Retro-Actions: Learning 'Close' by Time-Reversing 'Open' Videos

Will Price
University of Bristol
will.price@bristol.ac.uk

Dima Damen
University of Bristol
dima.damen@bristol.ac.uk

Abstract

We investigate video transforms that result in class-homogeneous label-transforms. These are video transforms that consistently maintain or modify the labels of all videos in each class. We propose a general approach to discover invariant classes, whose transformed examples maintain their label; pairs of equivariant classes, whose transformed examples exchange their labels; and novel-generating classes, whose transformed examples belong to a new class outside the dataset. Label transforms offer additional supervision previously unexplored in video recognition benefiting data augmentation and enabling zero-shot learning opportunities by learning a class from transformed videos of its counterpart.

Amongst such video transforms, we study horizontal-flipping, time-reversal, and their composition. We highlight errors in naively using horizontal-flipping as a form of data augmentation in video. Next, we validate the realism of time-reversed videos through a human perception study where people exhibit equal preference for forward and time-reversed videos. Finally, we test our approach on two datasets, Jester and Something-Something, evaluating the three video transforms for zero-shot learning and data augmentation. Our results show that gestures such as 'zooming in' can be learnt from 'zooming out' in a zero-shot setting, as well as more complex actions with state transitions such as 'digging something out of something' from 'burying something in something'.

1. Introduction

Without temporal ordering, individual frames from a video clip of an '*open jar*' action cannot be distinguished from frames of a '*close jar*'. Tampering with the temporal order, whether through shuffling or reversing the order of frames, has been frequently used to assess the utilisation of temporal signals in action recognition models [14, 29, 30]. Recent convolutional models [3, 25, 27, 29, 30] demonstrate increased robustness by explicitly modelling temporal relations in video. In a related problem, Arrow of Time (AoT) classification [9, 23, 28] (the task of determining whether a video is being played forwards or backwards) has been used for pretraining video understanding models.

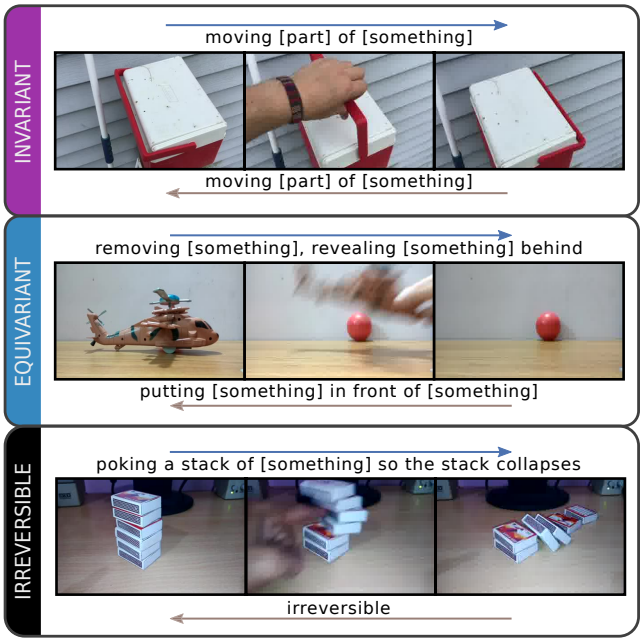

Figure 1: When time-reversing a video, *invariant* actions (top) maintain their label, *equivariant* actions (middle) exchange labels, while some actions (bottom) are *irreversible* producing motions that defy laws of physics.

In this work, we apply the time-reversal video transform on videos to produce new ones that cannot be differentiated from forward-time videos by a human observer. We validate the realism of these examples through a forced-choice human perception study. We observe that when time-reversed, reversible videos either maintain their label or undergo a label transformation (Fig. 1). We develop a technique for automatically extracting this label transform for each class from the predictions of a trained classification model. Next, we apply our findings to other video transforms: horizontal-flipping and the composition of time-

reversal with horizontal-flipping. We then put label transforms to work in zero-shot learning and data augmentation. Our contributions are summarised as follows:

- We introduce label-altering video transforms, and identify their corresponding label transforms from model predictions.
- We evaluate our proposal on two datasets, demonstrating the efficacy of example synthesis for both zero-shot learning and data augmentation.
- Our zero-shot learning results demonstrate novel opportunities for learning additional classes through video transforms. On Something-Something, we learn 16 zero-shot classes i.e. *without a single example* (out of 174 total classes), and report 46.6% accuracy compared to 49.5% with full supervision. On Jester, we learn 7 zero-shot classes (out of 27 total classes), and report 92.4% accuracy compared to 94.9% with full supervision.

2. Related work

In this section, we review relevant works to our proposal related to: 1) the rise in *temporally-sensitive* video recognition models, 2) using time reversal in video and 3) using video transforms for self-supervision. To the best of our knowledge, no prior work has investigated label-altering video transforms for the automatic synthesis of additional labelled training data.

Action recognition. Action recognition is the task of classifying the action demonstrated in a trimmed video segment. Classification in early video action recognition datasets [18, 24] has been shown to be solvable largely through visual appearance alone [14, 30]. These datasets have been supplanted by larger and more temporally challenging datasets [4, 11, 12, 17, 21] where this is no longer the case. This gave rise to papers questioning the ability of both convolutional and recurrent models to capture the temporal order or evolution of the action [5, 8, 13, 14, 29, 30]. For example, in [14] a C3D network trained with hallucinated motion and a single frame from the video is shown to perform comparably to the original video.

Accompanying this evolution has been an increased focus in proposing models that exploit temporal signals in video [3, 6, 25–27, 29, 30]. In [26], actions are modelled as state transformations, showing improved performance and better generality across actions. Zhou *et al*. [30] introduce a dedicated layer to correlate the predictions of multiple temporally-ordered video segments, averaging over multiple temporal scales. The model's ability to exploit *time* is tested by shuffling frames in the video. They report no drop in performance for UCF101, but a clear degradation on Something-Something [11] showing the latter is more suitable for learning and evaluating temporal features.

Time-reversal in video. Time-reversing videos is used for Arrow of Time (AoT) classification [9, 23, 28]. First introduced in [23] and recently revisited in [28], AoT classification is successfully used in self-supervision for pre-training action recognition models. Of particular relevance to our work is the human perception study of time-reversed videos on Kinetics by Wei *et al*. [28], showing humans achieve a 20% error-rate classifying a video's AoT, thus demonstrating that dataset subsets contain realistic videos when reversed.

Video transforms for self-supervision. Video transforms offer a form of self-supervision [2, 7, 16, 20, 28]. In [2], a video-jigsaw solving task is used for pre-training before fine-tuning for action recognition, and in [16] geometric rotation classification is used for pre-training. In all these works, video transforms are only used in a separate task from which knowledge is transferred to the target task. The only prior work that has used video transforms for what could be seen as zero-shot learning is [22]. They utilise time-reversal for training a robot arm to put two blocks together by observing these blocks exploding apart.

3. Label-altering video transforms

In this section we introduce label-altering (video) transforms (LATs) and describe how their corresponding class transforms can be determined from predictions of trained models.

Introducing LATs. Given an oracle video labelling function f and a dataset with videos V and labels $Y = \{f(v) \,|\, v \in V\}$, we aim to learn the parameters of a model $\hat{f}$ using the videos V and the supervision Y. We define a video transform T as an operation that takes a video $v \in V$ and transforms it into another video $\hat{v} = \mathrm{T}(v)$ that is a valid input to the trainable model $\hat{f}$. We restrict our study to video transforms that satisfy the self-inversion property $(\mathrm{T} \circ \mathrm{T})(v) = v$, and distinguish between two types: *label-preserving* video transforms (LPTs), and *label-altering* video transforms (LATs). In LPTs, the mapping between a video and its label remains intact

$$\forall v \in V : f(v) = y \Leftrightarrow f\big(\mathrm{T}(v)\big) = y, \tag{1}$$

however in LATs, the video transforms which we are interested in, result in a label change such that

$$\exists v \in V : f(v) = y \Rightarrow f\big(\mathrm{T}(v)\big) \neq y. \tag{2}$$

Of all possible LATs, we are interested in ones where the application of the video transform T to every example of a given class results in transformed labels belonging to the same class, we call these *class homogeneous* LATs:

$$\forall \{v, w\} \subset V : f(v) = f(w) \Rightarrow f\big(\mathrm{T}(v)\big) = f\big(\mathrm{T}(w)\big). \tag{3}$$

Without class homogeneity, new ground-truth of all transformed videos would be required. However, when class homogeneity is preserved, class transforms are sufficient to label all transformed videos. Accordingly, for a class homogeneous LAT, we aim to define the corresponding class transform $\mathrm{T}_y(y)$ for all $y \in Y$ where possible. Given $V_y = \{v \in V \,|\, f(v) = y\}$, we identify three categories of classes:

1. *Invariant classes*, Y_i: classes whose examples maintain their label after transformation

$$Y_i = \left\{y \in Y \,|\, \forall v \in V_y : f\big(\mathrm{T}(v)\big) = y\right\}. \quad (4)$$

   The class transform for invariant classes can thus be defined: $y \in Y_i \Rightarrow \mathrm{T}_y(y) = y$.

2. *Equivariant*, Y_e: classes whose examples change label after transformation

$$Y_e = \{y \in Y \,|\, \exists y' \in Y \; \forall v \in V_y : f\big(\mathrm{T}(v)\big) = y' \neq y\}. \quad (5)$$

   We thus define $\mathrm{T}_y(y) = y'$, referring to (y, y') as a pair of equivariant classes where y' is the *counterpart* of y and vice versa. Since we desire T_y to be equivariant to T, we restrict T_y to be self-invertible, in line with the self-invertible behaviour of T:

$$\forall y \in Y_e : \mathrm{T}_y\left(\mathrm{T}_y(y)\right) = y. \quad (6)$$

3. *Novel-generating*, Y_n: these include classes whose transformed examples no longer belong to any of the dataset's classes Y. We revisit these classes later, using them for zero-shot learning.

$$Y_n = \{y \in Y \,|\, y \notin Y_i \cup Y_e\}. \quad (7)$$

3.1. Discovering class transforms

In order to automatically determine the class transform T_y, we propose a method based on the response of the trained model $\hat{f}$ to all videos from the same dataset transformed by T. We first calculate the recall of each class y using the model $\hat{f}$. We define

$$\hat{V}_y = \{v \in V \,|\, \hat{f}(v) = f(v) = y\}, \quad (8)$$

and measure the class recall, $\Lambda(y|\hat{f}) = |\hat{V}_y|/|V_y|$. If $\Lambda(y|\hat{f}) \geq \lambda$ (*i.e.* the model performs sufficiently well on that class), the model can be used to establish the class transform $\hat{\mathrm{T}}_y(y)$, assuming minimal noise exists in the dataset labels. Conversely, if $\Lambda(y|\hat{f}) < \lambda$, the class transform cannot be established for y from predictions of the model $\hat{f}$. We then calculate the proportion of videos in $\hat{V}_y$ that are predicted as y' when T is applied

$$\Gamma(y, y'|\hat{f}, \mathrm{T}) = \left|\left\{v \in \hat{V}_y \,|\, \hat{f}\big(\mathrm{T}(v)\big) = y'\right\}\right| / \left|\hat{V}_y\right|, \quad (9)$$

and measure affinity between the two classes

$$\Omega(y, y'|\hat{f}, \mathrm{T}) = \Gamma(y, y'|\hat{f}, \mathrm{T})\Gamma(y', y|\hat{f}, \mathrm{T}). \quad (10)$$

We calculate a candidate target class y_t per class y:

$$y_t = \arg\max_{y'} \Omega(y, y'|\hat{f}, \mathrm{T}). \quad (11)$$

and introduce a novel target y_n for the class. Finally, the approximated class transform $\hat{\mathrm{T}}_y$ is:

$$\hat{\mathrm{T}}_y(y) = \begin{cases} y & \Omega(y,y) \geq \alpha \\ y_t & \Omega(y,y_t) \geq \alpha \wedge \Omega(y,y) < \alpha \wedge \Omega(y_t,y_t) < \alpha \\ y_n & \text{otherwise.} \end{cases} \quad (12)$$

where α controls the trade off between extracting invariant and equivariant transforms.

3.2. Applications of class tranforms

Next, we describe how class homogenous LATs with their class transforms (T_y) can be used for data augmentation and zero-shot learning.

Data augmentation. LPTs have long been used for data augmentation and range from the simple, like adjusting the frame rate of a video, to the complex, like the learnt transformations used in adversarial training [10]. We propose using LATs for augmenting both invariant and equivariant classes through target-conditional data augmentation

$$V_y^{\mathrm{aug}} = V_y \cup \{\mathrm{T}(v) \,|\, v \in V_{y'} \wedge \mathrm{T}_y(y') = y \in Y\} \quad (13)$$

Zero-shot learning. The novel-generating (NG) classes of T facilitate zero-shot learning by synthesising examples of a novel class y as follows:

$$V_y^{\mathrm{zs}} = \{\mathrm{T}(v) \,|\, v \in V_{y'} \wedge y' \in Y_n \wedge \mathrm{T}_y(y') = y\} \quad (14)$$

The model $\hat{f}$ is trained with synthesised examples V_y^{zs} of the zero-shot class y and tested on real examples.

3.3. LAT examples

We apply the generalisation above on two LATs, as well as their composition (Fig. 2):

Transform 1: horizontal-flipping. While in some video datasets, horizontal-flipping is a LPT, it is a LAT when the dataset includes classes with a defining uni-directional horizontal movement, *e.g.* '*swipe right*' or '*rotate clockwise*'.

Transform 2: time-reversal. Unlike horizontal-flipping, time-reversal is a fairly new transform used by the community. Whilst many classes in action datasets are irreversible, we show that a subset of these maintain realism under time-reversal—an observation that has received little attention. For example, time reversing an action such as '*cover*' reverses the state change to produce an '*uncover*' action. We

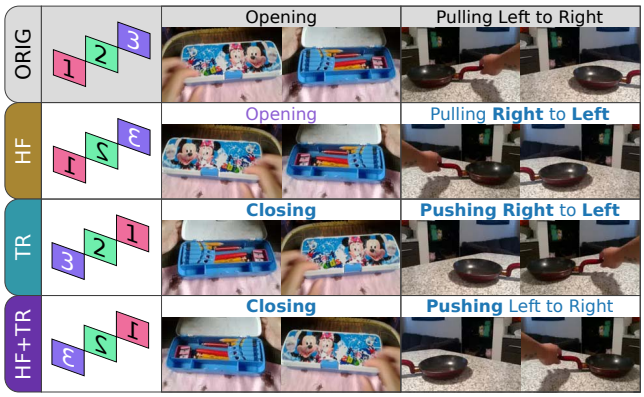

Figure 2: Class transforms for horizontal-flipping, time-reversal, and their composition for two videos from Something-Something (*bold indicates label changes*).

note that many classes invariant under horizontal-flipping become equivariant under time-reversal (Fig. 2). A number of classes can't be mapped to semantically meaningful classes after the transform as a result of the irreversibility of their examples.

What makes a video irreversible? We find the realism of reversed videos to be betrayed by *reversal artefacts*, aspects of the scene that would not be possible in a natural world. Some artefacts are subtle, while others are easy to spot, like a reversed '*throw*' action where the thrown object spontaneously rises from the floor. We observe two types of reversal artefacts, *physical*, those exhibiting violations of the laws of nature, and *improbable*, those depicting a possible but unlikely scenario. These are not exclusive, and many reversed actions suffer both types of artefacts, like when uncrumpling a piece of paper. Examples of physical artefacts include: inverted gravity (*e.g.* '*dropping something*'), spontaneous impulses on objects (*e.g.* '*spinning a pen*'), and irreversible state changes (*e.g.* '*burning a candle*'). An example of an improbable artefact: taking a plate from the cupboard, drying it, and placing it on the drying rack.

Transform 3: horizontal flipping + time reversal. We also explore the composition of the two transforms above. This not only offers new opportunities for data augmentation and zero-shot learning, but also removes some of the biases from the dataset or model. For example, we note that motion blur affects zero-shot learning when using time-reversal. Combining both transforms removes the model's bias. Similarly, when a dataset is biased (e.g. more right-handed than left-handed people in our datasets), this composition assists in balancing the dataset.

4. Datasets and perception study

To showcase how LATs can be utilised for action recognition, we use two large-scale crowd-sourced datasets. **Jester** [1] is a gesture-recognition dataset with 148k videos split into 119k/15k/15k for training/validation/testing with 27 classes (*e.g.* '*sliding two fingers down*', '*thumb up*'). **Something-Something (v2)** [11] is an object interaction dataset containing 221k videos split into 169k/25k/27k for training/val/testing with 174 classes (*e.g.*'*taking something out of something*', '*tearing something a little bit*').

Dataset	Transform	# Invariant	# Equivariant	# Novel-generating	
				Realistic	Unrealistic
Jester	Horizontal-flip	21	6	0	0
Jester	Time-reverse	8	14	5	0
Something	Horizontal-flip	168	6	0	0
Something	Time-reverse	34	32	28	80

Table 1: Transform class category counts for the ground truth T_y defined on horizontal-flipping and time-reversal. Note the increased number of equivariant and novel-generating classes of time-reversal compared to horizontal-flipping.

Class transforms. We manually define a class transform T_y for each LAT; this is used as ground truth for both the assessment of the automated discovery of $\hat{\mathrm{T}}_y$, and in evaluating its applications. We obtain this through inspection of class semantics followed by visual verification. For horizontal-flipping, we map pairs of classes with defining horizontal motions (e.g. '*left to right*') to one another and map other classes to themselves. For time-reversal, we consider what motions and state changes are reversed and how these interact across classes, then examine reversed examples checking for reversal artefacts that prevent otherwise reasonable mappings from being defined.

Table 1 shows the number of classes within each category for the ground truth T_y. As the table shows, time-reversal results in more equivariant classes than horizontal-flipping. We find 5 and 28 novel-generating reversible classes in Jester and Something-Something where the transformed label is not part of the label set (*e.g.* '*putting S underneath S*' has no counterpart '*taking S from underneath S*', *S* = something).

Arrow of Time: perception study Before attempting to use time-reversal as a video transform in our applications, we crowd-sourced a human perception study to confirm the similarity between forward-time and reversed-time examples of our reversible classes. In Table 1, we highlight (in blue) the 22 classes from Jester and 66 classes from Something-Something that we deemed time-reversible and on which we conducted this study.

Participants were asked to select the better example of two videos in a forced-choice setup (UI shown in Fig. 3). They were not given any further instructions of what makes a video a better/worse example of the class, and were not informed that one video was time-reversed. In each pair, one of the videos was randomly sampled from the training set for that class, while the other was a reversed video sampled from the training set of the label-transformed class. We randomised the left-right placement of videos.

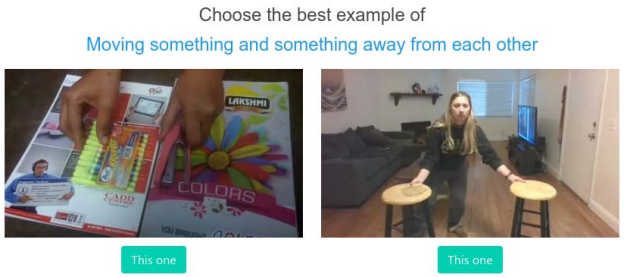

Figure 3: AMT UI showing an unaltered/time-reversed video.

We used Amazon Mechanical Turk (AMT) for the study, testing 20 video pairs in each task. In k video pairs, the reversed video was replaced with a forward-time video from an unrelated class as a way to filter out low quality annotations. We used $k = 3$ in Jester and $k = 5$ in Something-Something, only accepting submissions that correctly chose 3/3 and 3/5 of these examples respectively. The bar was set lower for Something-Something due to overlapping classes and occasional low video quality. In total, 257 individuals annotated 200 videos per class in Jester, and 120 videos per class in Something-Something amounting to 5.8% and 10.4% of videos in the reversable class subsets.

To determine which classes are reversible, we model the results for each class as a binomial distribution with $p = 0.5$ approximated by a normal distribution. We consider classes reversible if their forward-time preference is within $\mu \pm 3\sigma$. We present the results of this study in Fig. 4, showing all classes in Jester are within bounds, and only 2 are outside for Something-something (both invariant). The class with the largest preference for forward-time is '*pretending to throw something*' which exerts asymmetric impulses that participants seem to detect when time-reversed.

Having confirmed that reversed-time examples were sufficiently similar to forward-time ones in our chosen classes, we move on to using these time-reversed examples in zero-shot learning and data augmentation.

5. Experiments and results

Following a description of implementation details, we examine the behaviour of the network when exposed to transformed videos, and evaluate our method to automate class transforms (Section 5.1). We then present experiments using LATs for zero-shot learning (Section 5.2) and data augmentation (Section 5.3).

Implementation details. We employ a Temporal Relational Network (TRN) [30] with a batch-normalised Inception (BNInception) backbone [15] trained on RGB video due to its temporal-sensitivity, computational efficiency through sparse sampling, and high performance on benchmark datasets (including those we test on). In TRNs, the input video is split into n segments from which a frame is randomly sampled. Segment features, extracted by the backbone network, are combined by a FC layer to compute temporal relations, followed by class predictions.

Dataset	Transform	λ	α	TP	FP	FN	TN
Jester	HF	0.90	0.80	24	0	2	1
Jester	TR	0.90	0.80	22	1	0	4
Something	HF	0.04	0.09	172	0	2	0
Something	TR	0.04	0.06	59	69	7	39

Table 2: Evaluation of $\hat{\mathrm{T}}_y$ compared to ground truth T_y.

We first replicated the validation set results reported by the authors [30] and assessed the effect of multi-scale model variant and number of segments before settling on 8-segment single-scale TRN for our experiments. We restrict our model evaluations to single center-crops to avoid unintended label transformations introduced by horizontal-flipping. In all experiments, we train our networks for 100 epochs with an initial learning rate of 1×10^{-3} divided by 10 at epochs 40 and 80. We use a batch size of 80 for Jester and 128 for Something-Something training on 4 GPUs. All other parameters follow the default values from the TRN GitHub codebase. We report all our results on the validation set of both datasets.

5.1. Discovering class transforms

This first experiment assesses how a model trained on forward-time videos responds to video transforms, with and without label transformation. We show the confusion matrices for each LAT in Fig. 5. For each dataset, we show the baseline performance and the performance after horizontal-flipping or time-reversal without (red) and with (green) label transformation (using the manually defined ground truth label transform). For easier viewing, we re-order classes so equivariant class pairs are adjacent. These figures show that equivariant classes are misclassified into their counterparts without the application of LTs and that when employed, LTs resolve this misclassification whilst maintaining the correct classification of invariant classes.

One case worthy of note relates to the confusion between '*turning hand clockwise*' and '*turning hand counterclockwise*' in Jester. Horizontal flipping with LT increases the confusion, which we believe is a result of a population bias towards right-handed people; in a right-handed clockwise hand turn, the back of the hand is shown first then the front, whereas the order is reversed for a left-handed person.

Having shown the base model's response to video transforms matches the manually defined ground truth label transform, we evaluate our method for automatically extracting $\hat{\mathrm{T}}_y$ through the process described in Section 3.1. In Table 2, we report true/false positives/negatives for the λ, α that maximises the true positive count when treating the extraction of a mapping $y \rightarrow y'$ as a binary classification task. Note that the optimal λ, α seem to be independent of the transform, and only different for the dataset/model.

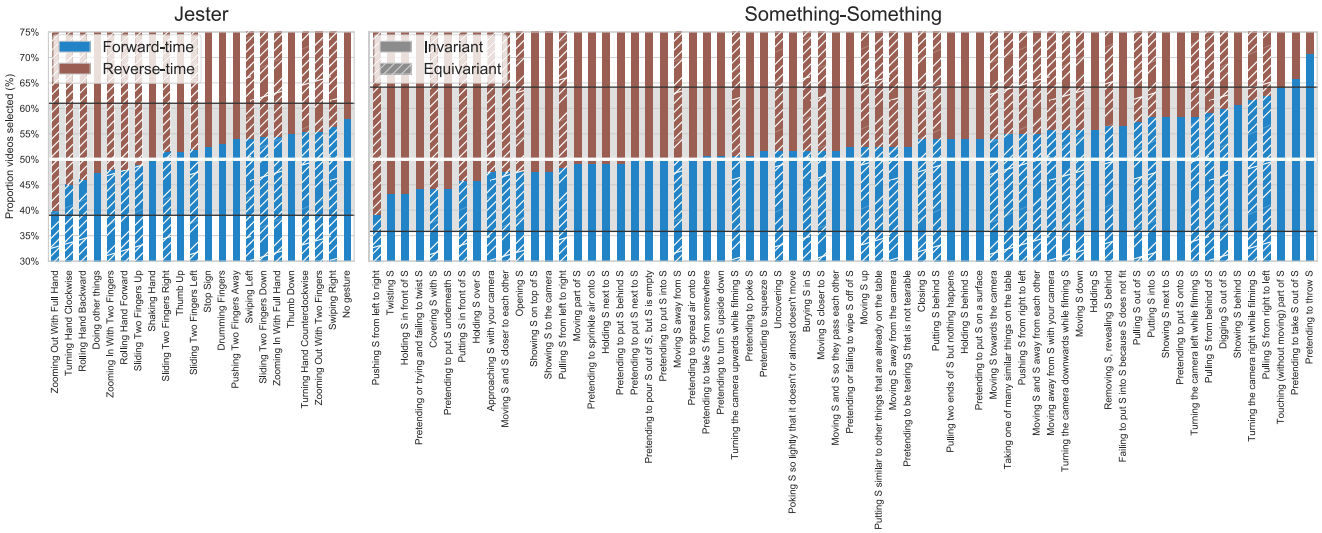

Figure 4: Proportion of forward-time examples selected over reverse-time examples across the 66 reversible classes in Something-Something and 22 in Jester. The interval between the dark horizontal lines depicts $50\% \pm 3\sigma$, within which we consider classes reversible.

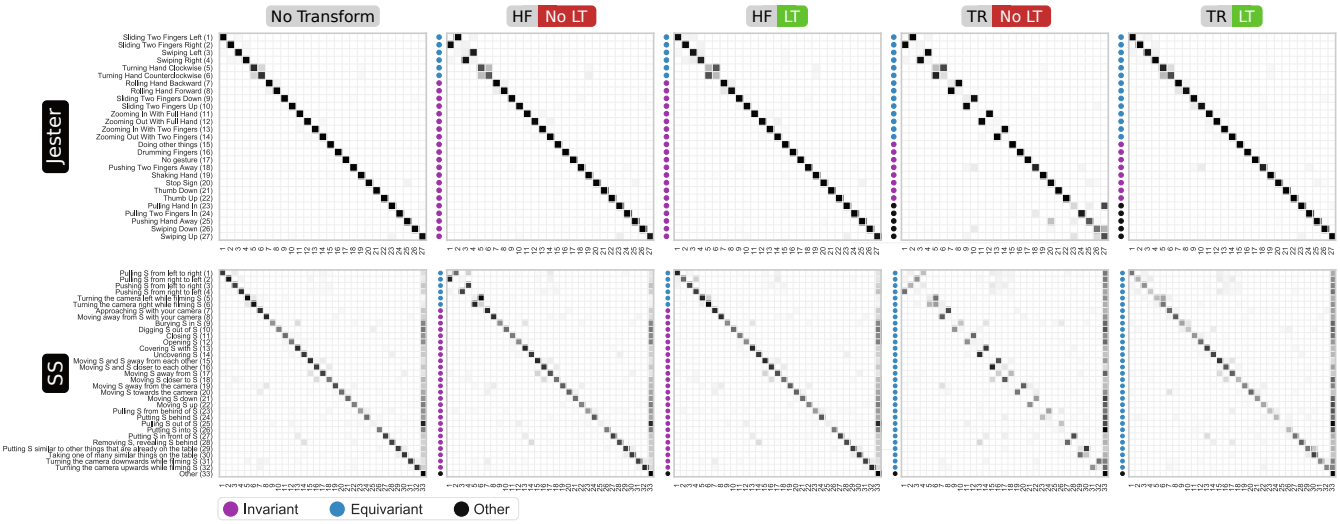

Figure 5: Models trained on forward-time unflipped videos are tested on videos transformed using horizontal flipping or time-reversal, each without or with the correct label transform. We only plot the confusion in Something-Something for the 32 time-reversal equivariant classes for clarity. [Best viewed on screen].

Most class transforms are correctly estimated in Jester for both horizontal-flipping and time-reversal. For Something-Something, we attribute the larger number of FP due to the models's lower performance (49/78% top-1/5 accuracy) and overlapping classes in the dataset. Frequently, the established class transforms were reasonable. For example '*Moving S away from S*' $\leftrightarrow$ '*Putting S next to S*' is a logical mapping, compared to an equally logical ground truth '*Moving S away from S*' $\leftrightarrow$ '*Moving S closer to S*'.

We investigated the use of NLP for semantically renaming y into its time-reversed class y by their antonyms, however we found existing lexical databases lacking. WordNet [19] does contain antonym relations, but these are quite sparse and are missing for common words like '*put*', '*take*', and '*remove*'. Additionally, the antonym relations that are present are general and don't always embody the time-reversed class *e.g.* '*move*' has the antonym '*stay*'.

In the following sections (Sections 5.2 and 5.3), we report results using the manual ground-truth rather than the discovered ones, avoiding propagating errors into the data augmentation and zero-shot evaluation. Finally, in Section 5.4, we test data augmentation and zero-shot learning using the automatically discovered class transforms.

5.2. Zero-shot learning

The novel-generating classes are ideally suited for zero-shot learning, extending the model's recognition abilities to previously-unseen classes. However, without a test set

Dataset	# classes	# examples	Dataset	# classes	# examples
Jester-HF	3	1387	SS-HF	3	523
Jester-TR	7	3450	SS-TR	16	2622

Table 3: Dataset subset zero-shot class and example counts. We still train for all classes in the full dataset, these counts are only for zero-shot classes.

		Zero-shot		NG many-shot		All classes	
	Supervision	Top-1	Top- 5	Top-1	Top-5	Top-1	Top- 5
Jester-HF	Chance	03.14	14.78	03.17	15.03	04.13	18.69
	HF	67.92	98.85	91.64	99.50	93.16	99.57
	Full	90.34	99.64	90.01	99.65	94.89	99.66
Jester-TR	Chance	03.35	15.64	03.41	15.99	04.14	18.69
	TR	78.90	98.70	94.01	99.66	91.99	99.40
	TR + HF	81.57	99.01	93.04	99.52	92.41	99.46
	Full	93.07	99.71	92.61	99.63	94.89	99.66
SS-HF	Chance	00.76	03.73	00.80	03.99	00.86	04.21
	HF	71.70	89.29	72.50	90.71	49.38	78.41
	Full	77.25	91.20	71.79	89.92	49.45	78.02
SS-TR	Chance	00.85	04.20	01.10	05.42	00.86	04.21
	TR	30.93	58.73	61.02	81.42	46.01	75.59
	TR + HF	39.89	64.80	60.45	80.84	46.56	76.24
	Full	62.01	81.88	62.41	83.10	49.45	78.02

Table 4: Zero-shot learning results compared to the upper-bound full-supervision. NG stands for novel-generating.

that includes examples of zero-shot classes, the model cannot be evaluated. We instead construct four train/test subsets to evaluate our approach. We turn pairs of equivariant classes into pairs of novel-generating many-shot and zero-shot classes. For each equivariant class pair, we retain the class with the highest training support as the novel-generating many-shot class and remove all examples of its counterpart, which then becomes a zero-shot class. The number of zero-shot classes and corresponding instances synthesised within those classes are listed in Table 3.

For each of the four sub-datasets (Jester-HF, Jester-TR, SS-HF, SS-TR), we compare chance (no supervision) as a lower bound to full supervision on all classes as an upper bound. We present our results in Table 4. Note that the number of zero-shot and many-shot classes differs per horizontal block. The results show that training for these zero-shot classes does not affect the performance of the many-shot classes compared to their full supervision performance. Over the four subsets, we report an overall drop in top-1 accuracy compared to full supervision of 1.7%, 2.9%, 0.1%, 3.4%, when dropping all training examples of 11%, 26%, 2% and 9% classes, respectively.

Figure 6 shows the confusion matrices for the pairs of many-shot and zero-shot classes in each subset. For Jester-HF and Jester-TR, we see good performance, but with the same confusion between classes '*turning hand clockwise*' and '*turning hand counterclockwise*' as in the base model. However, all other classes can be learnt in the zero-shot setting using horizontal flipping. For SS-HF, zero-shot classes

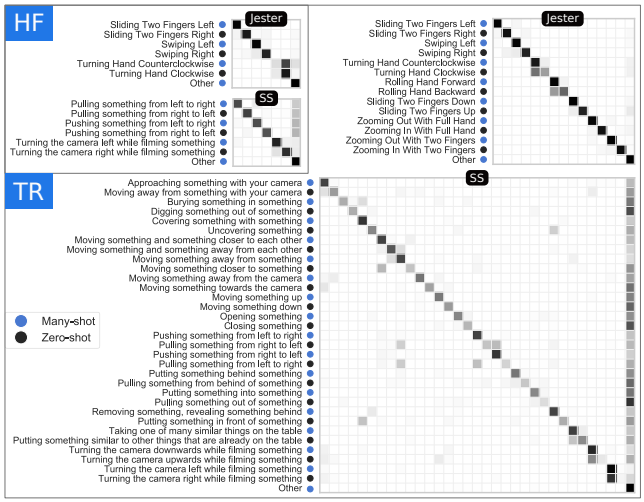

Figure 6: Confusion matrices of the many-shot and zero-shot classes showing minimal confusion between zero-shot classes and their many-shot counterparts. The final column of each confusion matrix shows confusion amongst all other classes not listed.

are distinguishable from their many-shot counterparts. In SS-TR, the camera movement zero-shot classes: '*turning the camera upwards/right*' have been confused with their many-shot class counterparts: '*downwards/left*'. This suggests that the model may be using motion blur in individual frames to classify the action as the model has never seen upwards/right motion blur effects. Overall these confusion matrices show that in the majority of cases, the use of LATs has resulted in impressive performance on zero-shot classes.

In Fig. 7, we show qualitative results on six examples from Something-Something. **Top Row:** A zero-shot model trained only on left-to-right examples can correctly classify zero-shot right-to-left actions. The final example shows a case where, although both models incorrectly predict the ground-truth class, their predictions are both reasonable. The zero-shot model has a greater difference between the top-2 scores indicating increased discriminative ability in the model. **Bottom Row:** The time-reversal zero-shot model has been able to learn state inversions like '*close*' (first) and '*uncover*' (second) from time-reversed examples of '*open*' and '*cover*'.

5.3. Data augmentation

We train a model using data augmentation as described in Section 3.1. For each video, the transform is applied with a probability of 0.5 along with the corresponding label transform. This approach results in balancing class support within each equivariant class pair. In TR+HF, we stack the randomly applied transforms to produce a mixture of videos with time-reversal, horizontal-flipping, or their composition. The results are presented in Table 6. Additionally, we include the results of augmenting with horizontal-

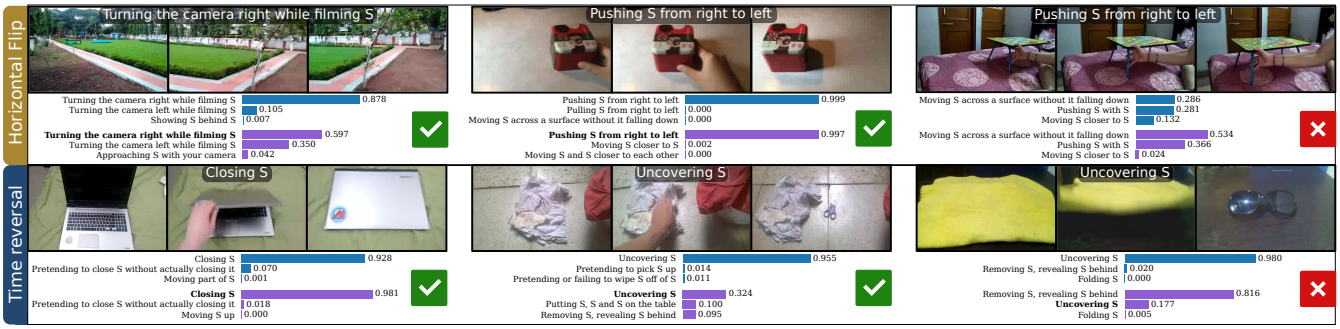

Figure 7: Sample from SS-HF (top), SS-TR (bottom) comparing the results using full supervision vs. zero-shot learning. Fully supervised model scores are blue, zero-shot model results are purple and zero-shot classes are **bold**. [Best viewed on screen].

		Zero-shot		NG many-shot		All classes	
Model	$\lvert Y_{\text{zs}}\rvert$	Top-1	Top- 5	Top-1	Top-5	Top-1	Top-5
TR	18	32.69	57.80	58.70	80.44	45.45 ▼-0.56	74.77 ▼-0.82
TR + HF	19	38.30	61.43	58.46	80.68	45.84 ▼-0.71	74.92 ▼-1.32

	All classes	
Model	Top-1	Top- 5
TR	48.52 ▼-0.48	77.77 ▼-0.14
TR + HF	49.02 ▼-1.25	78.08 ▼-0.92

Table 5: Something-Something zero-shot (*left*) and data augmentation (*right*) results using extracted class transforms time-reversal (*TR*) or horizontal-flipping (*HF*). $\lvert Y_{\text{zs}}\rvert$ indicates the number of zero-shot classes.

			All		Invariant		Equivariant	
	Augmentation	LT	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Jester	None	-	94.89	99.66	95.99	99.67	90.18	99.64
	HF (invariant only)	-	95.00	99.65	96.11	99.67	90.21	99.57
	HF	✗	94.55	99.65	96.21	99.67	86.89	99.57
	HF	✓	95.01	99.67	96.25	99.67	89.71	99.68
	None	-	94.89	99.66	97.20	99.71	92.84	99.67
	TR	✓	94.95	99.65	97.16	99.65	93.01	99.66
	TR + HF	✓	94.68	99.61	97.06	99.73	92.55	99.56
Something	None	-	49.45	78.02	48.31	77.45	74.42	90.49
	HF	✗	49.38	78.98	49.75	78.53	41.46	88.83
	HF	✓	50.26	78.94	49.20	78.36	73.50	91.78
	None	-	49.45	78.02	36.33	70.48	62.23	82.56
	TR	✓	49.00	77.91	35.12	69.10	60.52	82.97
	TR + HF	✓	50.27	79.00	36.95	69.85	61.23	83.52

Table 6: LAT data augmentation validation set results. 'LT' stands for label transform, where a hyphen indicates that a LT wouldn't make a difference. 'Invariant only' refers to applying the data augmentation to the invariant classes solely.

flipping but without label transformation, as this is a default, yet incorrect, data augmentation technique implemented in TRN and similar video recognition networks. This shows a clear drop (highlighted in Table 6) for equivariant classes.

On Jester, we find the best two configurations to be horizontal-flipping with label transformation and horizontal-flipping of invariant classes only. Horizontal-flipping with label-transformation improves performance on invariant classes by reducing confusion with equivariant classes. Time-reversal with label transformation slightly improves performance on equivariant classes.

On Something-Something, We find the combination of time-reversal and horizontal flipping improves top-1/5 accuracy by 0.8/1.0%, performing comparably to horizontal flipping with label transformation alone. Notably, without label transformation, horizontal flipping results in a model that underperforms the one trained without augmentation, but with label transformation, the model outperforms the unaugmented model by 0.8%. Note that we used all training examples in addition to transformed ones in this experiment. Data augmentation for few-shot learning (i.e. by using a subset of the training videos) is left for future work.

5.4. Using discovered class transforms

Up until this point, we have used manually defined class transforms to report results. This allowed evaluating LATs separately from the discovery of their class transforms. We report results on our whole pipeline, on Something-Something, for both TR and HF + TR from discovered class transforms, in Table 5. The performance is comparable (with a small drop 0.14-1.32% shown in red) to zero-shot learning in Table 4 and data augmentation in Table 6.

6. Conclusion

In this paper, we introduced the notion of label-altering video transforms, label transforms. We show example synthesis can be used for zero-shot learning and data augmentation with evaluations on two datasets: Something-Something and Jester. Future directions involve investigating other label-altering video transforms like video trimming or looping and exploring additional applications of these transforms, *e.g.* in few shot learning. We aim to also investigate learning optimal video transforms to achieve a particular class transform.

References

[1] 20BN. The 20BN-Jester Dataset. https://20bn.com/datasets/jester. Online; accessed 2019-02-17. 4

[2] Unaiza Ahsan, Rishi Madhok, and Irfan A. Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019. 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[5] Debidatta Dwibedi, Pierre Sermanet, and Jonathan Tompson. Temporal reasoning in videos using convolutional gated recurrent units. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. *arXiv e-prints*, page arXiv:1812.03982, Dec 2018. 2

[7] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[8] Basura Fernando, Efstratios Gavves, Jose M. Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[9] Amir Ghodrati, Efstratios Gavves, and Cees Snoek. Video Time: Properties, Encoders and Evaluation. In *British Machine Vision Conference (BMVC)*, page 160, 2018. 1, 2

[10] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 3

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter 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 *The IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 4

[12] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[13] Farnoosh Heidarivincheh, Majid Mirmehdi, and Dima Damen. Action completion: A temporal model for moment detection. In *British Machine Vision Conference (BMVC)*, 2018. 2

[14] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. PMLR, 2015. 5

[16] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-Supervised Spatiotemporal Feature Learning via Video Rotation Prediction. *arXiv e-prints*, page arXiv:1811.11387, Nov 2018. 2

[17] Will Kay, Joao 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. *arXiv e-prints*, page arXiv:1705.06950, May 2017. 2

[18] Hilde Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2011. 2

[19] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995. 6

[20] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In *The European Conference on Computer Vision (ECCV)*, 2016. 2

[21] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019. 2

[22] Suraj Nair, Mohammad Babaeizadeh, Chelsea Finn, Sergey Levine, and Vikash Kumar. Time Reversal as Self-Supervision. *arXiv e-prints*, page arXiv:1810.01128, Oct 2018. 2

[23] Lyndsey C. Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T. Freeman. Seeing the arrow of time. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1, 2

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv e-prints*, page arXiv:1212.0402, Dec 2012. 2

[25] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[26] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions ˜ transformations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[27] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *The IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), 2018. 1, 2

[28] Donglai Wei, Joseph J. Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[29] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

[30] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2, 5