37] Drew Linsley, Alekh Karkada Ashok, Lakshmi Narasimhan Govindarajan, Rex Liu, and Thomas Serre. Stable and expressive recurrent vision models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems*, 2020.

[38] Drew Linsley, Junkyung Kim, Alekh Ashok, and Thomas Serre. Recurrent neural circuits for contour detection. In *International Conference on Learning Representations*, 2020.

[39] Siyuan Ma, Raef Bassily, and Mikhail Belkin. The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning. In *International Conference on Machine Learning*, 2018.

[40] Joonatan Mänttäri*, Sofia Broomé*, John Folkesson, and Hedvig Kjellström. Interpreting Video Features: a Comparison of 3D Convolutional Networks and Convolutional LSTM Networks. In *Asian Conference on Computer Vision. (*Joint first authors)*, 2020.

[41] Alexandre Milesi. Accelerating SE(3)-Transformers Training Using an NVIDIA Open-Source Model Implementation. https://bit.ly/3wQac3v/. Accessed: 2021-11-01.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An

imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

[43] Lyndsey C. Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Schölkopf, and William T. Freeman. Seeing the arrow of time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[44] Paul Antoine Salin and J. Bullier. Corticocortical connections in the visual system: structure and function. *Physiological Reviews*, 75 1:107–54, 1995.

[45] Javier Selva, Anders S. Johansen, Sergio Escalera, Kamal Nasrollahi, Thomas B. Moeslund, and Albert Clapés. Video transformers: A survey. *arXiv preprint arXiv:2201.05991*, 2022.

[46] Thomas Serre. Deep learning: The good, the bad, and the ugly. *Annual Review of Vision Science*, 2019.

[47] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *IEEE Winter Conference on Applications of Computer Vision*, 2021.

[48] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang chun Woo. Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, 2015.

[49] Gunnar A. Sigurdsson, Olga Russakovsky, and Abhinav Kumar Gupta. What actions are needed for understanding human actions in videos? In *IEEE International Conference on Computer Vision*, 2017.

[50] Mahdi Soltanolkotabi, Adel Javanmard, and J. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65:742–769, 2019.

[51] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR*, abs/1212.0402, 2012.

[52] Hans Supèr, Henk Spekreijse, and Victor A. F. Lamme. Two distinct modes of sensory processing observed in monkey primary visual cortex (v1). *Nature Neuroscience*, 4:304–310, 2001.

[53] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. VIM-PAC: Video Pre-Training via Masked Token Prediction and Contrastive Learning, 2021.

[54] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE International Conference on Computer Vision*, 2015.

[55] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[56] Ruben S van Bergen and Nikolaus Kriegeskorte. Going in circles is the way forward: the role of recurrence in visual inference. *Current Opinion in Neurobiology*, 65:176–193, 2020. Whole-brain interactions between neural circuits.

[57] Heng Wang. TimeSformer-PyTorch. Implementation of TimeSformer from Facebook AI, a pure attention-based solution for video classification. `https://github.com/lucidrains/TimeSformer-pytorch`, 2021. Accessed: 2021-11-13.

[58] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[59] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:2740–2755, 2019.

[60] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin P. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*, 2018.

[61] Zhiyu Yao, Yunbo Wang, Jianmin Wang, Philip Yu, and Mingsheng Long. VideoDG: Generalizing Temporal Relations in Videos to Novel Domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[62] Chenyu Yi, Siyuan Yang, Haoliang Li, Yap-Peng Tan, and Alex C. Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *Advances in Neural Information Processing Systems*, 2021.

[63] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision*, 2018.