International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Learning and Using the Arrow of Time

Donglai Wei[1] · **Joseph Lim**[2] · **Andrew Zisserman**[3] · **William T. Freeman**[4,5]

Received: date / Accepted: date

Abstract We seek to understand the arrow of time in videos – what makes videos look like they are playing forwards or backwards? Can we visualize the cues? Can the arrow of time be a supervisory signal useful for activity analysis? To this end, we build three large-scale video datasets and apply a learning-based approach to these tasks.

To learn the arrow of time efficiently and reliably, we design a ConvNet suitable for extended temporal footprints and for class activation visualization, and study the effect of artificial cues, such as cinematographic conventions, on learning. Our trained model achieves state-of-the-art performance on large-scale real-world video datasets. Through cluster analysis and localization of important regions for the prediction, we examine learned visual cues that are consistent among many samples and show when and where they occur. Lastly, we use the trained ConvNet for two applications: self-supervision for action recognition, and video forensics – determining whether Hollywood film clips have been deliberately reversed in time, often used as special effects.

1 Introduction

We seek to learn to *see* the arrow of time – to tell whether a video sequence is playing forwards or backwards. At a small scale, the world is reversible–the fundamental physics equations are symmetric in time. Yet at a macroscopic scale, time is often irreversible and we can identify certain motion patterns (e.g., water flows downward) to tell the direction of time. But this task can be challenging: some motion patterns seem too subtle for human to determine if they are playing forwards or backwards, as illustrated in Figure 1. For example, it is possible for the train to move in either direction with acceleration or deceleration (Figure 1d).

Furthermore, we are interested in how the arrow of time manifests itself visually. We ask: first, can we train a reliable arrow of time classifier from large-scale natural videos while avoiding artificial cues? (i.e. cues introduced during video production, not from the visual world); second, what does the model learn about the visual world in order to solve this task?; and, last, can we apply such learned commonsense knowledge to other video analysis tasks?

Regarding the first question on the arrow of time classification, we go beyond the previous work (Pickup et al 2014) to train a ConvNet, exploiting thousands of hours of online videos, and let the data determine which cues to use. Such cues can come from high-level events (e.g., riding a horse), or low-level physics (e.g., gravity). But as discovered in previous self-supervision work (Doersch et al 2015), a ConvNet can learn artificial cues from still images (e.g., chromatic aberration) instead of a useful visual representation. Videos, as collections of images, have additional artificial cues introduced during creation (e.g. *camera motion*), compression (e.g. *inter-frame codec*) or editing (e.g. *black framing*), which may be used to indicate the video's temporal direction. Thus, we design controlled experiments to

Donglai Wei
E-mail: donglai@seas.harvard.edu

Joseph Lim
E-mail: limjj@usc.edu

Andrew Zisserman
E-mail: az@robots.ox.ac.uk

William T. Freeman
E-mail: billf@mit.edu

[1] Harvard University
[2] University of Southern California
[3] University of Oxford
[4] Massachusetts Institute of Technology
[5] Google Research

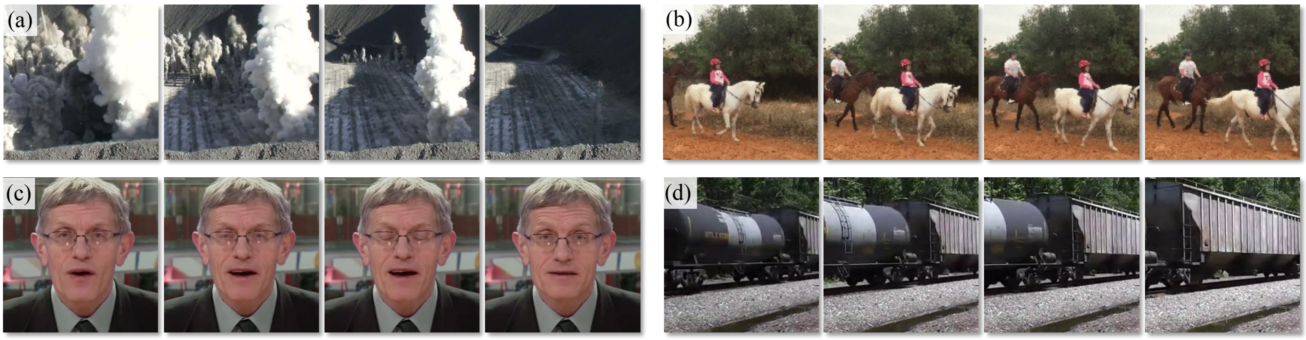


Fig. 1: Seeing these ordered frames from videos, can you tell whether each video is playing forward or backward? (answer below[1]). Depending on the video, solving the task may require (a) low-level understanding (e.g. physics), (b) high-level reasoning (e.g. semantics), or (c) familiarity with very subtle effects or with (d) camera conventions. In this work, we learn and exploit several types of knowledge to predict the arrow of time automatically with neural network models trained on large-scale video datasets.

understand the effect of artificial cues from videos on the arrow of time classification.

Regarding the second question on the interpretation of learned features, we highlight the observation from Zhou et al (2014): in order to achieve a task (scene classification in their case), a network implicitly learns what is necessary (object detectors in their case). We expect that the network will learn a useful representation of the visual world, involving both low-level physics and high-level semantics, in order to detect the forward direction of time.

Regarding the third question on applications, we use the arrow-of-time classifier for two tasks: video representation learning and video forensics. For representation learning, recent works have used temporal ordering for self-supervised training of an image ConvNet (Misra et al 2016; Fernando et al 2017). Instead, we focus on the motion cues in videos and use the arrow of time to pre-train action recognition models. For video forensics, we detect clips that are played backwards in Hollywood films. This may be done as a special effect, or to make an otherwise dangerous scene safe to film. We show good performance on a newly collected dataset of films containing time-reversed clips, and visualize the cues that the network uses to make the classification. More generally, this application illustrates that the trained network can detect videos that have been tampered in this way. In both applications we exceed the respective state of the art.

In the following, we first describe our ConvNet model (Section 2), incorporating recent developments for human action recognition and network interpretation. Then we identify and address three potential confounds to learning the arrow of time discovered by the ConvNet (Section 4), for example, exploiting prototypical camera motions used by directors. With the properly pre-processed data, we train our model using two large video datasets (Section 5): a 147k clip subset of the Flickr100M dataset (Thomee et al 2016) and a 58k clip subset of the Kinetics dataset (Kay et al 2017). We evaluate test performance and visualize the representations learned to solve the arrow-of-time task. Lastly, we demonstrate the usefulness of our ConvNet arrow of time detector for self-supervised pre-training in action recognition and for identifying clip from Hollywood films made using the reverse-motion film technique (Section 7).

1.1 Related Work

Several recent papers have explored the usage of the temporal *ordering* of images. Dekel et al (2014) consider the task of photo-sequencing – determining the temporal order of a collection of images from different cameras. Others have used the temporal ordering of frames as a supervisory signal for learning an embedding (Ramanathan et al 2015), for self-supervision training of a ConvNet (Misra et al 2016; Fernando et al 2017), and for construction of a representation for action recognition (Fernando et al 2015).

However, none of these previous works address the task of detecting the direction of time. Pickup et al (2014) explore three representations for determining time's arrow in videos: asymmetry in temporal behaviour (using hand-crafted SIFT-like features), evidence for causality, and an auto-regressive model to determine if a cause influences future events. While their methods work on a small dataset collected with known strong arrow of time signal, it is unclear if the method works on generic large-scale video dataset with different artificial signals. The study of the arrow of time is a special case of causal inference, which has been connected to machine learning topics, such as transfer learning and covariate shift adaptation (Schölkopf et al 2012). Recently, Xie et al (2017)

[1] Forwards: (b), (c); backwards: (a), (d). Though in (d) the train can move in either direction.

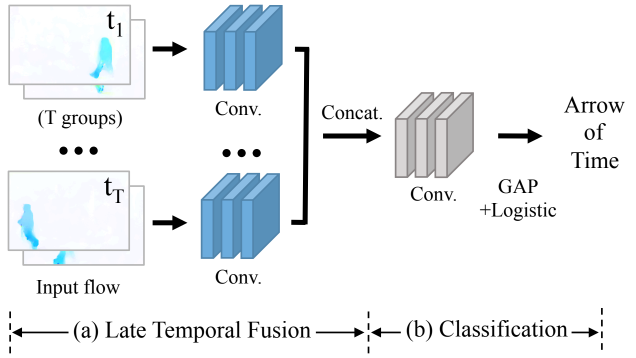


Fig. 2: Illustration of our Temporal Class-Activation-Map Network (T-CAM) for the arrow of time classification. Starting from the traditional VGG-16 architecture (Simonyan and Zisserman 2014b) for image recognition, (a) we first concatenate the *conv*5 features from the shared convolutional layers, (b) and then replace the fully-connected layer with three convolution layers and global average pooling layer (GAP) (Lin et al 2013; Springenberg et al 2014; Szegedy et al 2015; Zhou et al 2016) for better activation localization.

discovered that the I3D action recognition model (Carreira and Zisserman 2017) is invariant to the direction of time but not to the frame order, when trained on RGB frame input.

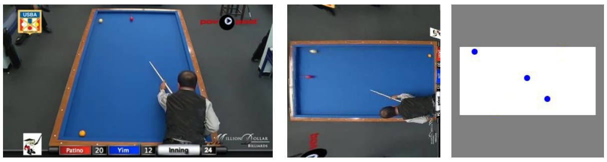
In terms of ConvNet architectures, we borrow from recent work that has designed ConvNets for action recognition in videos with optical flow input to explicitly capture motion information (Simonyan and Zisserman 2014a; Wang et al 2016). We also employ the Class Activation Map (CAM) visualization of Zhou et al (2016).

2 ConvNet Architecture

To focus on the time-varying aspects of the video, we only use optical flow as input to the ConvNet, and not its RGB appearance. Below, we first motivate the architecture, and then describe implementation details.

Model design. Our aim is to design a ConvNet that has an extended temporal footprint, and that also enables the learned features to be visualized. We also want the model to have sufficient capacity to detect subtle temporal signals. To this end, we base our model on three prior ConvNets: the VGG-16 network (Simonyan and Zisserman 2014b) as the backbone for the initial convolutional layers, for sufficient capacity; the temporal chunking in the model of Feichtenhofer et al (2016) to give an extended temporal footprint; and the CAM model of Zhou et al (2016) to provide the visualization.

The resulting architecture is referred to as "Temporal Class-Activation Map Network" (T-CAM) (Figure 2). For the temporal feature fusion stage (Figure 2a), we first modify the VGG-16 network to accept a number of frames (e.g. 10) of optical flow as input by expanding the number of channels of *conv*1 filters (Wang et al 2016). We use T such temporal chunks, with a temporal stride of τ. The *conv*5 features from each chunk are then concatenated. Then for the classification stage (Figure 2b), we follow the CAM model design to replace fully-connected layers with three convolution layers and global average pooling (GAP) before the binary logistic regression. Batch-Normalization layers (Ioffe and Szegedy 2015) are added after each convolution layer.



(a) original frame (b) warped frame (c) simulation

Fig. 3: The 3-cushion billiard dataset. (a) Original frame from a 3-cushion video; (b) the frame warped (with a homography transformation) to an overhead view of the billiard table; and, (c) a simulated frame to match the real one in terms of size and number of balls.

Implementation details. To replace the fully-connected layers from VGG-16, we use three convolution layers with size $3\times3\times1024$, stride 1×1 and pad 1×1 before the GAP layer. For input, we use TV-L1 (Zach et al 2007) to extract optical flow.

For all experiments in this paper, we split each dataset 70%-30% for training and testing respectively, and feed both forward and backward versions of the video to the model. The model is trained end-to-end from scratch, using fixed five-corner cropping and horizontal flipping for data augmentation. Clips with very small motion signals are filtered out from the training data using flow. Given a video clip for test, in addition to the spatial augmentation, we predict AoT on evenly sampled groups of frames for temporal augmentation. The final AoT prediction for each video is based on the majority vote of confident predictions (i.e. score $|x-0.5|>0.1$), as some groups of frames may be uninformative about AoT.

3 Learning from Simulation Videos

An an initial evaluation of the T-CAM model, we first avoid the confounding factors in real world videos (e.g. temporal codec or sample bias) and turn to graphics simulations where we have full control of the physics. We choose to simulate a simple world, the *three-cushion billiards game* (Figure 3a), where the principal signals for the arrow of time are: rolling friction and energy loss at collisions/bounces.

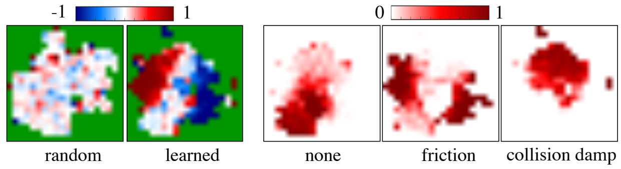
Method	3c-AoT-S	3c-AoT
Pickup et al (2014)	63%	59%
T-CAM (T=1)	78%	83%
T-CAM (T=2)	81%	85%

(a)

Signal	Acc
none	50%
friction	97%
collision	95%

(b)

Table 1: Test accuracy on the Three-cushion datasets. (a) We compare T-CAM model with either one (T=1) or two (T=2) temporal segments with the baseline model on both simulation (3c-AoT-S) and real wolrd (3c-AoT) datasets. (b) For the simulation dataset, we show the test accuracy for videos grouped by different physical settings.



(a) AoT heatmap (b) density map of physical settings

Fig. 4: Visualization results on 3c-AoT-S dataset. (a) heatmap; (b) density map.

We then apply the T-CAM model, trained on the simulations, to sequences of real billiards games.

3.1 Dataset

Three-cushion Arrow of Time Dataset (3c-AoT). We download YouTube videos from a three-cushion billiard tournament and extract 167 individual shots (around 200 frames each). These are only used for testing. As the cameras are placed at different angles, the perspective projection may cause the ball to appear to move faster as it comes towards the camera. To avoid this artifact, we warp the original frames (Figure3a) into a canonical overhead position (Figure3b).
Three-cushion Arrow of Time Simulation Dataset (3c-AoT-S). We extend the physics engine in Fragkiadaki et al (2015) to handle multiple balls with friction and collision damping (Figure 3c). We simulate 15k videos (100 frames) and randomly make them with one of three scenarios: no friction nor collision damping, friction only, and collision damping only. The physical parameters (rolling friction coefficient $\mu = 0.5$, and collision damping factor $\eta = 0.5$) are estimated from real videos in 3c-AoT.

3.2 Experiments

Classification. We train our model on the simulation dataset (3c-AoT-S) with different number of input segments (T=1 and T=2). We not only test on 3c-AoT-S, but also on the real sequences in 3c-AoT directly. As the baseline, we train Pickup et al (2014) (results reproduced on TA180) with the same setup.

Table 1 shows that temporal fusion (i.e. T=2) helps most when the signal is weak (i.e. friction) and the collision damping is a stronger signal. Note, there are frequent collisions, and so collision damping makes a significant contribution to the time asymmetry.
Visualization. In addition to test accuracy, we visualize the 2-dimensional t-SNE space of the last-layer motion feature learned by our T-CAM model (T=2). For each test video from 3c-AoT-S, equal chance to be forward or backward, we only extract the feature on the central crop.

Shown in Figure 4a, we discretize the t-SNE space into 20×20 bins and compute the heatmap through averaging ground truth AoT labels for videos from each bin. For comparison, we also visualize the heatmap for the feature from the same network architecture with random weights. As the t-SNE space is learned without any supervision, the heatmap reveals that the network learns to transform the initially temporally symmetric feature space into a temporally asymmetric one.

Further, as we also have the labels for the physical settings for each video, we can visualize different groups on the t-SNE space. In Figure 4b, for each bin, we visualize the ratio of each group, that is the ratio that is close to 1 indicates most videos from this bin are from such group. see that forward and backward videos with "friction" are clearly separable; with "collision damp" are often separable as collisions may not happen within the central temporal crop; with "none" are inseparable.

4 Avoiding Artificial Cues from Videos

A learning-based algorithm may "cheat" and solve the arrow-of-time task using artificial cues, instead of learning about the video content. In this section, we evaluate the effect of three artificial signals, black framing, camera motion and inter-frame codec, on ConvNet learning and the effectiveness of our data pre-procession to avoid them.

4.1 Datasets regarding artificial cues

We use the following two datasets to study artificial cues.
UCF101 (Soomro et al 2012). To examine the black framing and camera motion signal, we use this popular human action video dataset (split-1). Through automatic algorithms (i.e. black frame detection and homography estimation) and manual pruning, we find that around 46% of the videos have black framing, and 73% have significant camera motion (Table 2).

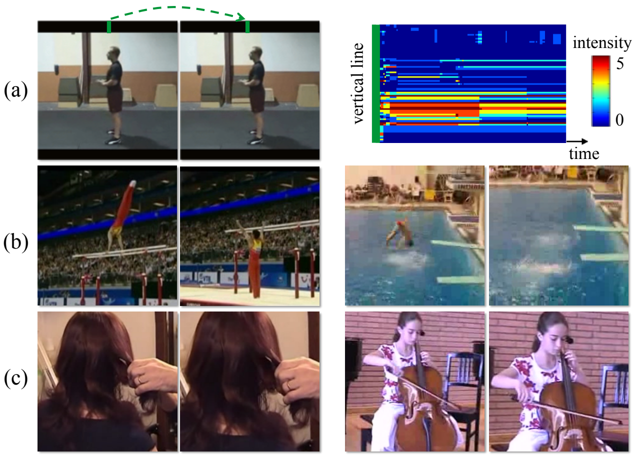


Fig. 5: Illustration of artificial signals from videos in UCF101 dataset. (a) The black framing of the clip has non-zero intensity value (left), and a vertical slice over time displays an asymmetric temporal pattern (right). After training, we cluster the learned last-layer feature of top-confident test clips. We find some clusters have consistent (b) tilt-down or (c) zoom-in camera motion. We show two frames from two representative clips for each cluster.

		Black frame	+Camera motion
Percent of videos		46%	73%
Acc.	before removal	98%	88%
	after removal	90%	75%

Table 2: AoT classification results to explore the effect of black framing and camera motion on UCF101 dataset. AoT test accuracy drops around 10% after removing black framing and drops another 10% after removing camera motion.

MJPEG Arrow of Time Dataset (MJPEG-AoT). To investigate the effect of inter-frame codec, we collect a new video dataset containing 16.9k individual shots from 3.5k videos from Vimeo[2] with diverse content. The collected videos are either uncompressed or encoded with intra-frame codecs (e.g. MJPEG and ProRes) where each frame is compressed independently without introducing temporal direction bias. We can then evaluate performance with and without inter-frame codecs by using the original frames or the extracted frames after video compression with an inter-frame codec (e.g. H.264). The details of the dataset are in the appendix.

4.2 Experiments regarding artificial cues

We choose the T-CAM model to have two temporal segments and a total of 10 frames. More experimental details are in the appendix.

Train/Test	Original	H.264-F	H.264-B
Original	59.1%	58.2%	58.6%
H.264-F	58.1%	58.9%	58.8%
H.264-B	58.3%	59.0%	58.8%

Table 3: AoT classification results to explore the effect of the inter-frame codec on MJPEG-AoT dataset. We train and test on three versions of the data: original (no temporal encoding), encoded with H.264 in forward (H.264-F) and backward (H.264-B) direction. Similar AoT test accuracy suggests that the common H.264 codec doesn't introduce significant artificial signals for our model to learn from.

Black framing. Black frame regions present at the boundary may not be completely black after video compression (Figure 5a). The resulting non-zero image intensities can cause different flow patterns for forward and backward temporal motion, providing an artificial cue for the AoT.

For control experiments, we train and test our model on UCF101 before and after black framing removal, i.e., zero out the intensity of black frame regions. The test accuracy of the AoT prediction drops from 98% to 90% after the removal. This shows that black frame regions provides artificial cues for AoT and should be removed.

Camera motion. To understand the visual cues learned by our model after black framing removal, we perform K-means (K=20) clustering on the extracted feature before the logistic regression layer for the top-1K confidently classified test videos (foward or backward version). We estimate the homography for each video's camera motion with RANSAC, and compute the average translation and zoom in both horizontal and vertical directions. We find some video clusters have consistently large vertical translation motion (Figure 5b), and some have large zoom-in motion (Figure 5c). Such strong correlation among the confident clips between their learned visual representation and the camera motion suggests that cinematic camera motion conventions can be used for AoT classification.

For control experiments, we use a subset of UCF101 videos that can be well-stabilized. The test accuracy of the AoT prediction further drops from 88% to 75% before and after stabilization. Thus, we need to stabilize videos to prevent the model from using camera motion cues.

Inter-frame codec. For efficient storage, most online videos are compressed with temporally-asymmetric video codecs, e.g. H.264. They often employ "Forward prediction", which may offer an artificial signal for the direction of time. As it is almost impossible to revert the codecs, we train and test on our specially collected MJPEG-AoT dataset, where videos are not subject to this artificial signal.

We first remove black framing from these videos and choose individual shots that can be well-stabilized, based

[2] http://vimeo.com

on the discoveries above. Then we create different versions of the downloaded MJPEG-AoT dataset (Original) by encoding the videos with the H.264 codec in either the forward (H.264-F) or backward direction (H.264-B), to simulate the corruption from the inter-frame codec. In Table 3 we show results where the model is trained on one version of the MJPEG-AoT dataset and tested on another version. Notably, our model has similar test accuracy, indicating that our model can not distinguish videos from each dataset for the AoT prediction. This finding offers a procedure for building a very large scale video dataset starting from videos that have been H.264 encoded (e.g. Youtube videos), without being concerned about artificial signals.

Conclusion. We have shown that black framing and camera motion do allow our model to learn the artificial signals for the AoT prediction, while the inter-frame codec (e.g. H.264) does not introduce significant signals to be learned by our model. For the experiments in the following sections we remove black framing and stabilize camera motion to preprocess videos for the AoT classification.

5 Learning the Arrow of Time

After verifying our T-CAM model on simulation videos and removing the known artificial signals from real world videos, we benchmark it on three real world video datasets and examine the visual cues it learns to exploit for the AoT.

5.1 Datasets

The previous AoT classification benchmark (Pickup et al 2014) contains only a small number of videos that are manually selected with strong AoT signals. To create large-scale AoT benchmarks with general videos, we pre-process two existing datasets through automated black framing removal and camera motion stabilization within a footprint of 41 frames. We use a fixed set of parameters for the data preprocessing, with the details in the appendix. We then use the following three video datasets to benchmark AoT classification.

TA-180 (Pickup et al 2014). This dataset has 180 videos manually selected from Youtube search results for specific keywords (e.g. “dance” and “steam train”) that suggest strong low-level motion cues for AoT. As some videos are hard to stabilize, in our experiments we only use a subset of 165 videos that are automatically selected by our stabilization algorithm.

Flickr Arrow of Time Dataset (Flickr-AoT). The Flickr video dataset (Thomee et al 2016; Vondrick et al 2016) is unlabeled with diverse video content, ranging from natural scenes to human actions. Starting from around 1.7M Flickr videos, we obtain around 147K videos after processing to remove artificial cues.

# chunks	T=1			T=2		T=4
# frame	10	20	40	10	20	20
0% overlap	65%	62%	67%	79%	**81**%	71%
50% overlap	N/A			75%	76%	73%

Table 4: Empirical ablation analysis of T-CAM on Flickr-AoT. We compare the AoT test accuracy for models with a different number of input chunks (T), total number of frames, and overlap ratio between adjacent chunks. The best model takes in a total 20 frames of flow maps as input, and divides them into two 10-frame chunks without overlap to feed into the model.

Kinetics Arrow of Time Dataset (Kinetics-AoT). The Kinectics video dataset (Kay et al 2017) is fully labeled with 400 categories of human actions. Starting from around 266K train and validation videos, we obtain around 58K videos after processing to remove artificial cues. To balance for the AoT classification, we re-assign train and test set based on a 70-30 split for each action class.

5.2 Empirical ablation analysis

On the Flickr-AoT dataset, we present experiments to analyze various design decisions for our T-CAM model. With the same learning strategies (e.g. number of epochs and learning schedule), we compare models trained with (i) a different number of temporal segments (chunks); (ii) differing total number of input frames of flow; and (iii) varying overlap ratio between adjacent temporal segments.

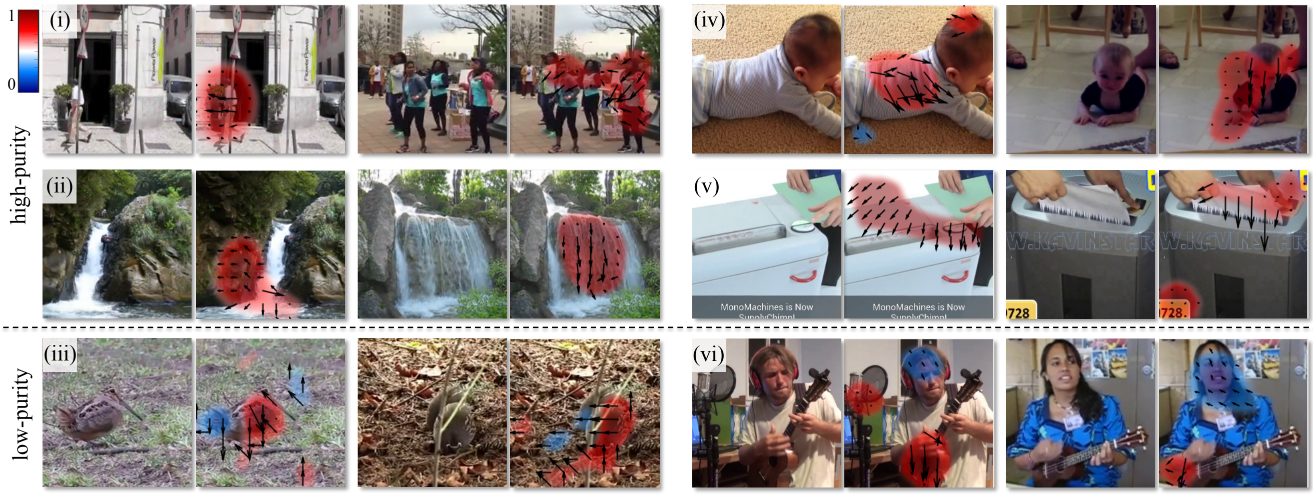
In Table 4, we find that the best T-CAM model on Flickr-AoT has two temporal segments with 20 frames total without overlap. We use this model configuration for all the experimental results in this section.

5.3 Experiments

In the following, we benchmark AoT classification results on all three datasets above.

Setup. For the baseline comparison, we implement the previous state-of-the-art, statistical flow method (Pickup et al 2014), and achieve similar 3-fold cross-validation results on the TA-180 dataset. To measure human performance, we use Amazon Mechanical Turk (AMT) for all three benchmark datasets (using random subsets for the large-scale datasets), where input videos have the same time footprint (i.e. 20 frames) as our T-CAM model. More details about the AMT study are in the appendix.

Classification results. On the TA-180 benchmark (Pickup et al 2014), we only test with models trained on Flickr-AoT



(a) Clusters in Flickr-AoT

(b) Action classes in Kinetics-AoT

Fig. 6: Examples of T-CAM localization results on test clips from (a) Flickr-AoT and (b) Kinetics-AoT dataset. For each input clip, we compute its class activation map (CAM) from the model trained on the same dataset. We show its middle frame on the left, and overlay color-coded CAM (red for high probability of being forward, blue for backwards) and sparse motion vector on regions with confident AoT classification. For each dataset, we show localization results for two AoT-consistent clusters (i.e., most clips have the same AoT label within the cluster) and one AoT-inconsistent cluster. All the examples here are played in the forward direction and AoT in regions with red CAM are correctly classified. Notice that examples from AoT-inconsistent clusters have a mix of red and blue regions.

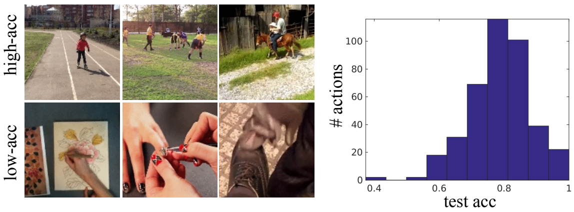
Data (#clip)	Flow-Word	T-CAM		Human
		Flickr	Kinetics	
TA-180 (165)	82%	**83%**	79%	93%
Flickr-AoT (147k)	62%	**81%**	73%	81%
Kinetics-AoT (58k)	59%	71%	**79%**	83%

Table 5: AoT classification benchmark results on three datasets. We compare the T-CAM model, trained on either Flickr-AoT or Kinetics-AoT, with the previous state-of-the-art method (Pickup et al 2014) and with human performance. The T-CAM models outperform Pickup et al (2014) on the large-scale datasets and achieves similar results on the previous TA-180 benchmark (Pickup et al 2014) (for test only).

or Kinetics-AoT dataset, as the dataset is too small to train our model. As shown in Table 5, the performance of the T-CAM models on TA-180, without any fine-tuning, are on-par with Pickup et al (2014), despite being trained on different datasets. Testing on the large-scale datasets, Flickr-AoT and Kinetics-AoT, our T-CAM models are consistently better than Pickup et al (2014) and are on par with human judgment. To compare the effectivenss of different architecture, we replace the backbone from VGG-16 with ResNet-50 and get similar performance on Flicker-AoT dataset.

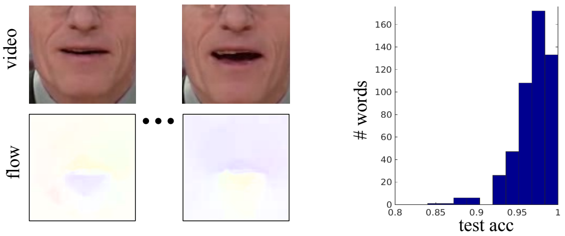
Localization results. We localize regions that contribute most to the AoT prediction using techniques in Zhou et al (2016). Given the 14×14 class activation map, we normalize it to a 0-1 probability heatmap p and resize it back to the original image size. Image regions are considered important for AoT prediction if their probability value is away from the random guess probability 0.5, i.e. $|p-0.5| > 0.2$. To visualize these important regions, we compute both the color-coded heatmap with a "blue-white-red colormap", where time forward evidence is red (close to 1) and backward is blue (close to 0), and also the sparse motion vectors on the middle frame of the input. In Figure 6, for each example we show its middle frame and that with the heatmap and motion vector overlay for regions with confident predictions.

Clustering results. For both Flickr-AoT and Kinetics-AoT datasets, we discover clusters of consistent motion pattern that are either indicative or not for the AoT. Given the feature maps from the last convolutional layer, we perform K-means clustering with K=50. For each cluster, we compute the standard deviation of AoT label from the cluster samples. A cluster with low standard deviation of AoT label is "AoT-consistent", as its samples share the common feature that is indicative for AoT prediction. For the Flickr-AoT dataset, the two visualized AoT-consistent clusters correspond to the visual concepts of "human walk" and "water fall" (Figure 6a). For the Kinetics-AoT dataset, the two visualized AoT-conistent action classes are "crawling baby" and "shredding paper", while the AoT-inconsistent action class is "playing ukulele" (Figure 6b).



(a) top-3 action classes (b) histogram of acc for each action

Fig. 7: AoT classification results on the Kinectics-AoT dataset by human action classes. (a) top three action classes that have either highest or lowest test accuracy. The high accuracy classes (“roller skating”, “passing American football in game” and “riding mule”) have clear motion direction; while the low accuracy classes (“brush painting”, “doing nails” and “shining shoes”) are visually repetitive. (b) distribution of test accuracy over different actions.



(a) example video and input flow (b) histogram of acc for each word

Fig. 8: AoT classification results on the lip reading dataset by word labels. (a) input motion of mouth region, and (b) distribution of test accuracy over different words.

6 Is Human Motion Visually Symmetric in Time?

It is interesting to consider whether human motion are visually reversible. For example, a person sitting down is aided by gravity, but standing up must work against gravity. This asymmetry is reflected in the muscle patterns and the temporal sequence of the body posture. For case studies, we make use of semantic labels of human motion and investigate what type of body motion or lip motion is visually symmetric in time.

6.1 Body Motion

As clips from the Kinetics-AoT dataset are labeled with human action class, we directly analyze the AoT classification result from the previous section for each action class. We show the histogram of the test accuracy over different human action (Figure 7a), where most have around 80% test accuracy. For the extremes, we show the top three actions with highest accuracy are “roller skating”, “passing American football in game” and “riding mule”, and the actions with lowest accuracy (chance performance) are: “brush painting”, “doing nails” and “shining shoes” (Figure 7b). As expected, action classes with high AoT accuracy have clear motion direction; while those with low accuracy are visually repetitive.

6.2 Lip Motion

We investigate if it is possible to determine if a face is speaking backwards or forwards using visual information alone. That the direction can be determined will depend, of course, on what is spoken – some words or phrases will be the visual equivalent of a palindrome, and it will not be possible to tell the direction for these. Conversely, if we start with a character palindrome (like ‘racecar’), it does not follow that this will be a visual palindrome, just as it does not follow that it will be an audio palindrome.

Dataset. We use the large-scale ‘Lip Reading in the Wild’ (LRW) dataset (Chung and Zisserman 2016). This has 1-second long video clips for 500 different words, with around 1000 examples for each word ‘spoken’ by hundreds of different speakers. We use the dataset’s training and test partitions, so that all tests are on unseen samples of 50 clips per word for the 500 word test set. The dataset provides a stabilized lip region (which we train and test on) as well as full faces. An example clip is shown in Figure 8a.

Models and results. We train a new T-CAM model on the LRW dataset, using the training procedure described in section 2. The performance is significantly bettern than that on generic videos (i.e. on Flickr-AoT), with the T-CAM model able to capture the arrow-of-time signal very well in this specific domain. Training on the entire training set gives a time’s arrow classification test performance of 97.6%. However, the model is actually able to learn from far fewer words than this – for example it can reach a performance of 83.2% when trained on as few as 10 words (meaning that it has not seen examples of the other 490 words at all).

We show the histogram of the test accuracy over different words (Figure 8b), where most have around 95% test accuracy. For the extremes, we find that the top five words are: ‘Warning’, ‘Weekend’, ‘Today’, ‘Morning’ and ‘Build’; and the words with lowest accuracy (chance performance) are: ‘System’, ‘National’, ‘Global’, ‘George’ and ‘Enough’. Of these, ‘George’ is fairly close to an audio palindrome, though none are character palindromes.

7 Using the Arrow of Time

In this section, we describe two applications of the arrow of time signal: self-supervised pre-training for action recognition, and reverse film detection for video forensics.

Input	Pre-train		Arch.	Fine-tune		
	Label	Dataset		Last layer	After fusion	All layers
Flow	N/A (Random init.)		VGG-16	-	-	81.7% (Wang et al 2016)
			T-CAM	38.0%	53.1%	79.3%
	1k Object class	ImageNet	VGG-16	-	-	85.7% (Wang et al 2016)
			T-CAM	47.9%	68.3%	84.1%
	2 AoT class (ours)	UCF101	T-CAM	**58.6%**	**81.2%**	**86.3%**
		Flickr-AoT		57.2%	79.2%	84.1%
		Kinetics-AoT		55.3%	74.3 %	79.4%

Table 6: Action classification on UCF101 split-1 with flow input for different pre-training and fine-tuning methods. For random and ImageNet initialization, our modified T-CAM model achieves similar result to the previous state-of-the-art (Wang et al 2016) that uses a VGG-16 network. Self-supervised pre-training of the T-CAM model using the arrow of time (AoT) consistently outperforms random and ImageNet initialization, i.e. for all three datasets and for fine-tuning on three different sets of levels.

Method/Dataset	UCF101			HMDB51
	split1	split2	split3	
Wang et al (2016)	85.7%	88.2%	87.4%	55.0%
AoT (ours)	**86.3%**	**88.6%**	**88.7%**	**55.4%**

Table 7: Additional action classfication results on UCF101 (3 splits) and HMDB51. We use the flow input and fine-tune them for all layers. Our T-CAM models pre-trained with AoT classes on the respective action recognition data outperforms the previous state-of-the-art (Wang et al 2016) using VGG-16 models pre-trained with object classes on ImageNet.

7.1 Self-supervised pre-training

Initialization plays an important role in training neural networks for video recognition tasks. Self-supervised pre-training has been used to initialize action classification networks for UCF101, for example by employing a proxy task such as frame order, that does not require external labels (Misra et al 2016; Fernando et al 2017; Liu et al 2017). For image input (i.e. the spatial stream), these approaches show promising results that are better than random initialization. However, their results are still far from the performance obtained by pre-training on a supervised task such as ImageNet classification (Simonyan and Zisserman 2014a; Wang et al 2016). Further, there has been little self-supervision work on pre-training for the flow input (the temporal stream). Below we first show that the AoT signal can be used to pre-train flow-based action recognition models to achieve state-of-the-art results on UCF101 and HMDB51. Then to compare with previous self-supervision methods, we explore the effects of different input modalities and architectures on self-supervision with the AoT signal for UCF101 split-1.

Results with T-CAM model. To benchmark on UCF101 split-1, we pre-train T-CAM models with three different datasets and fine-tune each model with three different sets of layers. For pre-training, we directly re-use the models trained in the previous sections: one on UCF101 (on the subset that can be stabilized with black framing removed) from section 4, and also those trained on Flickr-AoT and Kinetics-AoT. To fine-tune for action classification, we replace the logistic regression for AoT with classification layers (i.e., a fully-connected layer + softmax loss), and fine-tune the T-CAM model with action labels. To understand the effectiveness of the AoT features from the different layers, we fine-tune three sets of layers separately: the last layer only, all layers after temporal fusion, and all layers. To compare with Wang et al (2016), we redo the random and ImageNet initialization with the T-CAM model instead of the VGG-16 model, use 10 frames' flow maps as input, and only feed videos played in the original direction.

In Table 6, we compare self-supervision results for different initialization methods that use flow as input. First, it can be seen from the random and ImageNet initializations, that a VGG-16 model (Wang et al 2016) has similar performance to the T-CAM model when fine-tuned on all layers. Second, self-supervised training of the T-CAM model with AoT on each of the three datasets outperforms random and ImageNet initialization for fine-tuning tasks at *all three* different levels of the architecture. Third, our AoT self-supervision method exceed the state-of-the-art when pre-trained on UCF101.

To benchmark on UCF101 other splits and HMDB51 dataset, we choose the best setting from above, that is to pre-train T-CAM model on the action recognition data and fine-tune for all layers. As shown in Table 7.1, our AoT self-supervision results outperform ImageNet pre-training (Wang et al 2016) by around 0.5% consistently.

Comparison with other input and architectures. To further explore the effect of backbone architectures and modalites, we compare T-CAM with VGG-16 to ResNet-50, and stacked

Input	Pre-train	Arch.	Accuracy
Flow	AoT	VGG-16	86.3%
		ResNet-50	**87.2%**
RGB		VGG-16	78.1%
		ResNet-50	**86.5%**
D-RGB		VGG-16	85.8%
		ResNet-50	**86.9%**

Table 8: Action classification on UCF101 split-1, using AoT self-supervision but with other input and architectures. We compare results using VGG-16 and ResNet-50 backbone architectures, and flow, RGB and D-RGB input.

Input	Pre-train	Arch.	Accuracy
RGB	Rand.	AlexNet	38.6%
	Shuffle		50.9%
	AoT (ours)		**55.3%**
D-RGB	Odd-One		60.3%
	AoT (ours)		**68.9%**

Table 9: Action classification on UCF101 split-1, using AlexNet architecture but different self-supervision methods. We compare our results pre-trained with AoT to previous self-supervision methods: shuffle-and-learn (Misra et al 2016) with RGB input and odd-one network (Fernando et al 2017) with D-RGB input.

frames of flow to those of RGB and RGB difference (D-RGB) for action recognition on UCF101 split-1 dataset (Table 8). All models are pre-trained on UCF101 split-1 with 20-frame input and fine-tuned with all layers. To pre-train AoT with RGB and D-RGB input, we modify the number of channels of *conv1* filters correspondingly. In terms of the backbone architecture, the ResNet-50 models consistently outperform VGG-16 for each input modality. In terms of the input modality, all three modalities have similar performance for action recognition using ResNet-50 with our AoT pre-training and also with ImageNet pre-training as shown in Bilen et al (2016).

Comparison with other self-supervision methods. To compare with previous self-supervision methods (Misra et al 2016; Fernando et al 2017) that have used AlexNet as the backbone architecture and fine-tuned with all layers, we include fine-tuning results for models pre-trained using AoT on UCF101 split-1 for AlexNet with RGB or D-RGB inputs. In Table 9, our AoT results significantly outperform the prior art.

7.2 Video forensics: reverse film detection

Reverse action is a type of special effect in cinematography where the action that is filmed ends up being shown backwards on screen. Such techniques not only create artistic scenes that are almost impossible to make in real life (e.g. broken pieces coming back together), but also make certain effects easier to realize in the reverse direction (e.g. targeting a shot precisely). Humans can often detect such techniques, as the motion in the video violates our temporal structure prior of the world (e.g. the way people blink their eyes or steam is emitted from an engine). For this video forensics task, we tested the T-CAM model trained on the Flickr-AoT and Kinetics-AoT datasets with 10 frames of flow input, as some clips have fewer than 20 frames.

	Chance	Flow-Word	T-CAM (ours)		Human
			Flickr	Kinetics	
Acc.	50%	58%	76%	72%	80%

Table 10: AoT test accuracy on the Reverse Film dataset. The T-CAM model pre-trained on either Flicker-AoT or Kinetics-AoT outperforms the Flow-Word method (Pickup et al 2014), and is closer to human performance.

Reverse Film Dataset. We collected clips from Hollywood films which are displayed in reverse deliberately. Thanks to the “trivia” section on the IMDB website, shots that use reverse action techniques are often pointed out by the fans as Easter eggs. With keyword matching (e.g. “reverse motion”) and manual refinement on the trivia database, we collected 67 clips from 25 popular movies, including ‘Mary Poppins’, ‘Brave Heart’ and ‘Pulp Fiction’. See the project page for the movie clips and more analysis of the common cues that can be used to detect the arrow of time.

Classification and localization results. As can be seen in Table 10, the overall test accuracy of the T-CAM model is 76% (trained on Flickr-AoT) and 72% (trained on Kinetics-AoT), where human performance (using Amazon Mechanical Turk) is 80%, and the baseline model (Pickup et al 2014) achieves 58%. In Figure 9, we visualize both successful and failure cases, and show the T-CAM heatmap score of being backward in time. The successful cases are consistent with our earlier finding that the model learns to capture both low-level cues such as gravity (Figure 9a,c) and entropy (Figure 9b), as well as high-level cues (Figure 9d-e), and . For the failure cases, some are due to the symmetric nature of the motion, e.g. wheel rotation (Figure 9f).

8 Summary

In this work, we manage to learn and use the prevalent arrow of time signal from large-scale video datasets. In terms of *learning* the arrow of time, we design an effective ConvNet and demonstrate the necessity of data pre-processing to avoid learning artificial cues. We develop two large-scale arrow of time classification benchmarks, where our model achieves around 80% accuracy, significantly higher than the

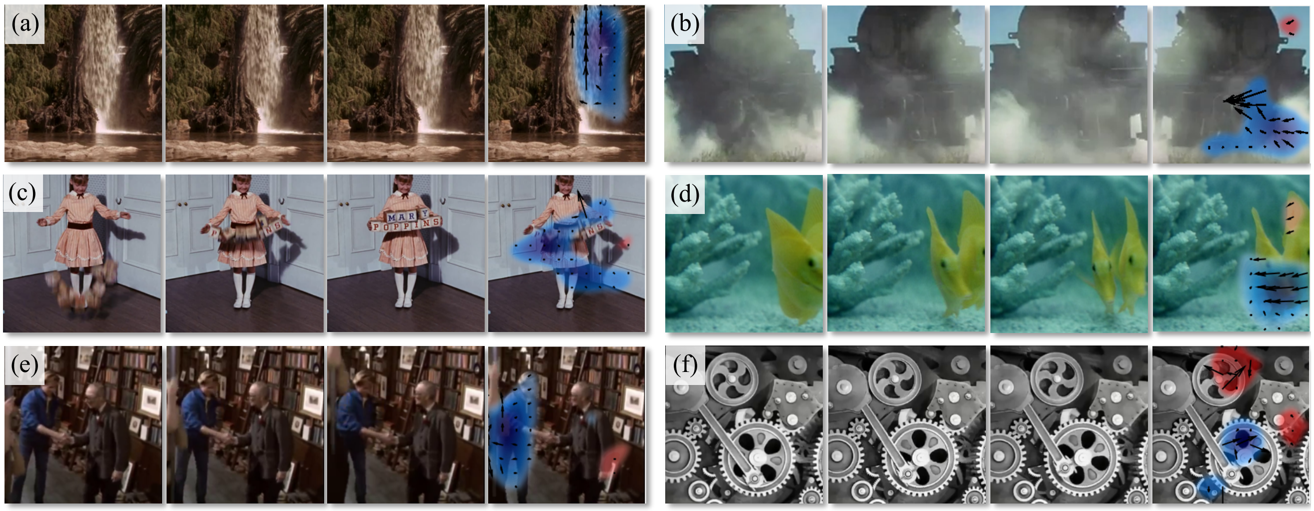


Fig. 9: Example results from our Reverse Film dataset–short clips appearing time-reversed in Hollywood movies. For each example, we show four images: three frames from the input clip for our T-CAM model in their displayed order in the movie, and the class activation map with sparse motion field overlaid on the middle frame. As all the clips are played in the reverse direction, the ground truth class activation map color is blue. On examples that our T-CAM model classifies AoT correctly and confidently, the model exploits both low-level physical cues, e.g. (a) water falls, (b) smoke spreads, and (c) block falls and spreads; and high-level cues, e.g. (d) fish swim forwards, and (e) human action. The T-CAM model is unconfident about a motion that can be intrinsically symmetric, e.g. (f) wheel rotation.

previous state-of-the-art method at around 60%, and close to human performance. In addition, we can identify the parts of a video that most reveal the direction of time, which can be high- or low-level visual cues.

In terms of *using* the arrow of time, our model outperforms the previous state-of-the-art on the self-supervision task for action recognition, and achieves 76% accuracy on a new task of reverse film detection, as a special case for video forensics.

Acknowledgments. This work was supported by NSF Grant 1212849 (Reconstructive Recognition), and by the EPSRC Programme Grant Seebibyte EP/M013774/1.

References

Bilen H, Fernando B, Gavves E, Vedaldi A, Gould S (2016) Dynamic image networks for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3034–3042 10

Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp 4724–4733 3

Chung JS, Zisserman A (2016) Lip reading in the wild. In: ACCV 8

Dekel T, Moses Y, Avidan S (2014) Photo sequencing. IJCV 2

Doersch C, Gupta A, Efros AA (2015) Unsupervised visual representation learning by context prediction. In: ICCV 1

Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: CVPR 3

Fernando B, Gavves E, Oramas JM, Ghodrati A, Tuytelaars T (2015) Modeling video evolution for action recognition. In: CVPR 2

Fernando B, Bilen H, Gavves E, Gould S (2017) Self-supervised video representation learning with odd-one-out networks. In: CVPR 2, 9, 10

Fragkiadaki K, Agrawal P, Levine S, Malik J (2015) Learning visual predictive models of physics for playing billiards. In: ICLR 4

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML 3

Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P, Suleyman M, Zisserman A (2017) The kinetics human action video dataset. arXiv preprint arXiv:170506950 2, 6

Lin M, Chen Q, Yan S (2013) Network in network. In: ICLR 3

Liu Z, Yeh R, Tang X, Liu Y, Agarwala A (2017) Video frame synthesis using deep voxel flow. In: ICCV 9

Misra I, Zitnick L, Hebert M (2016) Shuffle and learn: Unsupervised learning using temporal order verification. In: ECCV 2, 9, 10

Pickup LC, Pan Z, Wei D, Shih Y, Zhang C, Zisserman A, Scholkopf B, Freeman WT (2014) Seeing the arrow of time. In: CVPR 1, 2, 4, 6, 7, 10

Ramanathan V, Tang K, Mori G, Fei-Fei L (2015) Learning temporal embeddings for complex video analysis. In: ICCV 2

Schölkopf B, Janzing D, Peters J, Sgouritsa E, Zhang K, Mooij J (2012) On causal and anticausal learning. In: ICML 2

Simonyan K, Zisserman A (2014a) Two-stream convolutional networks for action recognition in videos. In: NIPS 3, 9

Simonyan K, Zisserman A (2014b) Very deep convolutional networks for large-scale image recognition. In: ICLR 3

Soomro K, Zamir AR, Shah M (2012) UCF101: A dataset of 101 human actions classes from videos in the wild. In: arXiv preprint arXiv:1212.0402 4

Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2014) Striving for simplicity: The all convolutional net. In: ICLR 3

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: CVPR 3

Thomee B, Shamma DA, Friedland G, Elizalde B, Ni K, Poland D, Borth D, Li LJ (2016) Yfcc100m: The new data in multimedia research. Communications of the ACM 59(2):64–73 2, 6

Vondrick C, Pirsiavash H, Torralba A (2016) Generating videos with scene dynamics. In: NIPS 6

Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: towards good practices for deep action recognition. In: ECCV 3, 9

Xie S, Sun C, Huang J, Tu Z, Murphy K (2017) Rethinking spatiotemporal feature learning for video understanding. arXiv preprint arXiv:171204851 2

Zach C, Pock T, Bischof H (2007) A duality based approach for realtime tv-l 1 optical flow. In: JPRS 3

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2014) Object detectors emerge in deep scene CNNs. In: ICLR 2

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: CVPR 3, 7

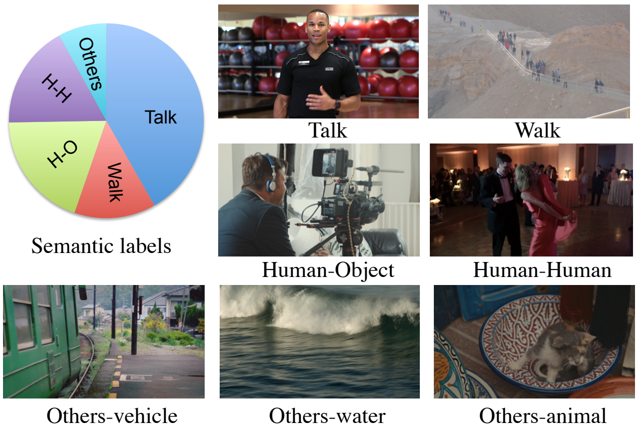


Fig. 10: The MJPEG AoT dataset. It contains around 16.9k clips without inter-frame video coding from Vimeo. (a) ratio of videos for each of the five semantic categories: walk, talk, human-object interaction (H-O), human-human interaction (H-H), and others (e.g. water, animals, vehicle); (b) sample frames from the dataset.

Appendices

A Dataset Details

In the main submission, we construct three datasets of real videos, MJPEG-AoT, Flickr-AoT and Kinetics-AoT, to study the arrow of time classification problem. Below, we describe the details of our video collection for MJPEG-AoT and video pre-processing for all three datasets.

A.1 Video Collection for MJPEG-AoT

Unlike Youtube, where videos are mosty H.264 compressed, Vimeo[3]hosts many professional videos in a variety of original formats, which are known. To download videos without temporal codec compression, we search on Vimeo with keywords such as “mjpeg”, “prores”, and “cannon+raw” etc. We verify the codec of the downloaded video with “ffmpeg”. We initially obtain around 7,000 videos before pre-processing. To show the diversity of the videos, we use Amazon Mechanical Turk to label them into five categorie: talk, walk, human-object interaction, human-human interaction, and others (Figure 10).

A.2 Video Pre-processing

We pre-process the videos as follows:
Black frame removal: For each video, we select five uniformly spaced frames and compute the mean RGB value for each row and column. To remove the black frame, we find the first or last consecutive rows or columns whose mean value are less than our manually set threshold.
Shot detection: We compute the frame difference for each video and save the mean RGB difference for each frame. Then, we select 41-frame-long clips whose mean frame differences are not 0 (e.g. static frames) and are not large (e.g. shot transition or fast camera motion).
Clip stabilization: For each 41-frame-long clip, we first compute the homography between each frame and the central frame independently, and then smooth the estimated homography with outlier rejection. We stabilize each clip with these estimated homographies.
Human selection: As a final check on quality, Amazon Mechanical Turk is used to remove clips with either black stripes, multiple-shots or an unstabilized camera. Then, we manually refine the selection for the final round.

B Artificial Cues: Controlled Experiments

Here we provide experimental details regarding the effect of black framing and camera motion for the arrow of time prediction (Section 4). We train the T-CAM model on two versions of the same dataset, one original (A) and one with identified artificial signals removed (A^*). If the model trained on A has significantly better test result on A than A^*, then it is likely that this model does learn to rely on artificial signals for predictions. If both models have similar test accuracy on A^*, then it is likely that the artificial signal removal procedure doesn't introduce new biases.
Black Framing. Given a video with black framing, we can remove the artificial signals in either of two ways: zero out

[3] http://vimeo.com

Train/Test	original	zero-out	crop-out
original	**98.1%**	87.9%	**90.3%**
zero-out	88.1%	89.9%	87.6%
crop-out	86.4%	86.5%	90.5%

(a) black framing

Train/Test	original	stabilization
original	**88.3%**	**75.2%**
stabilization	70.7 %	78.4%

(b) camera motion

Table 11: An examination of the artificial cues for arrow of time prediction. We show results of controlled experiments on the effect of black framing and camera motion on UCF101 (bold numbers are used in the main submission).

the flow values in the black bar region or crop out the black bar region. Given the three versions of the UCF-101 dataset, we train and test our T-CAM model for all nine possible combinations of train and test sets (Table 11a).

Consistently, the model trained on "original" data has significantly higher test accuracy on "original" test data, implying that the black framing is significantly contributing to the video direction classification. In contrast, all three models have similar test accuracies on either version of the modified data, suggesting both removal procedures are effective to avoid black frame signals. Thus, to avoid learning the artificial signals from black frame, we can either zero-out or crop-out the corresponding flow maps regions.

Camera Motion. Out of the 13.3k videos from UCF101, we select 9.6k videos which can be well-stabilized within a chunk of 41 frames. To avoid the effect of black framing, we crop out the black bar region of the videos. We train-test on original videos and the stabilized ones (Table 11b).

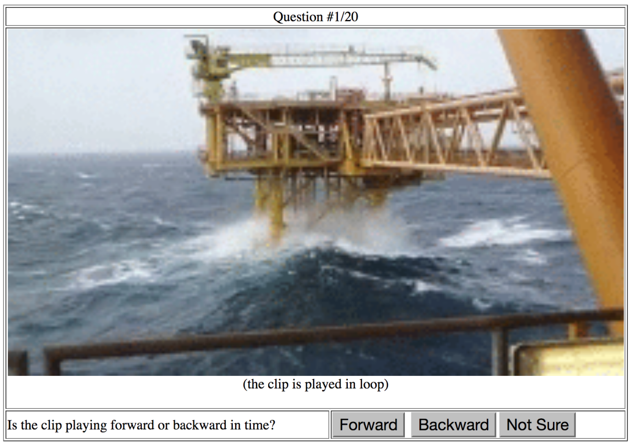
The T-CAM model trained on 'original" videos has significantly worse performance on the stabilized videos, where camera motion cues (e.g. zoom-in) not longer exist. On the other hand, both models have similar test accuracy on the stabilized videos, suggesting the stabilization method is effective to cancel out the camera motion bias. Thus, to avoid learning the artificial camera motion bias, we can stabilize the videos.

C Human Performance

In Table **??**, we show human performance on MJPEG-AoT and TA180 dataset through Amazon Mechenical Turk (AMT). Below, we describe the details of the experiment design and the interface design.

Experiment Design. Similar to the input duration of our ConvNet model, we show an animated gif with only 10 frames (around 0.4 sec) either in forward or backward direction in a loop. The test subject can either predict the direction of the displayed clip or choose "not sure". To control the quality of the experiment, we add tests inside each AMT job where the arrow of time signal is obvious (e.g. water falls). For each clip, we ask five different AMT workers and report the average accuracy.

Interface. We show the browser interface of our AMT job in Figure 11.



Question #1/20

(the clip is played in loop)

Is the clip playing forward or backward in time?

Fig. 11: Browser interface for our Amazon Mechanical Turk study to obtain human performance on the arrow of time prediction.

D Reverse Film Detection

D.1 Common Cues

To understand the common cues used by the T-CAM model to detect the reverse-playing movie clips, we find clips with high prediction confidence, cluster their last layer feature and manually associate these clusters with interpretable cues. In Figure 12, we show three clips (all played in the backward direction) from each of three identified motion cues: human head, gravity and human body. For each 10-frame clip, we show its first, middle and last frame and its prediction heatmap (red for "forward" and blue for "backward") overlayed on the middle frame with flow vector in the confident regions.

Human head cluster (Figure 12a) the T-CAM model achieves high accuracy and the heatmaps are dominated by blue color in the head region. The first and the third row have global head motion while the second row shows eye movement.

Gravity cluster (Figure 12b) the T-CAM model is correctly confident in the region where either water or snow is moving upward in certain patterns against gravity.

Human body cluster (Figure 12c) the T-CAM model achieves around chance accuracy, where the heatmap color varies for different human motion patterns. Some motion is indicative to human for backward arrow of time (e.g. stepping backward motion of the man behind in the first row), while some motion can be subtle (third row) . Notably, the second row features Charlie Chaplin's performance which intentionally tries to make his backward motion seem natural and successfully fools our T-CAM model.

D.2 Full list of videos

The 67 reverse action clips are collected from the following 23 movies (in chronological order): Demolition d'un mur (1896), Modern Times (1936), Shane (1953), The Ten Commandments (1956), Mary Poppins (1964), The Rounders (1965), Butch Cassidy and the Sundance Kid (1969), Chisum (1970), Superman (1978), Star Wars V: The Empire Strikes Back (1980), Unknown Chaplin (1983), Top Secret! (1984), Evil Dead II (1987), Raising Arizona (1987), Pulp Fiction (1994), Brave Heart (1995), Anaconda (1997), A Life Less Ordinary (1997), Bringing Out The Dead (1999), Memento (2000), The Railway Children (2000), 2 Fast 2 Furious (2003), Sin City (2005).

Notably, 'Top Secret!' has a 1.5-minute long single shot of reverse action!

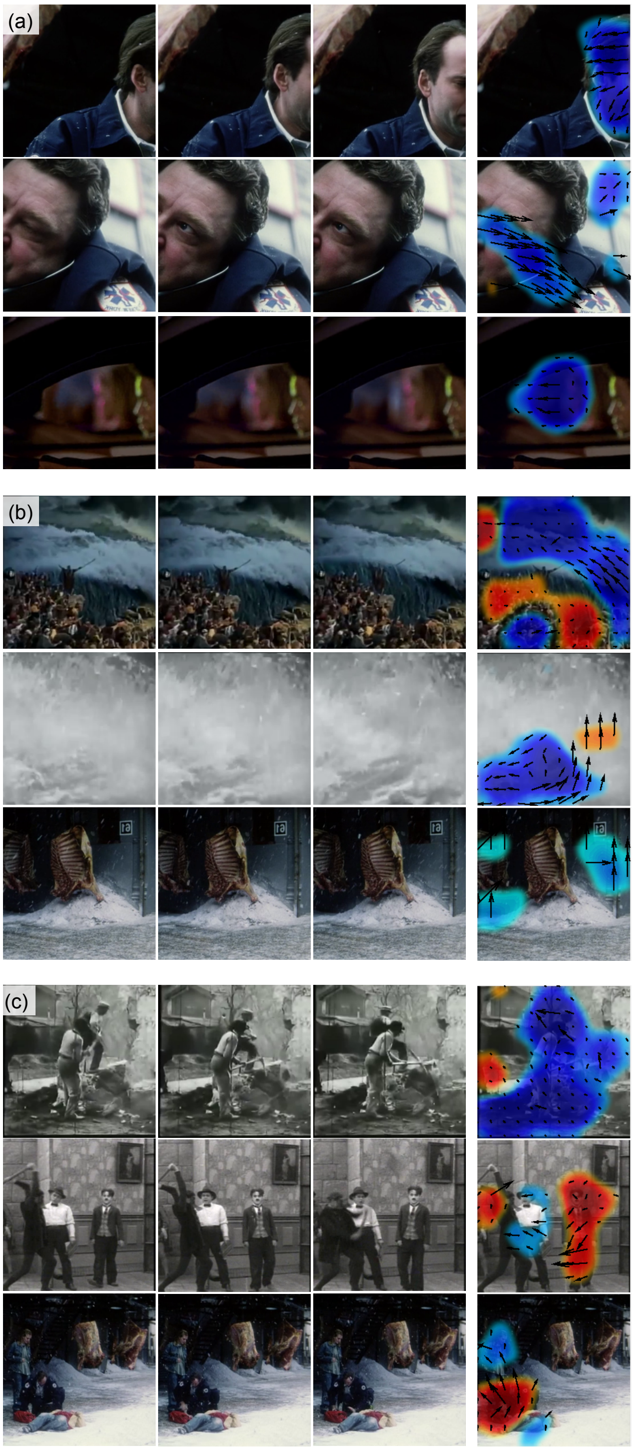


Fig. 12: Common cues for reverse film detection. In addition to results in Figure 9, the T-CAM model consistently focuses on regions with (a) head motion, (b) motion against gravity, and (c) human body motion. For (c), the T-CAM model can be fooled by professional backward-acting (second row) and subtle motion (third row) where red regions are around performers.