Adversarial Augmentation Training Makes Action Recognition Models More Robust to Realistic Video Distribution Shifts

Kiyoon Kim Shreyank N Gowda Panagiotis Eustratiadis
Antreas Antoniou Robert B Fisher

University of Edinburgh

Abstract

Despite recent advances in video action recognition achieving strong performance on existing benchmarks, these models often lack robustness when faced with natural distribution shifts between training and test data. We propose two novel evaluation methods to assess model resilience to such distribution disparity. One method uses two different datasets collected from different sources and uses one for training and validation, and the other for testing. More precisely, we created dataset splits of HMDB-51 or UCF-101 for training, and Kinetics-400 for testing, using the subset of the classes that are overlapping in both train and test datasets. The other proposed method extracts the feature mean of each class from the target evaluation dataset's training data (*i.e.* class prototype), and estimates test video prediction as a cosine similarity score between each sample to the class prototypes of each target class. This procedure does not alter model weights using the target dataset and it does not require aligning overlapping classes of two different datasets, thus it is a very efficient method to test the model robustness to distribution shifts, without prior knowledge of the target distribution. We address the robustness problem by adversarial augmentation training – generating augmented views of videos that are "hard" for the classification model by applying gradient ascent on the augmentation parameters – as well as "curriculum" scheduling the strength of the video augmentations. We experimentally demonstrate the superior performance of the proposed adversarial augmentation approach over baselines across three state-of-the-art action recognition models - TSM, Video Swin Transformer, and Uniformer. Our curated datasets and source code are publicly available[1]. The presented work provides critical insight into model robustness to distribution shifts and presents effective techniques to enhance video action recognition performance in a real-world deployment.

Keywords: Action recognition, Distribution shifts, Adversarial training, Data augmentation.

Contents

[1] https://github.com/kiyoon/video-adversarial-augmentation

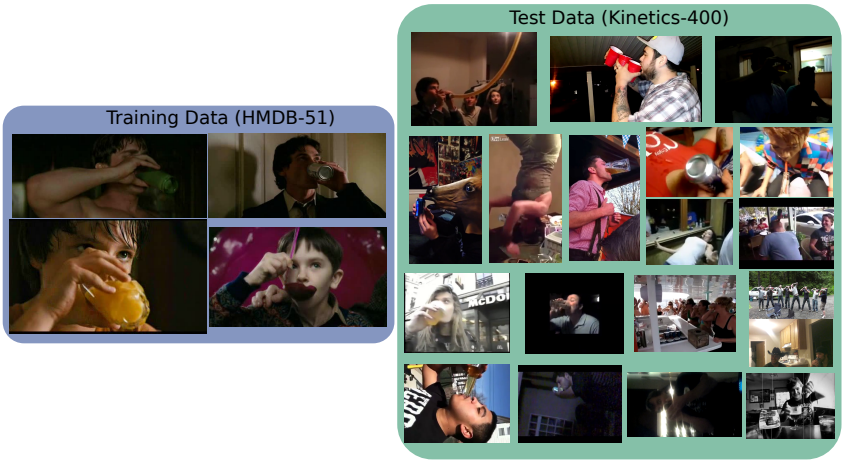


Figure 1: A common scenario with biased training data and testing with real-life data, with an example class of "drink". The training data is from the HMDB-51 dataset whose video samples are usually taken from movies, and the test data is from the Kinetics-400 dataset which is from YouTube videos. There are many reasons why the test data looks so different: poor camera quality, wrong orientation, extreme camera shake, inconsistent frame rate, frame rate conversion artifacts (interlacing), poor lighting, lack of professional post-processing (*e.g.* color grading), different ways of performing the action, poor framing, inconsistent aspect ratios, editing artifacts, various actions happening at the same time. Thus, it is common for the performance to drop significantly when the trained model is applied in more general applications.

1 Introduction

Video action recognition is a vital computer vision task with applications in surveillance, robotics, and more. Video data exhibits greater diversity than image data, and therefore action recognition architectures are not as robust to distribution shifts [38, 43, 58]. In addition to image-level effects like viewpoint and appearance changes, video introduces effects such as camera motion, focus shifts, and background object movements. Moreover, an action class incorporates substantial intra-class variation as illustrated in Fig. 1. For example, the class "playing basketball" may involve dribbling, running, or shooting in different contexts. Furthermore, depending on the data source, there are biased video processing artifacts. For example, videos collected from YouTube have standardized YouTube processing (VP9 compression), making the dark areas have extremely low quality. Often, the videos go through a frame rate conversion algorithm, which can create frame interlacing artifacts. As a result, the slight distribution shift in video data can dramatically reduce the action recognition performance.

Data augmentation is one potential solution to account for this fragility. It is a popular method to create synthetic variations of the existing training data that will enable classification models to generalize better to previously unseen test data. However, it is not yet clear what kind of augmentation is necessary to generalize to test data with different distribution shifts. There has been much work on automatically selecting augmentation policies given training and validation data [8, 22, 24, 34, 35]. However, such approaches optimize augmentation of the training data, and it is not clear how well the resulting generalization applies to test data with much distribution shift.

To address the video domain shift problem, we propose an adversarial augmentation scheme that generates "hard" video examples for the action recognition networks. The pipeline is simple to implement, and results in a meaningful improvement in performance on the proposed datasets with realistic

distribution shifts compared to no augmentation and random augmentation baselines. The benefits are demonstrated using three popular action recognition architectures. The training and validation datasets are subsets of HMDB-51 or UCF-101, and the test data are a subset from equivalent Kinetics-400 classes, to realistically evaluate distribution shifts over the same action classes.

This approach and evaluation requires multiple datasets with common aligned action classes. The paper also presents another simple method to evaluate robustness using a target dataset with different classes using cosine similarity of features as logits. The method requires no training (*i.e.* fine-tuning) on the target dataset, and thus it correctly measures the transferability of the trained classifier on the target dataset.

To summarize, our contributions are:

1. Experiments reveal substantial performance degradation on our cross-dataset benchmarks, quantitatively demonstrating the challenge posed by real-world distribution shift.
2. We propose two novel evaluation protocols to assess model resilience to distribution disparity using naturally sourced datasets, as opposed to solely artificially corrupted data:
   - 2*a*) We construct new cross-dataset benchmarks by identifying overlapping classes between HMDB-51, UCF-101, and Kinetics-400. Models are trained on either HMDB or UCF, and evaluated on Kinetics data.
   - 2*b*) We further introduce a similarity-based evaluation approach that estimates predictions using cosine similarity between embedded training and test features, without requiring class alignment.
3. Through extensive experiments across multiple state-of-the-art architectures, we empirically demonstrate that the proposed adversarial augmentation and curriculum adversarial training frameworks enhance robustness to realistic distribution shifts between the training and test datasets.
4. We publicly release the constructed subsets of HMDB-51, UCF-101, and Kinetics-400 to enable further research on this important problem.

2 Related Work

Action recognition. Action recognition is the task of categorizing video sequences into predefined action classes. Architectures based on 3D convolutional neural networks were previously dominant for spatiotemporal feature learning. These include approaches such as inflating 2D models [7], incorporating relational reasoning with non-local operations [55], and dual-stream designs [14, 47]. More recently, transformer networks have become prominent, demonstrating strong performance [12, 33, 39] despite their exponential computational complexity [29]. For efficient video recognition, 2D backbone models remain popular, using techniques like temporal feature aggregation [52], relational modeling [63], temporal shift modules [36], frame selection [20, 31, 57], channel-wise convolutions (*i.e.* height-width, height-time, width-time) [56] or analyzing short-term and long-term temporal difference [53].

Data augmentation. [45] summarizes image data augmentation techniques for deep learning. AutoAugment [8] is an augmentation policy search algorithm that finds the best augmentation on a target dataset, based on the highest validation accuracy. Due to AutoAugment's expensive policy search, Population-Based Augmentation [24], FastAutoAugment [35], and FasterAutoAugment [22] proposed more efficient searching algorithms, by learning a schedule policy over a fixed-policy, using density matching, and using differential augmentation with a generative adversarial network (GAN) [18] architecture that involves a policy generator and a discriminator, respectively. Differentiable Automatic Data Augmentation [34] proposed a data augmentation policy searching algorithm (using the Relaxed Bernoulli distribution [26]) which is differentiable, similar to FasterAutoaugment, and further introduced an unbiased gradient estimator that enables joint optimization of the augmentation policies and network parameters, instead of using a GAN. RandAugment [9] showed that simple random augmentations with randomly sampled transformations achieve similar performance more efficiently.

However, the policies in most works are optimized on the training set, and we focus on the scenario where test data can have severe distribution disparities which are unknown during the training time.

Adversarial training. Adversarial training (AT) is framed as a min-max problem whereby the trained model uses observed training samples to minimize its prediction error, while an adversary attempts to generate training samples that maximize it. It is well-established that AT is the most effective way to achieve adversarial robustness [2]. It has further been shown to yield other types of robustness, *e.g.* against natural corruptions [10], domain shift [30, 44, 64], and others. Note that even though the classical definition of AT uses adversarial input noise [6, 19], more adversarial image augmentations have been studied, *e.g.* rotations [54, 62], contrast, jitter, etc. [3]. AT should be approached with care, as

generating adversarial training examples that are too challenging for the trained model may actually harm downstream performance [5].

In this paper, we employ two measures to control the trade-off between augmentation strength and performance: (i) We create maximally informative adversarial examples (confusing to the model, but near the classification boundaries) via maximum-entropy regularization, as per the work of [11, 27, 59]. (ii) We train with curriculum AT as per the work of [5, 60], which means training with harder adversarial examples over time.

Domain adaptation. Domain adaptation is a transfer learning task where the source and target datasets have a significant distribution shift while sharing the same task. [13] explains types of domain adaptation tasks and approaches.

There are discrepancy-based techniques that learn transferable features from a source domain to a target domain [40, 61], reconstruction-based methods that utilize autoencoders, which aim to extract useful features for the target domain [16, 17], and adversarial domain adaptation approaches involving a source / target discriminator that distinguishes where the data come from and a feature extractor that aims to confuse the discriminator by trying to produce generic features regardless of the domain [1, 4, 15, 25, 42, 46, 50, 51]. More recently, analyzing frequency components of deep feature maps using attention to filter domain-general components [37] is proposed.

Domain adaptation for video action recognition was first proposed by using a feature alignment approach on online test videos [38]. This work was evaluated using computationally simulated corrupted videos, while we propose to use real examples that involve more diverse types of discrepancy between the domains.

It is important to note that most domain adaptation techniques require examples from the target dataset to be present, while our work focuses on evaluating using a completely unknown dataset.

Corruption robustness analysis. [23] provided benchmarks for measuring a neural network's robustness to corruptions and perturbations, by evaluating with 15 algorithmically-generated corruptions (*e.g.* noise, blur, pixelate, compression artifacts). [58] extended this to video classification tasks and video corruptions (*e.g.* video compression artifacts, frame rate conversion, bit error, packet loss). [43] reported a large-scale robustness analysis of deep action recognition models again using pre-defined perturbations.

These approaches were evaluated using simulated data, while we propose to use real data for testing. Evaluating robustness with augmented data prohibits the same augmentations to be used for training. This paper focuses on a more realistic scenario where a known set of data augmentation strategies is used for training and evaluation is done with unprocessed real data.

3 Problem and Methodology

Action recognition predicts an action category label given a video sequence. This paper explores how well different variations of action recognition models, training, and loss functions generalize by evaluating on a different data domain.

The main difference with transfer learning is that our approach does not tune model parameters using the target evaluation dataset, whereas transfer learning usually involves fine-tuning the model with the target dataset's training set.

To improve generalizability, the training data is augmented. Hard-to-classify adversarial examples are generated by applying gradient ascent on the augmentation parameters which are fully differentiable. We then train the classifier using the AT (Adversarial Training) loss, calculated using both clean and adversarial examples.

3.1 Adversarial Augmentation Training

Adversarial augmentation training uses a two stage training loop. See Fig. 2.

Stage 1: Generate adversarial examples. "Hard" adversarial examples are found by tuning the augmentation parameters using gradient ascent.

Let $g_\theta(\mathbf{x})$ be an augmentation model with fully differentiable parameters θ, and $f_\phi(\mathbf{x})$ be a video classification model with parameters ϕ, that outputs class predictions given an input video $\mathbf{x}$. The goal of stage 1 is to find the augmentation parameters θ' that maximize categorical cross-entropy loss. Note that by *maximizing* the loss, we aim to find the augmentation strategy that is challenging for the classifier, and in Fig. 2, this is described as *minimizing* the negative cross-entropy loss.

At each training step, the augmentation parameters θ are randomly initialized. Gradient ascent is then done only on θ, freezing the classification parameters ϕ to learn adversarial augmentations, with

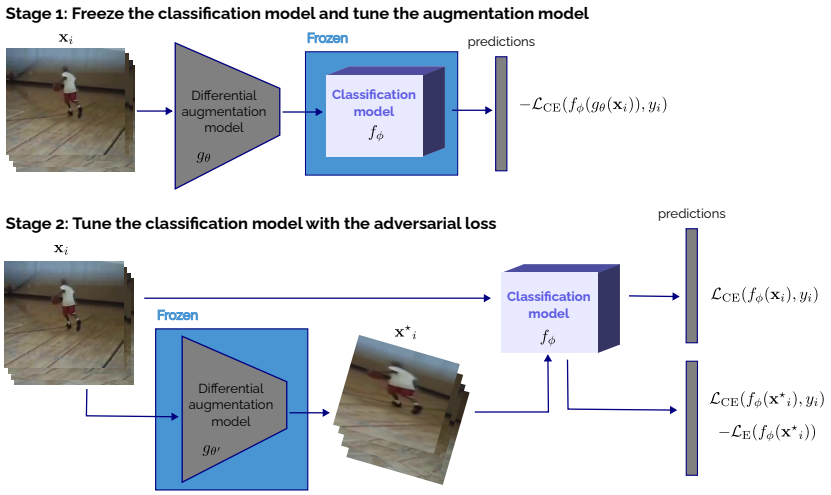


Figure 2: The proposed adversarial augmentation training has two separate stages. Firstly, the classification model is frozen while the differential augmentation model is trained using the negative cross-entropy loss. This is equivalent to performing gradient ascent to maximize normal cross-entropy loss. As a result, the augmentation model will generate hard augmentations for the classification model. The second stage trains the classification model using both clean and adversarial examples. The maximum entropy regularization loss is integrated by subtracting the entropy of the adversarial examples, encouraging the predictions to be evenly balanced.

the cross-entropy loss $\mathcal{L}_{\text{CE}}(f_\phi(g_\theta(\mathbf{x}_i)), y_i)$. This optimizes augmentation parameters θ' for generating adversarial examples.

Stage 2: Optimize the classification model. Next, the classification parameters ϕ are optimized while freezing the augmentation parameters θ. For simplicity, $\mathbf{x}_i^\star = g_{\theta'}(\mathbf{x}_i)$ denotes the generated adversarial example of $\mathbf{x}_i$. The vanilla AT loss is defined as:

$$\mathcal{L}_{\text{AT}}(f_\phi(\mathbf{x}_i), f_\phi(\mathbf{x}_i^\star), y_i) = \alpha\mathcal{L}_{\text{CE}}(f_\phi(\mathbf{x}_i), y_i) + (1-\alpha)\mathcal{L}_{\text{CE}}(f_\phi(\mathbf{x}^\star{}_i), y_i) \tag{1}$$

which is a weighted average of the cross-entropy loss using the clean sample and the augmented sample.

Max-Entropy Regularization. The cross-entropy loss encourages predictions to be over-confident by pushing the examples further from the classification boundaries. However, adversarial examples are supposed to be confusing. We regularize the cross-entropy-based adversarial loss in Eq. (1) by maximizing the entropy, encouraging the overall predictions to be evenly balanced for adversarial examples.

$$\begin{aligned}\mathcal{L}_{\text{AT-ME}}(f_\phi(\mathbf{x}_i), f_\phi(\mathbf{x}_i^\star), y_i) = \alpha\mathcal{L}_{\text{CE}}(f_\phi(\mathbf{x}_i), y_i) \\ + (1-\alpha)\mathcal{L}_{\text{CE}}(f_\phi(\mathbf{x}^\star{}_i), y_i) - \gamma\mathcal{L}_{\text{E}}(f_\phi(\mathbf{x}^\star{}_i))\end{aligned} \tag{2}$$

where entropy loss $\mathcal{L}_{\text{E}}$ is defined as

$$\mathcal{L}_{\text{E}}(f_\phi(\mathbf{x}_i^\star)) = -\frac{1}{C}\sum_{c=1}^{C} f_\phi^{(c)}(\mathbf{x}_i^\star)\log(f_\phi^{(c)}(\mathbf{x}_i^\star)) \tag{3}$$

C is the number of classes, and $f_\phi^{(c)}(\mathbf{x}_i^\star)$ is the prediction score of class c for the adversarial example $\mathbf{x}_i^\star$.

Curriculum Adversarial Training. Applying curriculum training by starting from training with "easy" samples and gradually generating "harder" samples makes the model more robust [5]. Here, the classification models are trained initially from clean data without augmentation, and gradually harder adversarial examples are added by scheduling the learning rate of the augmentation model.

3.2 Cross-Dataset Evaluation

We train on one dataset and test on another dataset, where there are distribution shifts between the train and the test data. We propose two different evaluation approaches.

Matched Class Evaluation (Expt. 1). The first approach creates two datasets that share the same classes, but have a significant disparity in the train and test data distribution. Classes that are common to the two initial datasets are identified following the procedure described in TruZe [21]. We describe

the procedure and curated datasets in Supplementary Material. This is the most realistic method to evaluate on a distribution shift, but it requires some manual procedures as well as finding datasets that share largely similar classes.

Cosine Similarity Evaluation (Expt. 2). The second method applies the feature extractor trained on the source dataset to the videos in the target dataset, which are split into a training subset and a test subset. This is a simpler method that does not require manual class matching. A more descriptive procedure can be found in the Supplementary Material.

Formally, the class prototype $\mathbf{c}_k \in \mathbb{R}^M$ for each class k is the M-dimensional mean vector of the embedded features belonging to that class in the training subset. Let S_k be the set of videos in the training subset of the target dataset from class k. $\mathbf{c}_k$ is computed by the following, where $y_i = k$ for all $(\mathbf{x}_i, y_i) \in S_k$:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} h_\omega(\mathbf{x}_i) \tag{4}$$

$h_\omega()$ is the embedded feature extraction function computed by training on the source dataset.

For a given test video from the test subset of the target dataset $\mathbf{x}_i \in S_k^{test}$, the probability of a given class label k is estimated using the cosine similarity of the embedding to the target dataset's class prototype $\mathbf{c}_k$, followed by softmax. Given the cosine similarity function $d(\mathbf{x}, \mathbf{c}) = \frac{\mathbf{x} \cdot \mathbf{c}}{\|\mathbf{x}\| \|\mathbf{c}\|}$ we get:

$$P(y = k | \mathbf{x}_i \in S_k^{test}) = \frac{\exp(-d(h_\omega(\mathbf{x}_i), \mathbf{c}_k))}{\sum_{k'} \exp(-d(h_\omega(\mathbf{x}_i), \mathbf{c}_{k'}))} \tag{5}$$

where y denotes the class label. If the largest estimated $P(\cdot)$ is for the same class as the ground truth, then this is a successful classification. Accuracy is computed over all samples in the test subset of the target dataset.

The parameter ω is not tuned during this operation. That is, the target dataset does not contribute to fine-tuning the model. The motivation of this approach is to measure the transferability of the model from a source dataset to a target dataset without actually tuning the model parameters, showing the robustness of the model to sample distribution shift.

4 Experiments

HMDB/Kinetics and UCF/Kinetics Datasets (Expt. 1). HMDB-51 [32] is a popular human action recognition dataset that is composed of around 7,000 video clips divided into 51 categories, collected from mainly movies as well as YouTube. The UCF-101 [49] dataset consists of 13,320 videos in 101 action classes collected from YouTube. Kinetics-400 [28] is a large-scale action dataset in which each video clip is around 10 seconds long, and there are over 300,000 videos in 400 classes.

Training datasets are created from subsets of HMDB-51 or UCF-101 and the subset of the Kinetics-400 test set is used for testing. The subsets share the same classes between the training and test sets. The motivation for this approach to creating the datasets is to simulate a real-world environment where the test data comes from many unknown sources, with many variations in actions, capture conditions, aspect ratio, and so on. The Kinetics-400 dataset has many more samples in the fine-grained classes, so it was used for testing.

TruZe [21] is used to identify shared classes from the HMDB-51 and Kinetics-400, and UCF-101 and Kinetics-400 datasets, based on the visual and semantic similarity. More details on this procedure can be found in the Supplementary Material. The final datasets are named HMDB-28/Kinetics-28 and UCF-65/Kinetics-65, where the training sets are subsets of the HMDB and UCF data, respectively, and test subsets come from Kinetics. The 28 and 65 refer to the number of shared classes. The three official published splits of HMDB-51 and UCF-101 are used, but only the shared classes are selected. The results in Table 1 are the average performance over the three splits. The HMDB-28/Kinetics-28 dataset consists of 3445 HMDB and 43406 Kinetics videos, and the UCF-65/Kinetics-65 dataset consists of 8935 UCF and 78583 Kinetics videos.

HMDB $\leftrightarrow$ UCF Evaluation (Expt. 2). For the experiments using the cosine similarity function, the HMDB-28 trained models are tested on UCF-101, and the UCF-65 trained models are tested on HMDB-51, using the cosine similarity measure with class prototypes as predictions. The feature extractor trained using the training data is then used on the target dataset. It is used to create class prototype vectors (for each target class) from one part of the target dataset, and evaluation is based on the

Model	Train Dataset	Augmentation	Test Accuracy	
			Kinetics-65 Matched Expt 1a	HMDB-51 Cosine Expt 2a
TSM	UCF-65 [77.50]	None	39.13 ± 0.56	37.61 ± 1.88
		Random	40.90 ± 0.32	38.19 ± 1.39
		Adversarial	42.42 ± 0.63	38.99 ± 1.26
		Curriculum	**42.51 ± 0.62**	**39.14 ± 1.21**
Swin	UCF-65 [81.15]	None	37.08 ± 1.43	38.91 ± 1.63
		Random	40.80 ± 1.90	40.61 ± 1.40
		Adversarial	**42.27 ± 0.24**	41.48 ± 1.58
		Curriculum	41.20 ± 0.72	**41.58 ± 1.63**
Uniformer	UCF-65 [52.05]	None	18.78 ± 0.22	21.39 ± 1.32
		Random	22.42 ± 0.98	24.92 ± 1.66
		Adversarial	22.95 ± 0.68	**26.16 ± 2.10**
		Curriculum	**23.61 ± 0.27**	24.93 ± 1.60

Model	Train Dataset	Augmentation	Test Accuracy	
			Kinetics-28 Matched Expt 1b	UCF-101 Cosine Expt 2b
TSM	HMDB-28 [55.45]	None	29.75 ± 1.08	60.51 ± 0.79
		Random	30.13 ± 0.42	61.21 ± 0.57
		Adversarial	32.44 ± 0.48	62.60 ± 0.59
		Curriculum	**32.82 ± 1.41**	**63.18 ± 0.71**
Swin	HMDB-28 [54.67]	None	25.26 ± 0.80	59.88 ± 0.55
		Random	26.63 ± 0.97	60.99 ± 1.30
		Adversarial	27.31 ± 0.57	62.57 ± 0.72
		Curriculum	**27.60 ± 0.62**	**62.95 ± 0.82**
Uniformer	HMDB-28 [29.43]	None	14.53 ± 0.51	28.23 ± 0.94
		Random	14.33 ± 1.03	28.67 ± 1.36
		Adversarial	15.13 ± 0.79	29.88 ± 2.62
		Curriculum	**15.25 ± 0.75**	**30.83 ± 1.04**

Table 1: Results from three models (TSM ResNet50, Video Swin Transformer Tiny, and Uniformer-S), two training datasets (UCF-65 and HMDB-28), four augmentation strategies (no augmentation, random, adversarial, and curriculum), and two test datasets (Kinetics, and HMDB/UCF). In all cases, adversarial augmentation or curriculum adversarial augmentation training outperformed all baselines. The columns labeled Test Accuracy show the performance on the target test set. The values in brackets in the Train Dataset columns show the "no augmentation" accuracy on the test set of the same dataset to demonstrate the performance drop when evaluating to the Kinetics dataset.

classification accuracy of the other part of the dataset. This assesses the quality of the feature extractor on another dataset with distribution shifts. The results in Table 1 are the average performance over the nine splits, three runs from the source dataset and each run evaluates with three splits from the target dataset.

See Supplementary Material for the resulting matching datasets in Table 2, and a summary table of all experimental datasets (Expt. 1 and Expt. 2), in Table 4.

Augmentation Methods. Results are compared for four different training approaches: training without augmentation, with random augmentation, with adversarial augmentation, and with curriculum AT [5] as described in Section 3.1. Experiments used the popular efficient 2D TSM model [36] with a ResNet50 backbone, Video Swin Transformer [39] Tiny, and Uniformer-S [33] model. ImageNet pre-trained models were used instead of Kinetics pre-trained, so that the models never get to see the Kinetics data distribution.

Augmentation used translation to a maximum of 28 pixels, 10° rotation, shear transform of 0.1, and scale from 0.9 to 1.5. For curriculum training, no augmentation was used for 20 epochs, then AT with a zero learning rate of the augmentation model was used for 20 more epochs. Then, AT with a triangular learning rate scheduling from 0.1 to 1.0 on the augmentation model was used for the rest of the training, except for the Uniformer model. For Uniformer, the above was done for only 20 epochs, and then trained with random augmentation for 20 more epochs and fine-tuning with no augmentation for the rest to mitigate under-fitting issues.

See the Supplementary Material for implementation details.

5 Results

5.1 Shared Class Experiments 1a, 1b

When the adversarial training approaches presented in Section 3.1 are applied, target dataset performance improves. Table 1 summarizes the main cross-dataset evaluation results for the four augmentation strategies presented in Section 3.1. The different adversarial augmentation strategies gave improved accuracy for the target datasets (see Kinetics-28 and 65 results columns).

Also, unlike what is reported in [43], the convolutional architecture performed better with the distribution shift compared to transformer models in most of the cases except for the Swin Transformer trained on UCF-65 and tested on HMDB-51. We hypothesize that this is because the Kinetics test dataset shows natural and realistic distribution shifts. In addition, we did not use the Kinetics pre-trained models, and the transformer architectures require large-scale data to reach the maximum potential.

In all cases, the adversarial augmentation or curriculum methods outperform all baselines, given a fixed network architecture, for all training and test datasets. Although the "random augmentation" and "adversarial augmentation" allow an identical range of transforms, generating adversarial examples through gradient ascent produces "harder than random" augmentation which improves the overall performance. Furthermore, adding the simple curriculum mostly improved over the adversarial benchmark.

See the Supplementary Material for confusion matrices that show per-class performance drop with distribution shifts and improvements using the proposed adversarial augmentation.

5.2 Cosine Similarity Evaluation 2a, 2b

The cosine-similarity-based accuracy results on HMDB-51 and UCF-101 are shown in the Cosine Expt columns of Table 1. The results follow a very similar trend to the "realistic" Kinetics Expt 1. The advantage of using this accuracy measure as compared to testing on Kinetics with overlapping classes is that it requires no thorough analysis of the source and target datasets to find overlapping classes, making it simple to set up the cross-dataset experiments even using new datasets.

6 Conclusion

This paper addressed the problem of model generalization to realistic test distribution shifts. Two new datasets that are comprised of three existing datasets were created that shared the same subset of label classes. Although the same classes were used, the variety of videos in the original dataset sources meant that there was a huge distribution shift from the source to the target datasets. When using the target datasets, action recognition performance dropped significantly. This led to trying adversarial augmentation, with and without curriculum scheduling, as an approach to generating hard adversarially augmented videos. This approach gave a small but meaningful improvement in performance, even with the large distribution shift in the test data. The second cross-dataset evaluation approach, using the cosine similarity as logits, also showed a similar trend as the matching dataset experiments, providing a simpler alternative method without having to curate datasets with matching classes.

Supplementary Material

A TruZe Class Matching Procedure

For Expt. 1a and 1b, we follow TruZe [21] to identify overlapping classes in two different datasets. To choose the common classes, visual features are extracted on two existing datasets using the I3D [7] model pre-trained on Kinetics-400 [28] and semantic cues are extracted using the sen2vec [41] model pre-trained on Wikipedia. The visual and semantic similarity features are combined and then used in the TruZe [21] procedure (*i.e.* include exact matches, matches that can be either superset or subset, and matches that predict the same visual and semantic matches) to obtain a set of extremely similar classes from the two source action recognition datasets. The matched classes from the two datasets are verified manually and a subset is selected that has a substantial overlap in visual and semantic cues. One dataset is used for training and validation, and the other dataset is used for testing.

In TruZe, normally, classes from UCF or HMDB that have overlapping context with the corresponding Kinetics class are removed, so that using the Kinetics pre-trained models would not bias the zero-shot settings. However, in our robustness problem, the train and test datasets are created with the *opposite* goal, where only overlapping classes are selected. For instance, the class "climb" in HMDB-51 is treated as the same class as "rock climbing", "ice climbing", "climbing a rope", and "climbing tree" in Kinetics-400. Examples of the resulting splits for the matched class experiments (Expt 1a and 1b) can be found in Table 2.

B Cosine Similarity Evaluation Procedure

In Expt 2a and 2b, we use cosine similarity between class prototypes and features on the evaluating dataset to estimate predictions. We then report accuracy with the simulated predictions to compare the performance of augmentation strategies given distribution shifts.

We first train the classification model using the source dataset. Since the number and types of classes of the training (source) and evaluation (target) datasets are different, we detach the last classification layer of the source dataset classification model to use it as a feature extractor. We then calculate class prototypes of all the classes in the target dataset by simply averaging the features of each class in the training set, inspired by [48]. The prediction score of a sample from the target dataset's test set is estimated using the cosine similarity of their feature vector and the class prototypes, followed by softmax. In other words, the cosine similarity is used as logits, which are formed by producing high activations on classes that are closely aligned to the training set of the target dataset. Using these estimated prediction probabilities, we report accuracy. We will show that adversarial augmentation of the source dataset also produces improved classification of the target dataset, by producing a more robust feature extractor for use with this similarity measure. Note that the source dataset is never used for evaluation, and only be used for training the model, and the target dataset's training set does not contribute to fine-tuning the model. The role of all four splits in two different datasets can be found in Table 3.

C Datasets

Table 4 summarizes all datasets used for the experiments (Expt. 1 and Expt. 2).

D Implementation Details.

Videos were resized to 224×224 and sampled to 8 frames sparsely similar to [52]. The classifier models were trained for 200 epochs using an SGD optimizer with an initial learning rate of 0.0001, decaying the learning rate using cosine annealing scheduling. For adversarial training (AT), the augmentation model used a learning rate of 0.1. For adversarial plus curriculum training, the $\mathcal{L}_{\text{AT-ME}}$ loss was used with $\alpha = 0.5$ and $\gamma = 0.5$. Two NVIDIA RTX 3090 GPUs were used with batch size 16 to train the TSM models, an NVIDIA A100 GPU with batch size 16 and 32 for training the Uniformer and Swin Transformer models, respectively.

HMDB-51 Classes	Kinetics-400 Classes
brush hair	curling hair, dying hair, brushing hair
cartwheel	cartwheeling, somersaulting
catch, throw	shooting goal, juggling, catching or throwing frisbee, catching or throwing baseball, catching or throwing softball, throwing axe, throwing ball, throwing discus
clap	clapping, applauding
climb	rock climbing, ice climbing, climbing tree, climbing a rope
dive	diving cliff, bungee jumping
dribble	dribbling basketball
drink	drinking beer, drinking shots, drinking
eat	eating burger, eating cake, eating carrots, eating chips, eating doughnuts, eating hotdog, eating ice cream, eating spaghetti, eating watermelon
golf	golf driving, golf chipping, golf putting
...	...

UCF-101 Classes	Kinetics-400 Classes
Basketball	shooting basketball, dribbling basketball, playing basketball
BasketballDunk	dunking basketball
BodyWeightSquats	lunge, squat, dead lifting
BreastStroke	swimming breast stroke, swimming back stroke
CleanAndJerk	clean and jerk, snatch weight lifting
CliffDiving	diving cliff, springboard diving
Haircut	shaving head, braiding hair, getting a haircut
HorseRiding	riding or walking with horse, riding mule
PlayingPiano	playing piano, playing organ
Skiing	skiing (not slalom or crosscountry), skiing crosscountry, skiing slalom
...	...

Table 2: The HMDB-28/Kinetics-28 (above) and UCF-65/Kinetics-65 (below) datasets which are subsets from the original HMDB-51 UCF-101, Kinetics-400 datasets. Visually and semantically similar classes were combined which allows testing with real-life data from YouTube that has a significant distribution shift from the training data. The HMDB or UCF data are used for training and validation, and the Kinetics data are used for testing.

Dataset Type	# Classes	Split	Usage
Source	N	train test	Training Validation
Target	K	train test	Obtaining K class prototypes Testing with cosine similarity as logits

Table 3: Purpose of all dataset splits for the cosine similarity evaluation method (Expt. 2). Given the four splits of the data, the source dataset is used to train (with or without AT) and validate the model. The target dataset is used to evaluate the model without further fine-tuning.

Expt	Name	Original Dataset	Usage
1a	UCF-65 Kinetics-65	UCF-101 Kinetics-400	Training, validation Testing UCF-65 trained models
1b	HMDB-28 Kinetics-28	HMDB-51 Kinetics-400	Training, validation Testing HMDB-28 trained models
2a	UCF-65 HMDB-51	UCF-101 (original)	Training, validation Testing UCF-65 trained models w/ cosine similarity
2b	HMDB-28 UCF-101	HMDB-51 (original)	Training, validation Testing HMDB-28 trained models w/ cosine similarity

Table 4: List of datasets used for our experiments. The HMDB-51, UCF-101, and Kinetics-400 are the original datasets. The UCF-65, Kinetics-65, HMDB-28, and Kinetics-28 are the proposed subsets.

E Adversarial Augmentation Examples

Some examples of adversarial augmentations in comparison to random augmentations are depicted in Fig. 3. One might think that the most extreme and unrealistic augmentations will be challenging to the classifier. However, adversarial augmentation can sometimes render more realistic yet challenging examples.

F Performance Drop with Distribution Shifts

Two confusion matrices are created collating the performance when training on HMDB-28 and testing on either HMDB-28 or Kinetics-28 (test sets) similar to Expt 1a and 1b. The former is subtracted from

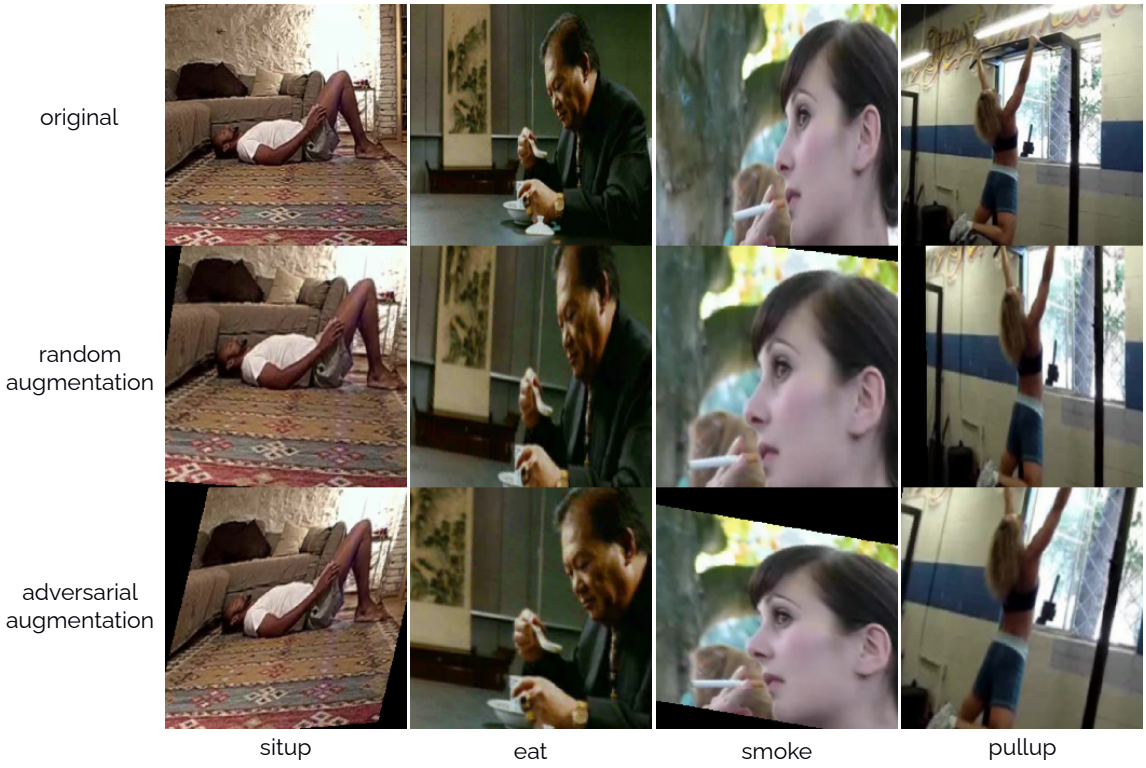


Figure 3: Examples of the vanilla and proposed augmentation examples in the HMDB-51 dataset while training the Video Swin Transformer model.

the latter to record how much difference there is in the performance, which is shown in Fig. 4(a). (The figure also shows the same results when using UCF-65 and Kinetics-65.) In this figure, no augmentation strategy is used. The figure shows that there is a dramatic performance drop when the models are tested on a different dataset (Kinetics) to training (HMDB or UCF). Some categories such as "situp", "shoot_bow", and "drink" are damaged more severely. As seen in Fig. 1, HMDB/UCF videos have objects and actors that are clearly visible and stable in the frame, while Kinetics videos tend to have lots of camera motion with different sizes of the objects. The overall accuracy drop from UCF-65 to Kinetics-65 is even more significant. These results show that the raw trained model is not robust to distribution shifts between datasets, which is not ideal for deployment in real applications.

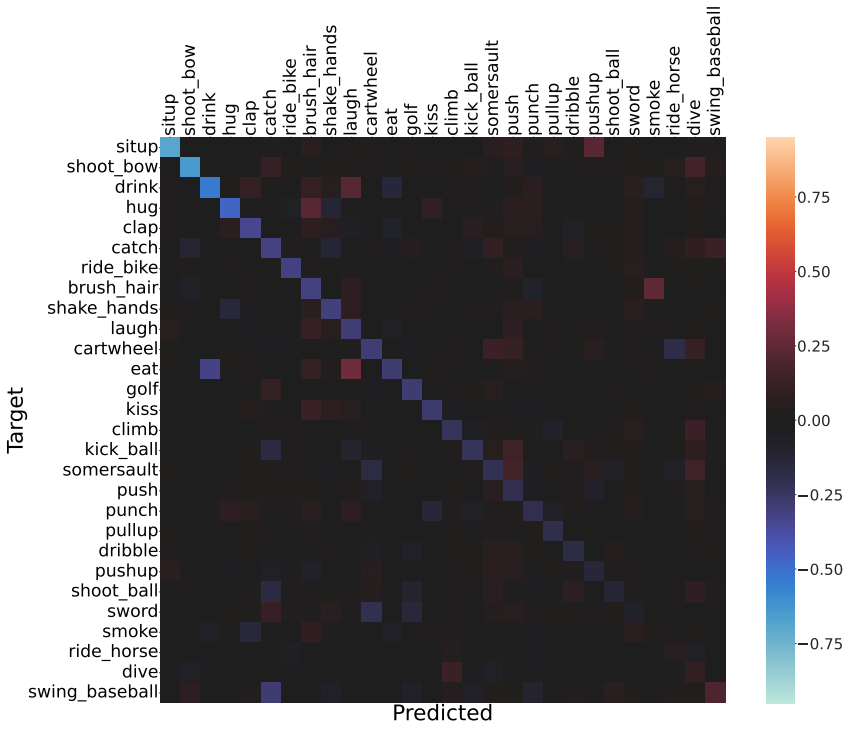
G Improvements on Classes with Larger Distribution Shifts

To demonstrate the improvement in transfer performance by the use of the curriculum adversarial strategy, we computed the difference of the confusion matrices for None and Curriculum augmentation in Expt. 1a and 1b. Fig. 5 shows the results, where the reddish boxes on the diagonal indicate improved performance. It is clear that the proposed adversarial augmentation makes the model more robust on the classes with the larger distribution shifts, as presented in Fig. 4. This supports our claim that the proposed adversarial augmentation makes the model more robust in classes with a huge distribution shift.

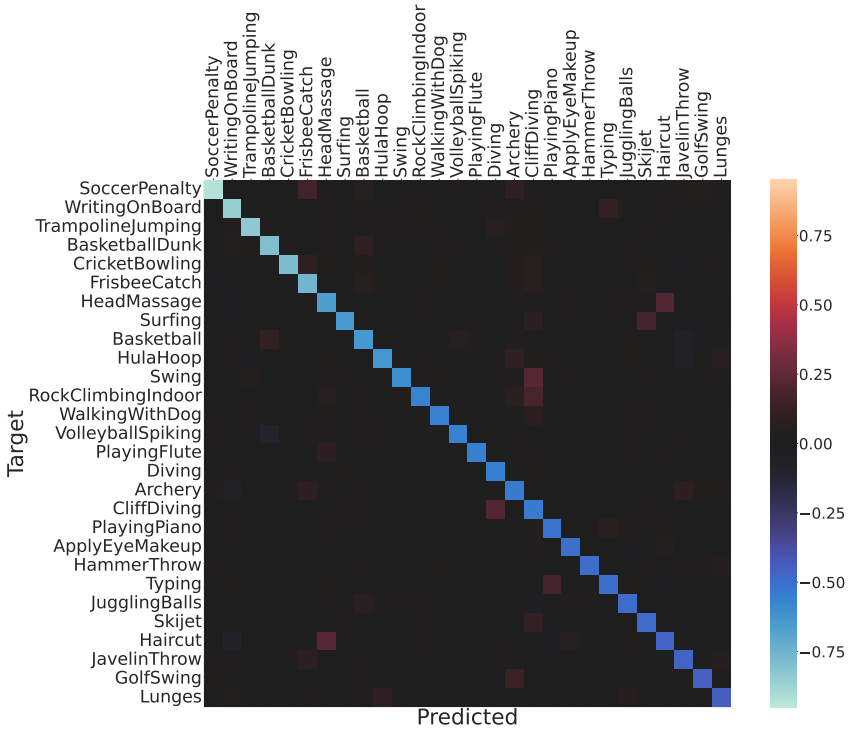
References

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[2] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.



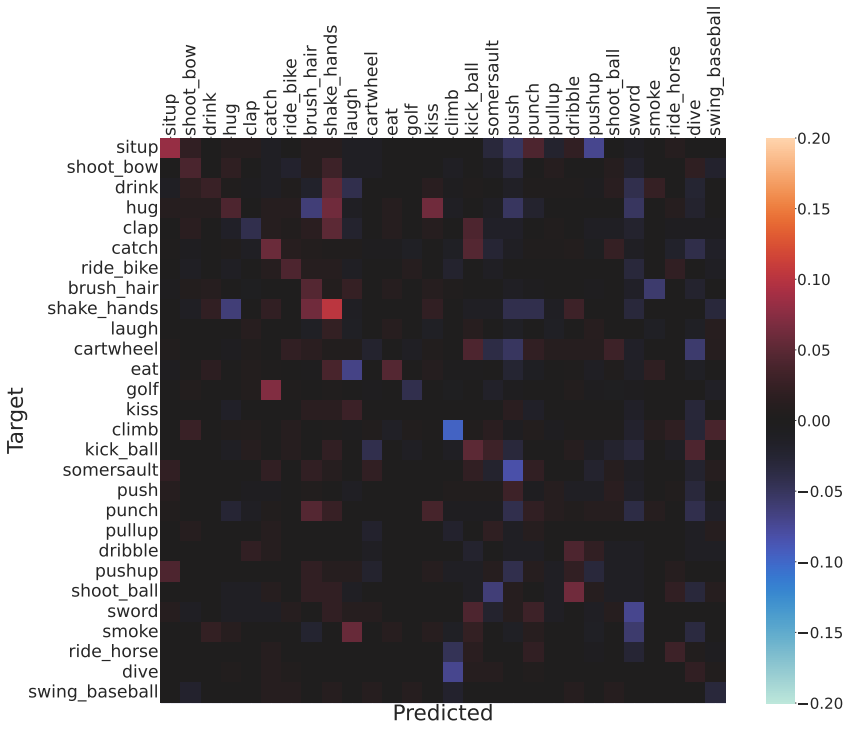
(a) HMDB-28



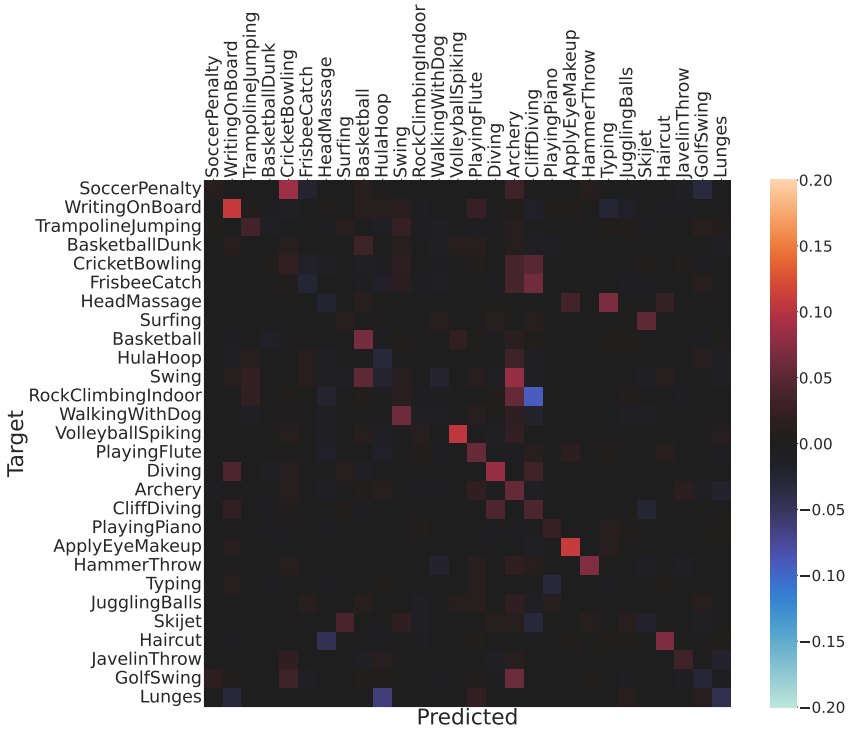
(b) UCF-65 (top 28 classes with the highest accuracy drop)

Figure 4: Confusion matrix difference between evaluating on the same dataset (a:HMDB at top or b:UCF at bottom) and a different one (Kinetics), given a TSM model with no augmentation strategy. The negative (blue) values on the diagonal line indicate the class accuracy drop when evaluated on Kinetics. Almost every class has a drastic drop. Overall, there is a 25.7% and 38.37% accuracy drop for the HMDB and UCF training datasets, respectively.

[3] Arno Blaas, Xavier Suau, Jason Ramapuram, Nicholas Apostoloff, and Luca Zappella. Challenges of adversarial image augmentations. In *I (Still) Can't Believe It's Not Better! Workshop at NeurIPS 2021*, 2021.



(a) HMDB-28



(b) UCF-65 (top 28 classes with the highest accuracy drop)

Figure 5: Confusion matrix difference between no augmentation strategy and the proposed curriculum adversarial augmentation training with the TSM model and evaluation on Kinetics. The positive (red) values on the diagonal line indicate the added class-wise accuracy by using the proposed approach. The classes are sorted by the drop in performance using the proposed cross-dataset evaluation, as seen in Fig. 4.

[4] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017.

[5] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

[6] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *Symposium on Security and Privacy*, 2017.

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[8] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[10] Junhao Dong, Seyed-Mohsen Moosavi-Dezfooli, Jianhuang Lai, and Xiaohua Xie. The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24678–24687, June 2023.

[11] Panagiotis Eustratiadis, Henry Gouk, Da Li, and Timothy M. Hospedales. Weight-covariance alignment for adversarially robust neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

[12] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[13] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020*, pages 877–894, 2021.

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[16] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 597–613. Springer, 2016.

[17] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

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

[20] Shreyank N Gowda, Marcus Rohrbach, and Laura Sevilla-Lara. Smart frame selection for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1451–1459, 2021.

[21] Shreyank N Gowda, Laura Sevilla-Lara, Kiyoon Kim, Frank Keller, and Marcus Rohrbach. A new split for evaluating true zero-shot action recognition. In *DAGM German Conference on Pattern Recognition*, pages 191–205. Springer, 2021.

[22] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 1–16. Springer, 2020.

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[24] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International conference on machine learning*, pages 2731–2741. PMLR, 2019.

[25] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.

[26] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[27] Ahmadreza Jeddi, Mohammad Javad Shafiee, Michelle Karg, Christian Scharfenberger, and Alexander Wong. Learn2perturb: An end-to-end feature perturbation learning to improve adversarial robustness. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[28] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[29] Kiyoon Kim, Shreyank N Gowda, Oisin Mac Aodha, and Laura Sevilla-Lara. Capturing temporal information in a single frame: Channel sampling strategies for action recognition. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. URL `https://bmvc2022.mpi-inf.mpg.de/0355.pdf`.

[30] Minyoung Kim, Da Li, and Timothy M. Hospedales. Domain generalisation via domain adaptation: An adversarial fourier amplitude approach. In *International Conference on Learning Representations (ICLR)*, 2023.

[31] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019.

[32] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[33] Kunchang Li, Yali Wang, Gao Peng, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *International Conference on Learning Representations*, 2022.

[34] Yonggang Li, Guosheng Hu, Yongtao Wang, Timothy Hospedales, Neil M Robertson, and Yongxin Yang. Differentiable automatic data augmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 580–595. Springer, 2020.

[35] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. *Advances in Neural Information Processing Systems*, 32, 2019.

[36] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019.

[37] Shiqi Lin, Zhizheng Zhang, Zhipeng Huang, Yan Lu, Cuiling Lan, Peng Chu, Quanzeng You, Jiang Wang, Zicheng Liu, Amey Parulkar, Viraj Navkal, and Zhibo Chen. Deep frequency filtering for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11797–11807, June 2023.

[38] Wei Lin, Muhammad Jehanzeb Mirza, Mateusz Kozinski, Horst Possegger, Hilde Kuehne, and Horst Bischof. Video test-time adaptation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22952–22961, June 2023.

[39] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[40] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[41] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*, 2017.

[42] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[43] Madeline Chantry Schiappa, Naman Biyani, Prudvi Kamtam, Shruti Vyas, Hamid Palangi, Vibhav Vineet, and Yogesh S. Rawat. A large-scale robustness analysis of video action recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14698–14708, June 2023.

[44] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations (ICLR)*, 2018.

[45] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[46] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[47] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[48] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

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

[50] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[51] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pages 4068–4076, 2015.

[52] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018.

[53] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1895–1904, 2021.

[54] Ruibin Wang, Yibo Yang, and Dacheng Tao. Art-point: Improving rotation robustness of point cloud classifiers via adversarial rotation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[55] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[56] Wenhao Wu, Dongliang He, Tianwei Lin, Fu Li, Chuang Gan, and Errui Ding. Mvfnet: Multi-view fusion network for efficient video recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2943–2951, 2021.

[57] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019.

[58] Chenyu Yi, SIYUAN YANG, Haoliang Li, Yap peng Tan, and Alex Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[59] Tianyuan Yu, Yongxin Yang, Da Li, Timothy M. Hospedales, and Tao Xiang. Simple and effective stochastic neural networks. In *Conference on Artificial Intelligence (AAAI)*, 2021.

[60] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020.

[61] Xu Zhang, Felix Xinnan Yu, Shih-Fu Chang, and Shengjin Wang. Deep transfer network: Unsupervised domain adaptation. *arXiv preprint arXiv:1503.00591*, 2015.

[62] Yue Zhao, Yuwei Wu, Caihua Chen, and Andrew Lim. On isometry robustness of deep 3d point cloud models under adversarial attacks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[63] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 803–818, 2018.

[64] Kaiyang Zhou, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Conference on Artificial Intelligence, (AAAI)*, 2020.