Shuffle and Attend: Video Domain Adaptation

Jinwoo Choi[1⋆], Gaurav Sharma[2], Samuel Schulter[2], and Jia-Bin Huang[1]

[1] Virginia Tech, Blacksburg, VA 24060, USA
{jinchoi,jbhuang}@vt.edu
[2] NEC Labs America, San Jose, CA 95110, USA

Abstract. We address the problem of domain adaptation in videos for the task of human action recognition. Inspired by image-based domain adaptation, we can perform video adaptation by aligning the features of frames or clips of source and target videos. However, equally aligning all clips is sub-optimal as not all clips are informative for the task. As the first novelty, we propose an attention mechanism which focuses on more discriminative clips and directly optimizes for video-level (cf. clip-level) alignment. As the backgrounds are often very different between source and target, the source background-corrupted model adapts poorly to target domain videos. To alleviate this, as a second novelty, we propose to use the clip order prediction as an auxiliary task. The clip order prediction loss, when combined with domain adversarial loss, encourages learning of representations which focus on the humans and objects involved in the actions, rather than the uninformative and widely differing (between source and target) backgrounds. We empirically show that both components contribute positively towards adaptation performance. We report state-of-the-art performances on two out of three challenging public benchmarks, two based on the UCF and HMDB datasets, and one on Kinetics to NEC-Drone datasets. We also support the intuitions and the results with qualitative results.

1 Introduction

Recent computer vision-based methods have reached very high performances in supervised tasks [2, 17, 18, 22, 51] and many real-world applications have been made possible such as image search, face recognition, automatic video tagging etc. The two main ingredients for success are (i) high capacity network design with an associated practical learning method, and (ii) large amounts of *annotated* data. While the first aspect is scalable, in terms of deployment to multiple novel scenarios, the second aspect becomes the limiting factor. The annotation issue is even more complicated in video-related tasks, as we need temporal annotation, i.e., we need to specify the start and end of actions in long videos. Domain adaptation has emerged as an important and popular problem in the community to address this issue. The applications of domain adaptation have ranged from simple classification [13, 33, 40, 47, 48, 56] to more complex tasks like semantic segmentation [5, 7, 20, 46, 49, 57] and object detection [1, 6, 19, 21, 26, 60]. However, the application on video tasks e.g., action recognition is still limited [3, 10, 23].

⋆ Part of this work was done when Jinwoo Choi was an intern at NEC Labs America.

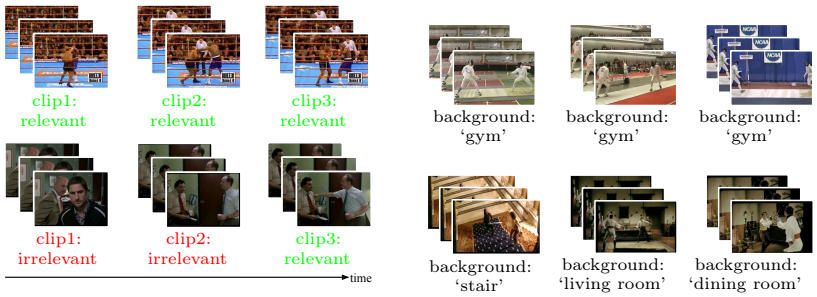


Fig. 1: **Motivation.** We do video domain adaptation and introduce the following two key components: (*Left*): Clip attention. The top video and the lower video have the same action *punching*. However, the lower video has only one relevant punching clip, while the top video has three relevant punching clips. Our proposed attention suppresses features from irrelevant clips, improving the feature alignment across domains. (*Right*): Clip order prediction. The top and bottom videos are from different domains, but all capture the action *fencing*. However, the backgrounds are different: the top domain has a gym as a background, and the lower domain has a dining room or a living room or a stair as a background. Predicting the order of clip encourages the model to focus more on the humans, not the background, as the background is uninformative for predicting temporal order.

We address this less studied but challenging and practically important task of video domain adaptation for human action recognition. We work in an unsupervised domain adaptation setting. That is, we have annotated data for the source domain and only *unannotated* data for the target domain. Examples domains that we use in experiments include (human) actions from movies, unconstrained actions from sports videos, YouTube videos, and even videos taken from drones.

We exploit two insights related to the problem and propose two novel adaptation components inspired by them. First, we note that the existing domain adaptation methods, when applied directly to the video adaptation task, sample frames or clips [3, 23], depending on whether the video encoding is based on a 2D network, e.g., temporal relation network [58] or a 3D network, e.g., C3D [44]. We sample clips (or frames) and then average the final outputs from multiple clips at test time, following the video classification networks they are built upon. Performing domain adaptation by aligning features for all sampled clips is suboptimal, as a lot of network capacity is wasted on aligning clips that are not crucial for the task. In the worst case, it can even be detrimental if a large number of unimportant clips dominate the learning loss and adversely affect the alignment of important clips. For example, in Figure 1 left, both the top video from one domain and the bottom video from another domain have the same action, *punching*. However, the bottom video contains a lot of clips irrelevant to *punching*. Aligning features from those irrelevant clips would not improve the target performance much.

Second, this clip-wise training method is likely to exploit correlations in the scene context for discriminating the action classes [9, 31, 32], e.g., in a formal sports-oriented dataset fencing might happen in a gym only as shown in the top right three videos of Figure 1. However, in the domain adaptation setting, the target domain might have vastly different scene contexts, e.g., the same fencing might happen in a living room or dining room, as shown in the bottom right three videos of Figure 1. When the source model uses the correlated gym information to predict a fencing action, it may perform poorly on the same class in the target domain, which does not have a gym scene. Similar scene context corruption issues have been identified for transfer learning, and few works have addressed the problem of *debiasing* the representations explicitly [9, 52].

Based on the above insights, we propose **Shuffle and Attend: Video domain Adaptation** (SAVA) with two novel components. First, we propose to identify and align *important* (which we define as *discriminative*) clips in source and target videos via an attention mechanism. The attention mechanism leads to the suppression of temporal background clips, which helps us focus on aligning only the important clips. Such attention is learned jointly for video-level adaptation and classification. We estimate the clip's importance by employing an auxiliary network and derive the video feature as the weighted combination of the identified important clip features.

Second, we propose to learn *spatial-background invariant human action representations* by employing a self-supervised clip order prediction task. While there could be some correlation between the scene context/background and the action class, e.g., soccer field for 'kicking the ball' action, the scene context is not sufficient for predicting the temporal clip order. In contrast, the actual human actions are indicative of the temporal order, e.g., for 'kicking the ball' action the clip order follows roughly the semantics of 'approaching the ball', 'swinging the leg' and 'kicking'; if we shuffle the clips, the actual human action representation would be able to recover the correct order, but the scene context based representation would be likely to fail.

Thus using the clip order prediction based loss helps us counter the scene context corruption in the action representations and improves adaptation performance. We employ the self-supervised clip order prediction task for both source and target. As this auxiliary task is self-supervised, it does not require any annotation (which we do not have for target videos).

We provide extensive empirical evaluations to demonstrate the benefits of the proposed method on three challenging video domain adaptation benchmark settings. We also give qualitative results to highlight the benefits of our system.

In summary, our contributions are as follows.

- We propose to learn to align important (discriminative) clips to achieve improved representation for the target domain.
- We propose to employ a self-supervised task which encourages a model to focus more on actual action and suppresses the scene context information, to learn representations more robust to domain shifts. The self-supervised task does not require extra annotations.
- We obtain state-of-the-art results on the HMDB to UCF adaptation benchmark, and Kinetics to NEC-Drone benchmarks.

2 Related Work

Action recognition. Action recognition using deep neural networks has shown quick progress recently, starting from two-stream networks [42] to 3D [2,44,51] or 2D and 1D separable CNNs [45,53] have performed very well on the task. More recent advances in action recognition model long-term temporal contexts [11,51]. However, most models still rely on target supervised data when finetuning on target datasets. In contrast, we are interested in unsupervised domain adaptation, where we do not have access to target labels during training.

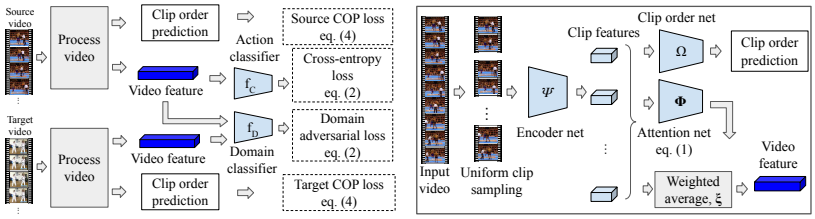
Unsupervised domain adaptation for images. Based on adversarial learning, domain adaptation methods have been proposed for image classification [13, 33,40,47,48,56], object detection [1,6,19,21,26,60], semantic segmentation [5, 7,20,46,49,57], and low-level vision tasks [39]. We also build upon adversarial learning. However, we work with videos and not still images.

Unsupervised domain adaptation for videos. Unlike image-related tasks, there are only a few works on video domain adaptation [3,10,23]. We also use the basic adversarial learning framework but improve upon it by adding auxiliary tasks that depend on the temporal order in videos, (i) to encourage suppression of spatial-background, and (ii) to focus on important clips in the videos to align.

Self-supervision. Image based self-supervised methods work with spatial context, e.g., by solving jigsaw puzzle [36], image inpainting [38], image colorization [29], and image rotation [15] to learn more generalizable image representation. In contrast, video based self-supervised methods exploit temporal context, e.g., by order verification [34], frame sorting [30], and clip sorting [54]. Recent video domain adaptation methods employ self-supervised domain sequence prediction [4], or self-supervised RGB/flow modality correspondence prediction [35].

We make a connection between the self-supervised task of clip order prediction [54] and learning a robust spatial-background decoupled representation for action recognition. We hypothesize (see Section 1) that, in combination with adversarial domain adaptation loss, this leads to suppression of domain correlated background, and simultaneous enhancement of the task correlated human part in the final representation leading to better domain adaptation performance.

Attention. There are numerous methods employing attention model for image [14,24] and video tasks [12,16,25,27,37,41,50,55]. The most closely related work is the that by Chen et al. [3]. While both the proposed method and Chen et al. [3] are based on attention, the main difference is in *what* they attend to. Chen et al. [3] attends to temporal relation features (proposed by Zhou et al. [58]) with *larger domain gaps*. In contrast, our proposed method attends to *discriminative* clip features. The clips in the same video may have different discriminative content, e.g., leg swinging (more discriminative) vs. background clips (less so) in a video of 'kicking a ball' class. The proposed method attends to more discriminative clips and focuses on aligning them. Chen et al. [3] samples $2 \sim 5$ frames relation features and attends to the ones with a larger domain gap measured by the entropy of the domain classifiers. However, the relation feature with a larger domain gap might come from frames irrelevant to the action, aligning them would be suboptimal. The proposed method addresses this problem. In another closely



(a) overall training (b) process video block

Fig. 2: **Overview of SAVA.** We employ standard domain adversarial loss along with two novel components. The first component is the self-supervised clip order prediction loss. The second is a clip attention based feature alignment mechanism. We predict attention weights for the uniformly sampled clips from the videos and construct the video feature as a weighted average of the clip features. Then we align the source and target video features. Best viewed with zoom and color.

related work Pan et al. [37] temporally align the source and target features using temporal co-attention and match their distributions. In contrast, the proposed method argues that human-focused representation is more robust to domain shifts, and captures it via self-supervised clip order prediction.

3 Method

We work in an unsupervised domain adaptation setting. We have (i) *annotated source data* $(\mathbf{x}_s, \mathbf{y}_s) \in \mathbf{X}^s \times \mathbf{Y}^s$, where $\mathbf{X}^s$ is the set of videos containing human-centered videos and $\mathbf{Y}^s$ is the actions label set, and (ii) *unannotated target data* $\mathbf{x}_t \in \mathbf{X}^t$. The task is to train a model, which performs well on the target data. Since the source data distribution, e.g., actions in movies, is expected to be very different from the target data distribution, e.g., actions in sports videos, the model trained on the source data only does not work well on target videos. The challenge is to design methods that can adapt a model to work on the target data, using both annotated source data and unannotated target data. The proposed method has three main components for adaptation: domain adversarial loss, clip order prediction losses, and an attention module for generating video features.

Figure 2 gives an overview of the proposed method, which we call Shuffle and Attend Video domain Adaptation (SAVA). We start with uniformly sampling N clips, with L frames, from an arbitrary length input video, as shown in Figure 2 (b). We encode source and target clips into clip features by an encoder network $\Psi(\cdot)$; which can be either the same for both or different. Here we assume it is the same for the brevity of notation. Then we use the clip features for (i) the clip order prediction network $\Omega(\cdot)$, and (ii) constructing the video-level features using the attention network $\Phi(\cdot)$. The video-level features obtained after the attention network, are then used with (i) a linear classifier, for source videos only, and (ii) a domain classifier, for both source and target videos, as shown in Figure 2 (a).

In total, there are three types of losses that we optimize, (i) domain adversarial loss, (ii) clip order prediction loss for both source and target, and (iii) classification

loss for source only. The clip order prediction loss works with clip level features, while the other two work on video-level features. As discussed in Section 1, the clip order prediction loss helps a model to learn a representation that is less reliant on correlated source data background. The attention network gives us the final video feature by focusing on important clips. The domain adversarial loss helps a model to align video-level features between source and target videos. All these are jointly learned and hence lead to a trained system that gives aligned representations and achieves higher action classification performance than the baselines. We now describe each of our proposed components individually in detail in the following subsections.

3.1 Clip order prediction

As shown on Figure 1 (right), the source videos of the same class may have correlations with similar background context [32], and the target videos of the same class might have a background which is vastly different from the source background. While the source model might benefit from learning representation, which is partially dependent on the correlated background, this would lead to poor target classification. To address this problem, we propose to employ clip order prediction (COP) to enable better generalization of the representation. COP would not be very accurate if a model focuses on the background as the background might not change significantly over time. However, the temporal evolution of the clip depends more on the humans performing actions, and possibly the objects. Thus, if we employ the COP, the representation would focus more on the relevant humans and objects, while relying less on the background.

We build our COP module upon the work by Xu et al. [54]. We show the illustration of the COP network Ω in Figure 3. We incorporate an auxiliary network, taking clip features as input, to predict the correct order of shuffled clips of an input video. We sample M clips, with L frames each, from an input video and shuffle them. The task of the module is to predict the order of the shuffled clips. We formulate the COP task as a classification task with $M!$ classes, corresponding to all permutation tuples of the clips, and consider the correct order tuple as the ground truth class. We concatenate clip features pairwise and pass them to a fully connected layer with ReLU activation followed by a dropout layer. Then we concatenate all of the output features and use a final linear classifier to predict the order of the input clips. Since this is a self-supervised task and requires no extra annotation, we can use the task for the videos from source, target, or both; we evaluate this empirically in Section 4.3.

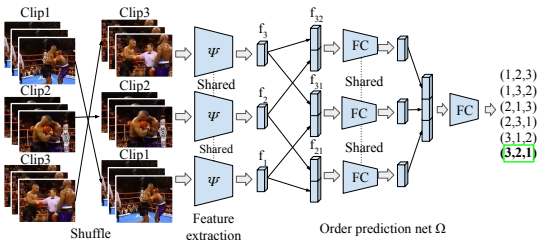


Fig. 3: **Clip order prediction network Ω (the layers after Ψ).**

3.2 Clip-attention based video-level features

As shown in the left side of Figure 1, all clips are not equally *important* (*discriminative* or *relevant*) for predicting the action. Aligning the irrelevant clip features is suboptimal, and it might even degrade performance if they dominate the loss cf. the important clips. Focusing on and aligning the important clips would lead to better adaptation and classification performance. To achieve such focus on important clips, we propose a clip attention module. The attention module takes N number of clip features as inputs, and outputs N softmax scores indicating the importance of each of them. The final video-level feature is obtained by the weighted average of the clip features. Formally, given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ as the N clips from an input video $\mathbf{x}$, we obtain the video-level feature $\mathbf{x}_v$ as

$$\mathbf{w} = \Phi(\Psi(\mathbf{x}_1), \ldots, \Psi(\mathbf{x}_N)), \quad \mathbf{x}^v = \xi(\mathbf{w}, \Psi(\mathbf{x}_1), \ldots, \Psi(\mathbf{x}_N)) = \sum_{i=1}^{N} w_i \Psi(\mathbf{x}_i), \quad (1)$$

where, $\xi(\cdot)$ is the weighted average function.

The attention module $\Phi(\cdot)$ is a network that takes N clip features with D dimension as an input. It outputs the importance vector $\mathbf{w} \in R^N$, which is used for weighted averaging to obtain the video-level feature. Thus, we can train the model end-to-end with the full domain adaptation system.

There can be multiple valid choices for the architecture of the attention module, e.g., a standard feed-forward network which takes concatenation of the clip features as input, or a recurrent network that consumes the clip features one by one. We explore two specific choices in an ablation experiment in Section 4.3, (i) Multi Layer Perceptron (MLP) similar to Kar et al. [25], and (ii) Gated Recurrent Units (GRU).

3.3 Training

We pre-train the attention module with standard binary cross-entropy loss, where we get the ground truth attention vector as follows. The ground truth label is 1 if the clip is correctly classified by the baseline clip-based classification network, and has confidence higher than a threshold c_{th}, and 0 otherwise. The pre-training makes the attention module to start from good local optima, mimicking the baseline classifier. Once pre-trained, the attention module can then either be fixed or can be trained end-to-end with the rest of the network. Please note that we train the attention module only on the source dataset as the training requires the ground truth action labels.

For the feature distribution alignment, we follow the well-known adversarial domain adaptation framework of ADDA [48]. We define our losses as,

$$\begin{aligned} L_{\text{CE}} &= -\mathbb{E}_{(\mathbf{x}_s, \mathbf{y}_s) \sim (\mathbf{X}^s, \mathbf{Y}^s)} \sum_{k=1}^{K} [y_{s,k} \log f_C(\mathbf{x}_s^v)], \\ L_{\text{ADV}_{f_D}} &= -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}^s}[\log f_D(\mathbf{x}_s^v)] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}^t}[\log(1 - f_D(\mathbf{x}_t^v)], \\ L_{\text{ADV}_{\psi_t}} &= -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}^t}[\log f_D(\mathbf{x}_t^v)], \end{aligned} \quad (2)$$

where f_C is the linear source classifier and f_D is the domain classifier. The video feature $\mathbf{x}^v = \xi(\mathbf{w}, \Psi(\mathbf{x}_1) \ldots, \Psi(\mathbf{x}_N))$ is the weighted average of clip level features, with weights $\mathbf{w} = \Phi(\Psi(\mathbf{x}_1), \ldots, \Psi(\mathbf{x}_N))$ obtained from the attention module. Then our optimization objective is as follows,

$$\theta_s^*, \theta_{f_C}^*, \theta_\Phi^* = \underset{\theta_s, \theta_{f_C}}{\operatorname{argmin}} L_{\mathrm{CE},\theta_\Phi}, \theta_{f_D}^* = \underset{\theta_{f_D}}{\operatorname{argmin}} L_{\mathrm{ADV}_{f_D}}, \theta_t^* = \underset{\theta_t}{\operatorname{argmin}} L_{\mathrm{ADV}_{\psi_t}}, \quad (3)$$

where θ_s is the parameter of the source encoder $\Psi_s(\cdot)$, θ_{f_C} is the parameter of the source classifier $f_C(\cdot)$, θ_t is the parameter of the target encoder $\Psi_t(\cdot)$, and θ_{f_D} is the parameter of the domain classifier $f_D(\cdot)$.

We optimize this objective function in a stage-wise fashion [48]. We first optimize the source cross-entropy loss L_{CE} over the source parameters θ_s and θ_{f_C} with the annotated source data. Then we freeze source model parameters θ_s and θ_{f_C}, and optimize the domain classification loss $L_{\mathrm{ADV}_{f_D}}$ over the domain classifier parameter θ_{f_D}, and the inverted GAN loss $L_{\mathrm{ADV}_{\psi_t}}$ over the target encoder parameter θ_t with both the labeled source and the unlabeled target data.

Clip order prediction. We define the COP loss as follows.

$$L_{\mathrm{COP}} = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim(\mathbf{X},\mathbf{Y})} \sum_{k=1}^{M!} [y_k \log f_O(\phi)]. \quad (4)$$

Here, f_O is the linear classification function for the COP, $\phi = \Omega(\Phi(\mathbf{x}_1), ..., \Phi(\mathbf{x}_M))$ is the ReLU activation of the MLP which takes M clip features as input. We can employ the L_{COP} for both source and target. We optimize the loss L_{COP} over the source encoder parameter θ_s, target encoder parameter θ_t, COP MLP parameter θ_Ω, and clip order cliassifier parameter θ_{f_O}.

3.4 Inference

At inference time, we remove the domain discriminator and clip order prediction network. We divide the input video into N clips and extract clip features. These features are then weight averaged with weights obtained using the attention network. The action classifier predicts the action using the video-level feature.

4 Experimental Results

4.1 Datasets

We show results on the publicly available benchmark based on the UCF [43] and HMDB [28] datasets. We further show the result in a more challenging setting where the source dataset is part of the Kinetics dataset [2], and the target dataset is drone-captured action dataset [10]. In the following, the direction of the arrow indicates the source (arrow start) to target (arrowhead).

UCF↔HMDB. Chen et al. [3] released the UCF-HMDB dataset for studying video domain adaptation. This dataset has 3,209 videos with 12 action classes. All the videos come from the original UCF [43] and HMDB [28] datasets. They

subsampled overlapping 12 classes out of 101/51 classes from the UCF/HMDB, respectively. There are two settings of interest, UCF → HMDB, and the other is HMDB → UCF. We show the performance of our method in both of the two settings. We use the official split provided by the authors [3].

Kinetics→NEC-Drone. We also test our method on a more challenging target dataset captured by drones [10]. The dataset contains $5K$ videos with 16 classes in total, while the domain adaptation subset used contains 994 videos from 7 classes, which overlap with Kinetics dataset. We use the official train/val/test split provided by Choi et al. [10]. We conduct domain adaptation experiments with Kinetics→NEC-Drone setting, which is more challenging than UCF↔HMDB as there is a more significant domain gap between source and target domains.

In all three settings, we report top-1 accuracy on the target dataset and compare it to other methods.

4.2 Implementation details

We implement our method with the PyTorch library. We use the I3D [2] network as our clip feature encoder architecture for both source and target. The source and target encoders are different from each other and do not share parameters. Both are initialized with the Kinetics pre-trained model weights and then trained further as appropriate. Such pre-training on a large dataset is common in domain adaptation, e.g. for images (ImageNet) [6,13,46,48] and videos (Sports-1M [23], Kinetics [4,35]). The input to the clip feature encoder is a 3 channels × 16 frames × 224 × 224 pixels clip. We set the number of clips per video to $N = 4$ via validation. During testing, we sample the same $N = 4$ number of clips. COP module is a 2-layer MLP with 512 hidden units. We sample $M = 3$ clips per video for the COP task by following Xu [54].

By using attention, we compute the weighted average of the clip-level softmax score as our final video-level softmax score. We evaluate two types of networks for the attention module. One is 4-layer MLP with 1024 hidden units in each layer, and the other is a GRU [8] with 1024 hidden units. We found GRU to be better in two out of the three cases (Section 4.3), so we report all results with GRU. We set the attention module's confidence threshold c_{th} as 0.96 for the UCF and HMDB and 0.8 for Kinetics by validation on the source dataset. We use 4-layer MLP with 4096 hidden units in each layer as our domain classifier.

We set the batch size to 72. The learning rate starts from 0.01, and we divide the learning rate by 10 after two epochs and ten epochs. We train models for 40 epochs. We set the weight decay to 10^{-7}. We use stochastic gradient descent with momentum 0.9 as our optimizer.

We follow the 'pre-train then adapt' training procedure similar to previous work [48]. (i) We train the feature extractor $\Psi(\cdot)$ with the COP loss (4). We train our feature extractor $\Psi(\cdot)$ on both source and target datasets as we do not require any labels. (ii) Given the trained feature extractor $\Psi(\cdot)$, we further train it on the labeled source and unlabeled target datasets with a domain classifier $f_D(\cdot)$ attached. We also train the attention module $\Phi(\cdot)$ on the labeled source dataset, given the trained feature extractor $\Psi(\cdot)$ from step 1. (iii) Given the

Table 1: Ablation experiments on the COP loss, on Kinetics→NEC-Drone.

Method	COP on Source	COP on Target	Top-1 acc (%)	Δ
Clip DA + COP	✓	✓	28.5	+ 11.3
Clip DA + COP	✓	×	25.9	+ 8.7
Clip DA + COP	×	✓	22.4	+ 5.2
Clip DA only	×	×	23.7	+ 6.5
Supervised source only	×	×	17.2	reference

Table 2: Ablation experiments on the clip attention on Kinetics→NEC-Drone.

Method	Align	Clip attention	Top-1 acc (%)
SAVA (ours)	video-level	✓	31.6
SAVA (ours) w/o. clip attention	video-level	×	30.3
Clip-level align	clip-level	×	28.5

feature extractor $\Psi(\cdot)$ and the attention module $\Phi(\cdot)$, we train our full model with the labeled source dataset and unlabeled target dataset.

4.3 Ablation study

We perform several ablation experiments to analyze the proposed domain adaptation method. We conduct the experiments on more challenging Kinetics→NEC-Drones setting except the attention module design choice experiment in Table 3, which we performed on the UCF, HMDB, and Kinetics datasets.

Effect of clip order prediction. Table 1 gives the results showing the effect of COP. Here, the source only is the I3D network trained on the source dataset, which we directly test on the target dataset without any adaptation. Clip-level domain adaptation (Clip DA) is the baseline where we randomly sample clips and align features of the clips without any attention. On top of the clip DA, we can optionally use the COP losses for either source or target or both.

Table 3: Effect of using different attention implementation. We show the attention module accuracy (%) on the Kinetics, UCF, and HMDB datasets.

Method	No. params	Kinetics	UCF	HMDB
MLP	6.3M	72.2	86.1	75.4
GRU	6.3M	78.0	78.9	76.6

The clip DA only (without COP) improves performance over the supervised source only baseline by 6.5%p. More interestingly, the results show that using both source and target COP improves performance significantly compared to the clip DA only baseline by 4.8%p. We also observe that the source COP is more crucial compared to the target COP. This is because the target NEC-Drone

dataset (i) contains similar background appearance across all videos (a high school gym), (ii) has a limited number of training videos ($\sim 1K$), and (iii) has the main activities occurring with small spatial footprint (as the actors are small given the videos were captured by drones). Thus, applying COP on the NEC-Drone dataset does not lead to improved results. However, applying COP on the source or both source/target produces large improvements over the baseline.

Clip attention performance. We evaluate the two different design choices, MLP and GRU, for our attention module in this experiment. We show the clip attention accuracy on the three source datasets in Table 3. We get the attention accuracy by comparing the ground truth importance label (see Section 3.3 for the details) and the predicted importance. We compute the clip attention performance on the three source datasets Kinetics, UCF, and HMDB. Using such curated ground truth ensures that the attention module starts from good local minima, which is in tune with the base I3D encoder network.

The GRU shows a higher attention performance in two out of the three cases, while it has a similar number of parameters to the MLP-based attention. Thus, we employ the GRU-based attention module on all experiments in this paper.

Effect of attention module. We show the effect of the attention module in the overall method in Table 2. Here, all the methods are pre-trained using source and target COP losses turned on. We train our domain adaptation network with three settings, (i) video-level alignment with clip attention (our full model), (ii) video-level alignment without clip attention (using temporal average pooling instead), and finally (iii) clip-level alignment.

The results show that video-level alignment gives an improvement over random clip sampling alignment, 30.3% vs. 28.5%. Our full model with clip attention alignment further improves the performance to 31.6%, over video-level alignment without attention. The video-level alignment without attention treats every clip equally. Hence, if there are some non informative clips, e.g., temporal background, equally aligning those clips is a waste of the network capacity. Our *discriminative* clip attention alignment is more effective in determining more discriminative clips and doing alignment based on those.

4.4 Comparison with other methods

Methods compared. The methods reported are (i) 'supervised source only': the network trained with supervised source data (a lower bound for adaptation methods), (ii) 'supervised target only': the network trained with supervised target data (an upper bound for the adaptation methods), and (iii) different unsupervised domain adaptation methods. For the TA^{3}N, we compare with the latest results obtained by running the public code[3] provided by the authors [3] and not the results in the paper (given in brackets for reference, in Table 4). While the original TA^{3}N [3] works with 2D features based temporal relation network (TRN) [58], we go beyond and integrate the TA^{3}N with stronger I3D [2] based TRN features. This allows a fair comparison with our method when all

[3] https://github.com/cmhungsteve/TA3N

Table 4: Results on UCF↔HMDB.

Method	Encoder	UCF→HMDB	HMDB→UCF
Supervised source only [3]	ResNet-101-based TRN	73.1 (71.7)	73.9 (73.9)
TA^{3}N [3]	ResNet-101-based TRN	75.3 (78.3)	79.3 (81.8)
Supervised target only [3]	ResNet-101-based TRN	90.8 (82.8)	95.6 (94.9)
Supervised source only [3]	I3D-based TRN	80.6	88.8
TA^{3}N [3]	I3D-based TRN	81.4	90.5
Supervised target only [3]	I3D-based TRN	93.1	97.0
TCoN [37]	ResNet-101-based TRN	87.2	89.1
Supervised source only	I3D	80.3	88.8
SAVA (ours)	I3D	82.2	91.2
Supervised target only	I3D	95.0	96.8

other factors (backbone, computational complexity, etc) are similar. For TCoN, we report the numbers from the paper [37] as code is not publicly available.

For the Kinetics → NEC-Drone setting, we implement video versions of the DANN [13] and ADDA [48], which align the clip-level I3D [2] features and show the results. We also compare with both unsupervised and semi-supervised methods of Choi et al. [10].

UCF → HMDB. We compare our method with existing methods in Table 4. The first three blocks contain the results of the method with TRN-based encoder [58]. The fourth block shows the results of our SAVA with domain adaptation as well as the source only I3D [2] baseline. We also show the result of fully supervised finetuning of the I3D network on the target dataset as an upper bound.

SAVA with the I3D-based encoder shows 82.2% top-1 accuracy on the HMDB dataset, in this setting. SAVA improves the performance of the strong I3D encoder, 80.3%, which in itself obtains better results than the TRN-based adaptation results, 75.3% with TA^{3}N. Our SAVA is closer to the upper bound (82.2% vs. 95.0%), than the gap between TA^{3}N and its upper bound (75.3% vs. 90.8%). Furthermore, SAVA outperforms TA^{3}N with I3D-based TRN features, 81.4%.

HMDB → UCF. Table 4 gives the comparison of our method with existing methods in this setting. We achieve state-of-the-art results in this setting while the other trend is similar to the UCF→HMDB setting. SAVA achieves 91.2% accuracy on the target dataset with domain adaptation and without using any target labels. The baseline source only accuracy of the I3D network is already quite strong cf. the existing best adaptation method, i.e., 88.8% vs. 90.5% for TA^{3}N with I3D-based TRN features. We improve this to 91.2%. SAVA is quite close to the upper bound of 96.8%, which strongly supports the proposed method. In contrast, TA^{3}N is still far behind its upper bound (79.3% vs. 95.6%).

Kinetics → NEC-Drone. This setting is more challenging, as the domain gap is larger, i.e., the gap between the source only and target finetuned classifiers is 64.5% cf. 14.7% for UCF→HMDB.

Table 5: Results on the Kinetics→NEC-Drone.

Method	Encoder	Target labels used (%)	Top-1 acc (%)
Supervised source only [3]	ResNet-101-based TRN	None	15.8
TA^{3}N [3]	ResNet-101-based TRN	None	25.0
Supervised source only [3]	I3D-based TRN	None	15.8
TA^{3}N [3]	I3D-based TRN	None	28.1
Supervised source only	I3D	None	17.2
DANN [13]	I3D	None	22.3
ADDA [48]	I3D	None	23.7
Choi et al. [10] (on val set)	I3D	None	15.1
SAVA (ours)	I3D	None	31.6
Choi et al. [10]	I3D	6	32.0
Supervised target only	I3D	100	81.7

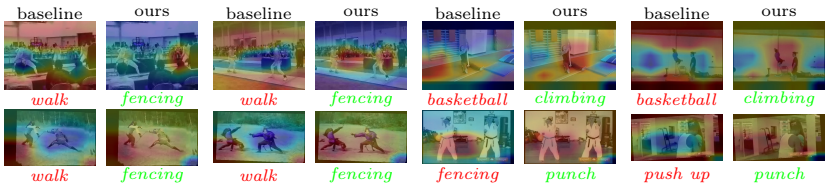


Fig. 4: **Class activation maps (CAM) on the UCF (first row) and HMDB (second row) datasets.** The actions green are correct predictions, and those in red are incorrect predictions. Here the baseline is ADDA without COP, and ours is ADDA with COP. Note how the COP encourages the model to focus more on human action instead of scene context.

Table 5 gives the results. The first block uses the TRN with ResNet-101 features, and the second block uses the TRN with I3D features while the others use I3D features. We observe that similar to previous cases, SAVA outperforms all methods, e.g., DANN (42% relative), ADDA (33% relative), TA^{3}N (26.4% relative), TA^{3}N with I3D features (12.4% relative), and Choi et al. (the unsupervised domain adaptation case). It is very close to the semi-supervised result of Choi et al. (31.6 vs. 32.0), where they use 5 target labeled examples per class.

While the improvements achieved by SAVA are encouraging in this challenging setting, the gap is still significant, 31.6% with adaptation vs. 81.7% with the model finetuned with the target labels. The gap highlights the challenging nature of the dataset, and the large margin for improvement in the future, for video-based domain adaptation methods.

4.5 Qualitative evaluation

Clip order prediction. To better understand the effect of the proposed COP module, we show class activation maps (CAM) [59] of target videos in Figure 4. We compute the CAM of the center (8th) frame of a 16 frames long clip. We

Fig. 5: **Attention visualization on center frames of 4 clips from 4 videos.** The frames with green borders are given more importance by our attention module cf. those with red borders. Note that our attention module can attend to relevant clips where the action is clearly visible, while the baseline without attention would align all clips equally, even those where the actor is missing or highly occluded.

show CAMs from models with and without COP (baseline/ours). The baseline without COP tends to focus more on the scene context. However, the proposed model with COP focuses more on the actual human action (typically around the actors). As the model with COP focuses more on the actual action, it generalizes better to a new domain with a completely different scene cf. the model without COP, which is heavily biased by the scene context.

Clip attention. We show the center frames of 4 clips per video with the clip attention module based selection. The videos demonstrate how the proposed clip attention module focuses more on the action class relevant clips and less on the irrelevant clips with either highly occluded actors or mainly background. E.g., in the fencing video in the second row, first and the fourth clips are not informative as the actor, or the object (sword), is highly occluded or cropped. Thus, aligning the features from the relevant second and the third clips is encouraged. Similarly, in the golf video of the first row, the last clip (green background) is irrelevant to the golf action, and our attention module does not attend to it. However, a model without attention treats all the clips equally.

5 Conclusion

We proposed **Shuffle and Attend: Video domain Adaptation** (SAVA), a novel video domain adaptation method with self-supervised clip order prediction and clip attention based feature alignment. We showed that both of the two components contribute to the performance. We achieved state-of-the-art performance on the publicly available HMDB→UCF and Kinetics→Drone datasets. We showed extensive ablation studies to show the impact of different aspects of the method. We also validated the intuitions for designing the method with qualitative results for both the contributions.

Acknowledgment. This work was supported in part by NSF under Grant No. 1755785 and a Google Faculty Research Award. We thank NVIDIA Corporation for the GPU donation.

References

1. Cai, Q., Pan, Y., Ngo, C.W., Tian, X., Duan, L., Yao, T.: Exploring object relation in mean teacher for cross-domain detection. In: CVPR (2019)
2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
3. Chen, M.H., Kira, Z., AlRegib, G., Woo, J., Chen, R., Zheng, J.: Temporal attentive alignment for large-scale video domain adaptation. In: ICCV (2019)
4. Chen, M.H., Li, B., Bao, Y., AlRegib, G., Kira, Z.: Action segmentation with joint self-supervised temporal domain adaptation. In: CVPR (2020)
5. Chen, M., Xue, H., Cai, D.: Domain adaptation for semantic segmentation with maximum squares loss. In: ICCV (2019)
6. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: CVPR (2018)
7. Chen, Y.C., Lin, Y.Y., Yang, M.H., Huang, J.B.: Crdoco: Pixel-level domain transfer with cross-domain consistency. In: CVPR (2019)
8. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: EMNLP (2014)
9. Choi, J., Gao, C., Messou, J.C., Huang, J.B.: Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In: NeurIPS (2019)
10. Choi, J., Sharma, G., Chandraker, M., Huang, J.B.: Unsupervised and semi-supervised domain adaptation for action recognition from drones. In: WACV (2020)
11. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV (2019)
12. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization of actions with actoms. TPAMI **35**(11), 2782–2795 (2013)
13. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML (2015)
14. Gao, C., Zou, Y., Huang, J.B.: ican: Instance-centric attention network for human-object interaction detection. In: BMVC (2018)
15. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
16. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. In: NeurIPS (2017)
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
19. He, Z., Zhang, L.: Multi-adversarial faster-rcnn for unrestricted object detection. In: ICCV (2019)
20. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
21. Hsu, H.K., Yao, C.H., Tsai, Y.H., Hung, W.C., Tseng, H.Y., Singh, M., Yang, M.H.: Progressive domain adaptation for object detection. In: WACV (2020)
22. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
23. Jamal, A., Namboodiri, V.P., Deodhare, D., Venkatesh, K.: Deep domain adaptation in action space. In: BMVC (2018)
24. Jetley, S., Lord, N.A., Lee, N., Torr, P.H.: Learn to pay attention. In: ICLR (2018)
25. Kar, A., Rai, N., Sikka, K., Sharma, G.: Adascan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos. In: CVPR (2017)

26. Khodabandeh, M., Vahdat, A., Ranjbar, M., Macready, W.G.: A robust learning approach to domain adaptive object detection. In: ICCV (2019)
27. Korbar, B., Tran, D., Torresani, L.: Scsampler: Sampling salient clips from video for efficient action recognition. In: ICCV (2019)
28. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A large video database for human motion recognition. In: ICCV (2011)
29. Larsson, G., Maire, M., Shakhnarovich, G.: Colorization as a proxy task for visual understanding. In: CVPR (2017)
30. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV (2017)
31. Li, Y., Vasconcelos, N.: Repair: Removing representation bias by dataset resampling. In: CVPR (2019)
32. Li, Y., Li, Y., Vasconcelos, N.: Resound: Towards action recognition without representation bias. In: ECCV (2018)
33. Luo, Z., Zou, Y., Hoffman, J., Fei-Fei, L.F.: Label efficient learning of transferable representations acrosss domains and tasks. In: NeurIPS (2017)
34. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV (2016)
35. Munro, J., Damen, D.: Multi-modal domain adaptation for fine-grained action recognition. In: CVPR (2020)
36. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
37. Pan, B., Cao, Z., Adeli, E., Niebles, J.C.: Adversarial cross-domain action recognition with co-attention. In: AAAI (2020)
38. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016)
39. Ren, Z., Jae Lee, Y.: Cross-domain self-supervised multi-task feature learning using synthetic imagery. In: CVPR (2018)
40. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: CVPR (2018)
41. Sikka, K., Sharma, G.: Discriminatively trained latent ordinal model for video classification. TPAMI **40**(8), 1829–1844 (2017)
42. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NeurIPS (2014)
43. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
44. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV (2015)
45. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR (2017)
46. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative representations. In: ICCV (2019)
47. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV (2015)
48. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR (2017)
49. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Dada: Depth-aware domain adaptation in semantic segmentation. In: ICCV (2019)
50. Wang, J., Wang, W., Huang, Y., Wang, L., Tan, T.: M3: Multimodal memory modelling for video captioning. In: CVPR (2018)
51. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)

52. Wang, Y., Hoai, M.: Pulling actions out of context: Explicit separation for effective combination. In: CVPR (2018)
53. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning for video understanding. In: ECCV (2018)
54. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: CVPR (2019)
55. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: ACM MM (2017)
56. Zhang, J., Li, W., Ogunbona, P.: Joint geometrical and statistical alignment for visual domain adaptation. In: CVPR (2017)
57. Zhang, Q., Zhang, J., Liu, W., Tao, D.: Category anchor-guided unsupervised domain adaptation for semantic segmentation. In: NeurIPS (2019)
58. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: ECCV (2018)
59. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. CVPR (2016)
60. Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: CVPR (2019)