Dual-Head Contrastive Domain Adaptation for Video Action Recognition

Victor G. Turrisi da Costa[1], Giacomo Zara[1], Paolo Rota[1], Thiago Oliveira-Santos[2],
Nicu Sebe[1], Vittorio Murino[3,4] and Elisa Ricci[1,5]
[1]University of Trento, [2]Universidade Federal do Espírito Santo
[3]University of Verona, [4]Huawei Technologies, Ireland Research Center, [5]Fondazione Bruno Kessler
{vg.turrisidacosta,giacomo.zara,paolo.rota,niculae.sebe,e.ricci}@unitn.it
todsantos@inf.ufes.br, vittorio.murino@iit.it

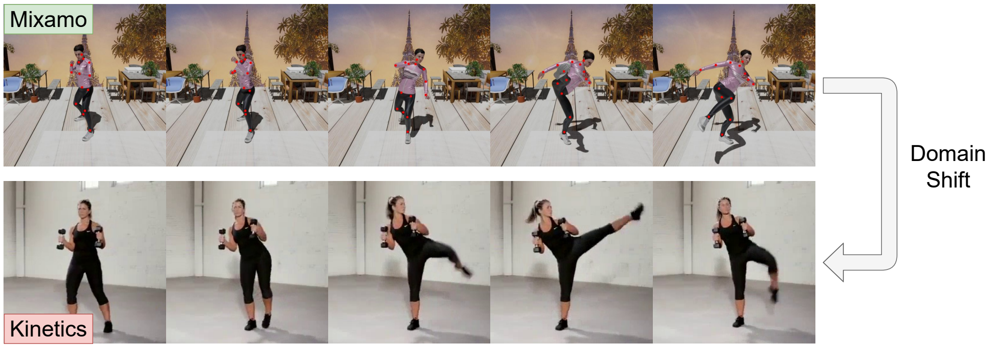

Figure 1: Sample sequences from the Mixamo and the Kinetics datasets. Keypoints are also provided for the Mixamo dataset.

Abstract

Unsupervised domain adaptation (UDA) methods have become very popular in computer vision. However, while several techniques have been proposed for images, much less attention has been devoted to videos. This paper introduces a novel UDA approach for action recognition from videos, inspired by recent literature on contrastive learning. In particular, we propose a novel two-headed deep architecture that simultaneously adopts cross-entropy and contrastive losses from different network branches to robustly learn a target classifier. Moreover, this work introduces a novel large-scale UDA dataset, Mixamo→Kinetics, which, to the best of our knowledge, is the first dataset that considers the domain shift arising when transferring knowledge from synthetic to real video sequences. Our extensive experimental evaluation conducted on three publicly available benchmarks and on our new Mixamo→Kinetics dataset demonstrate the effectiveness of our approach, which outperforms the current state-of-the-art methods. Code is available at `https://github.com/vturrisi/CO2A`.

1. Introduction

Visual recognition models are built under the assumption that the training and test data are drawn from the same distribution. Unfortunately, this assumption rarely holds in practice, leading to a drop in performance on the test data. To address this problem, over the years, several unsupervised domain adaptation (UDA) methods [11] have been developed. UDA approaches leverage relevant knowledge from labelled data in a source domain to learn a model for a different, but related, target domain where no annotations are provided. These methods have already proved to be effective in several image-related tasks, ranging from object recognition [30, 41, 50, 29] to semantic segmentation [19, 57, 18, 8] and object detection [24]. However, so far much less attention has been devoted to video analysis which, compared to image-related applications, is undoubtedly more challenging. In particular, videos introduce one more level of variation in the data, *i.e.* the temporal dimension, which increases the demand for hardware and leads to additional complexity. To address UDA in the context of video analysis, researchers have proposed to rethink the traditional strategies for images in order to learn robust classifiers for videos, using domain-invariant deep feature representations [9, 10, 34, 35, 5].

Action recognition [13, 58, 48, 4] is one of the fundamental problems in video analysis. This task is inherently challenging as actions can vary over time according to several factors, such as speed, duration, relative movement between the actor and the camera, and the actor's interaction with surrounding objects. Also, people can perform the same action in different ways, raising a challenging ambiguity. Although Convolutional Neural Networks-based (CNN) approaches have enabled significant advances, this task still poses many open problems. In particular, the important variation derived from video sequences makes the domain shift harder to address compared to the case of images. One way to address the variation issue without increasing the cost of data acquisition is to rely on synthetic data; however, such data still present the challenge of the large domain gap.

This paper advances the state of the art in UDA for video action recognition by proposing a novel two-headed deep architecture. The design of our model is motivated by the idea of jointly leveraging source supervision, target pseudo-labelling and contrastive learning to mitigate the domain shift arising in video action recognition. Our network consists of a shared encoder that extracts feature representations from clips of source and target videos and aggregates them with an attention mechanism. The video-level features provided by the encoder are then fed to two separate network heads that learn complementary classification models, one based on a cross-entropy loss and the other trained with contrastive losses. A key element of our approach is a novel consistency loss term that enforces the network to produce coherent predictions among the two network heads, resulting in more reliable pseudo-labels for the target samples. Target pseudo-labels and source labels are then jointly exploited by a novel inter-domain contrastive loss, which performs conditional feature alignment among data distributions of different domains, thus counteracting domain shift. Lastly, inspired by recent literature on contrastive learning [6], we leverage video-specific data augmentations, both at clip and video-level, to learn multi-scale spatio-temporal feature representations for target videos. Our Contrastive Conditional domain Alignment approach is named as CO^{2}A.

An important contribution of this work is also the introduction of *Mixamo→Kinetics*, a new large-scale dataset for video action recognition. The proposed dataset is the first benchmark that allows studying the challenging problem of UDA when source data are synthetically generated videos and target data are real Youtube videos. In our dataset, frame sequences in the synthetic domain are generated using realistic motion sequences gathered from Mixamo[1] and rendered using Blender[2], resulting in videos depicting actions performed by 3D avatars with different visual appearances in a randomised 3D scene. Data of the real domain, instead, are obtained from the popular Kinetics dataset [3]. Sample frames for two sequences of our dataset are shown in Figure 1.

Contributions. To summarise, our contributions are the following: (i) a UDA approach for action recognition in videos that exploits label and pseudo label information for semantic alignment of the source and target data distributions. The proposed method achieves state-of-the-art performance on several challenging benchmarks for action recognition, such as *UCF↔HMDB*[5], *UCF↔Olimpics Sports*[5] and *Kinetics→NEC-Drone*[9]; (ii) a novel deep architecture that seamlessly integrates three components: a dual head structure to learn two different but coherent models (based on classification loss and contrastive losses, respectively), an inter-domain contrastive loss, which exploits source labels and target pseudo-labels for domain distribution alignment, and a multi-level contrastive loss for target feature learning; (iii) a novel large-scale synthetic-to-real dataset, *Mixamo→Kinetics*, devised for testing UDA methods for action recognition.

To the best of our knowledge, the dataset we propose is the only benchmark that will be publicly available for assessing the ability of UDA approaches to transfer knowledge from the synthetic to the real domain in videos.

2. Related Work

Action recognition. Different deep architectures for action recognition have been proposed in the last few years. For instance, in [13], two-stream networks were proposed to jointly use RGB and optical flow frames within two 2D CNNs, modelling temporal information. Zhou *et al.* [58] introduced Temporal Relation Networks, a deep model that employs a specialised pooling layer to model temporal relations between frames. Other works considered 3D CNNs to learn spatio-temporal features. Tran *et al.* [48] proposed C3D, which directly employs 3D convolutions rather than 2D ones. Carreira *et al.* [4] introduced I3D, a deep network that integrates inflated 2D convolutional filters to leverage large-scale pre-trained 2D models. Very recently, some other works have proposed approaches based on contrastive learning for extracting useful motion representations for action recognition [56, 39, 54, 37]. Different from our work, all mentioned studies tackle the traditional supervised action recognition problem (no domain shift).

UDA for images. Existing approaches mostly differ on the strategy used to cope with domain shift. One category of methods performs domain distribution alignment by matching statistical moments of the first and second-order of the source and target data distributions [28, 2, 40, 31, 50]. Recently, these methods were improved considering label information during the alignment process [21]. Another prominent strategy in UDA is adversarial training [27, 19, 15, 49], where discriminative and domain-agnostic feature representations are learned by coupling a domain discriminator with the source classification loss. Similar to moment matching methods, the best performing approaches of this category leverage the semantic informa-

[1] https://www.mixamo.com/
[2] https://www.blender.org/

tion given by classifier predictions to perform adversarial adaptation. Generative adversarial networks [16] have also been considered to address the domain shift [17, 53, 42], as they permit to generate target-like images that can then be used to train a target classification model. Furthermore, recent works have considered self-supervised learning and auxiliary tasks, such as predicting rotations [47] or image patches permutations in a jigsaw setting [1] to learn domain-invariant features.

Our work is related to previous methods based on semantic distribution alignment [27, 21, 33], but innovates over past literature since adaptation is performed thanks to a novel domain contrastive alignment loss. CO^2A also shares some similarities with previous UDA approaches with double classifier structure [41]. However, one particular aspect of CO^2A is the choice of the supervised contrastive loss [22] for one of the two network heads. Lastly, contrastive learning for UDA has been recently studied in [36, 23]. However, the deep architectures in [36, 23] are radically different from ours and do not exploit contrastive learning within a two-headed neural network. Moreover, these works do not tackle the more challenging problem of video action recognition.

UDA for action recognition. Despite its importance in many real-world applications, only a few works addressed the problem of domain shift for action recognition [34, 35, 5, 9, 10]. Chen *et al.* [5] proposed the Temporal Attentive Adversarial Adaptation Network (TA^3N), which integrates a temporal relation module to simultaneously learn the temporal dynamics and achieve domain alignment. Pan *et al.* [35] introduced Temporal Co-attention Network (TCoN), a deep architecture with a cross-domain attention module to match the distributions of temporally aligned features between source and target domains. In [9], the problem of UDA for recognising actions was considered in the specific case of videos collected by drones and an adversarial adaptation framework was proposed. Furthermore, in [10], the problem of open-set domain adaptation has been also investigated. Choi *et al.* [10] introduced an attention mechanism to determine discriminative clips and used this information for video-level alignment within an adversarial learning framework. In [34], a domain adaptation approach based on self-supervision and multimodal learning (RGB+optical flow) was proposed for fine-grained first-person view action recognition. RGB+optical flow modalities were also exploited in [45] within a contrastive approach. Whereas [45] considers positives using the same data on another modality and negatives by perturbing the frames temporally, we consider positives and negatives in different ways: using the real labels for a source only contrastive loss, using real and pseudo-labels for an across domain contrastive loss, and using different augmentations for a target only contrastive. None of these previous works considers a dual-head contrastive framework for learning and aligning source and target video representations.

Table 1: UDA benchmarks for video action recognition

Dataset	**# classes**	**# videos**	1^{st} person	3^{rd} person	**Methods**
HMDB↔UCF	12	3,209		✓	[10] [5] [35]
Kinetics↔NEC Drone	7	994		✓	[9] [10]
UCF↔Olympic Sports	6	1,145		✓	[5] [35]
Charades-Ego dataset	157	4,000	✓	✓	[44] [9]
EPIC Kitchens DA	8	$\sim 8,500$	✓		[34]
Mixamo→Kinetics	14	36,195		✓	This work

UDA benchmarks for action recognition. Table 1 provides an overview of the publicly available benchmarks for UDA and video action recognition along with the previous UDA methods that have considered them. Only three datasets, *HMDB↔UCF* [5], *Kinetics→NEC Drone* [9] and *UCF↔Olympic Sports* [5] are available in a third-person view setting. Additionally, two other datasets for domain adaptation in a first-person view setting have been introduced, EPIC Kitchens [12] DA, and, in the hybrid first-person/third-person view settings, Charades-Ego dataset [44]. However, first-person and third-person views setting are quite different in terms of visual appearance. The Jester dataset used in [35] has not been publicly released, whereas the Gameplay dataset considered in [5] only addresses the real→synthetic scenario. As shown in Table 1, the proposed *Mixamo→Kinetics* dataset is significantly larger than the existing benchmarks. Furthermore, it can be easily extended in the future by generating more synthetic data. There are other synthetically-generated datasets for action recognition, such as SURREAL [52], SURREACT [51] and 3DPeople [38]. However, all of them render synthetic humans over a static photo background. Furthermore, they have not been generated with the purpose of UDA, and the overlap in terms of categories with existing datasets of real videos, *e.g.* Kinetics [3], is very limited.

3. Mixamo→Kinetics dataset

Synthetically generated images and videos are nowadays recognised as an important resource in the computer vision community and are widely used in many tasks. By using computer graphics software and simulators, it is possible to generate large-scale datasets with virtually infinite visual variability and with annotations readily available. However, when models are trained on synthetic data but tested on images and videos from the real world, the problem of domain shift naturally arises. This section describes *Mixamo→Kinetics*, the first large scale dataset for benchmarking domain adaptation methods for action recognition in the challenging task of transferring knowledge from the synthetic to the real domain. Our dataset comprises $36,195$ videos, divided into 14 action categories and two domains, *i.e.*, the source domain (synthetic videos from Mixamo) and the target domain (real videos from Kinetics).

Source dataset (Mixamo). It consists of $24,533$ videos synthetically generated using the 3D characters from Mixamo. The dataset comprises videos depicting actions performed by 6 distinct avatars, with different backgrounds,

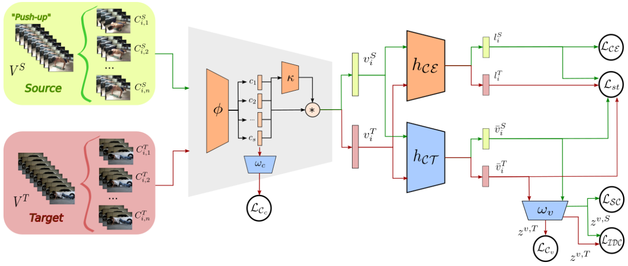

Figure 2: Overview of the proposed CO^{2}A approach.

camera positions and random 3D objects in the scene. Also, key-points are provided for each character following the scheme from the COCO dataset [26], but without the key-points for eyes and ears. Each frame is generated with a resolution of 512 by 512 and the mean length of the videos is 138 frames. To generate each video, we first created a Blender graphic environment; subsequently, for each video, the background and floor images were randomly selected from a set of ∼200 natural images and geometric patterns (*e.g.*, wood floor or other tiling patterns), all collected from the web. Different images were chosen for the background and the floor to avoid the unnatural effect of a “floating” avatar. We further enriched each scene with random 3D objects of varying shape, size and position. Since the objects were positioned using a predefined reference grid around the character, partial occlusions could be performed without the risk of completely hiding the character. Finally, we added to the scene a static sun-like light source and rendered it from 8 different camera angles. Using the light source, it was possible to produce realistic shadows for both the characters and the 3D objects in the scene, which is not possible on datasets that simply place a character in front of a background image. We also plan on generating a larger version of the dataset, with more camera angles and moving light source.

Target dataset (Kinetics). The target dataset was created considering $11,662$ videos from 14 action categories extracted from the Kinetics dataset [3]. The overlapping actions between the two datasets are *swing dancing*, *breakdancing*, *salsa dancing*, *throwing*, *capoeira*, *jogging*, *shouting*, *side kick*, *clapping*, *texting*, *golf putting*, *squat*, *punching* and *backflip*. Additional details about the dataset are provided in the supplementary material.

4. UDA for Action Recognition

4.1. Problem and Notation

The problem of UDA for action recognition can be formalised as follows. Let $\mathcal{X}$ be the sequence of frames from videos and $\mathcal{Y}$ the set of action categories. Given a labelled source domain $\mathcal{S}$ and an unlabelled target domain $\mathcal{T}$, the aim is to learn a function $f_\Theta : \mathcal{X} \to \mathcal{Y}$, where Θ denotes a model's parameters, that successfully predicts the corresponding action category from videos of the target domain. Since no annotation is available for the target domain, the training process leverages information from labelled videos of the source domain. The training set $T = T_\mathcal{S} \bigcup T_\mathcal{T}$ is composed by $N_\mathcal{S}$ annotated videos in the source domain $T_\mathcal{S} = \{(V^S_1, Y^S_1), \ldots, (V^S_{N_\mathcal{S}}, Y^S_{N_\mathcal{S}})\}$ and $N_\mathcal{T}$ unlabelled videos from the target domain $T_\mathcal{T} = \{V^T_1, \ldots, V^T_{N_\mathcal{T}}\}$. The main challenge of learning f_Θ lies in addressing the domain shift, *i.e.* the fact that the data from the two domains are drawn from two different distributions $p_\mathcal{S}(V, Y)$ and $p_\mathcal{T}(V, Y)$ over $\mathcal{X} \times \mathcal{Y}$.

4.2. Proposed Architecture

Overview. An overview of the proposed architecture is illustrated in Figure 2. First, source and target videos are divided into K non-overlapping parts of equal size, denominated clips. For simplicity, we omitted the fact that we used a minibatch of data and augmented target data. More formally, a video V_i is divided into K clips $C_{i,1}$, ..., $C_{i,K}$ consisting of evenly spaced frames, which are fed to an encoder network $\phi(\cdot)$ that produces clip-level features $\boldsymbol{c}_{i,j} = \phi(C_{i,j})$. From here onward, the video indexes i are omitted for simplicity ($\boldsymbol{c}_j$ indicates $\boldsymbol{c}_{i,j}$). At this level, a self-supervised contrastive loss $\mathcal{L}_{\mathcal{C}_c}$ is applied to learn clip-level feature representations. Following [6], we apply a projection head before computing the contrastive loss. In practice, the clip-level features $\boldsymbol{c}_j$ are passed through a module that outputs $\boldsymbol{z}^{\boldsymbol{c}_j} = \omega_c(\boldsymbol{c}_j)$ to which the contrastive loss is applied.

To produce video representations, clip features $\boldsymbol{c}_j$ are aggregated into video-level feature $\boldsymbol{v} = \sum_{j=1}^{K} \alpha_j \boldsymbol{c}_j$, where the weight vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ is computed using a simple attention module $\kappa(\cdot)$, implemented as multi-layer perceptron (MLP) that receives K clip feature vectors as input, *i.e.* $\boldsymbol{\alpha} = \kappa(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K)$. Subsequently, $\boldsymbol{v}$ is fed to two separate network branches, with a similar base structure, implementing our two-headed architecture. In the first branch, the classification head $h_{\mathcal{CE}}(\cdot)$ produces logits $\boldsymbol{l} = h_{\mathcal{CE}}(\boldsymbol{v})$. In the second branch, $\boldsymbol{v}$ is first provided as input to the contrastive head $h_{\mathcal{CT}}(\cdot)$ to produce features $\bar{\boldsymbol{v}} = h_{\mathcal{CT}}(\boldsymbol{v})$, and then to a projection head ω_v, resulting in a latent vector $\boldsymbol{z}^{\boldsymbol{v}} = \omega_v(\bar{\boldsymbol{v}})$. The first head is trained using a cross-entropy loss $\mathcal{L}_{\mathcal{CE}}$, whereas the second head uses a combination of three contrastive losses: a supervised contrastive loss $\mathcal{L}_{\mathcal{SC}}$ [22] for source data, a self-supervised contrastive loss $\mathcal{L}_{\mathcal{C}_v}$ to learn better video-level representations for the target data and a class-aware inter-domain contrastive loss $\mathcal{L}_{\mathcal{IDC}}$ that aligns the distributions of the features of both domains. Finally, the stability loss $\mathcal{L}_{\mathcal{ST}}$ enforces an agreement between the predictions of the two main network heads $h_{\mathcal{CE}}(\cdot)$ and $h_{\mathcal{CT}}(\cdot)$.

Our network is trained on minibatches of size $3M$, *i.e.* composed of M^S randomly chosen source videos, M^T randomly chosen target videos and their M^T augmented versions. So we end up with M^S source videos and $2M^T$ target videos. In practice, data augmentations are only used for target data to train the unsupervised contrastive loss terms.

Next, we describe in detail the main components of our approach: (i) the proposed multi-scale feature representations, where we used self-supervised contrastive learning to compute both video-level and clip-level features; (ii) our novel contrastive domain alignment loss $\mathcal{L}_{\mathcal{IDC}}$ and (iii) our novel dual-head classifier structure.

Multi-scale Contrastive Video Feature Learning. Clip-level and video-level features carry different information about a video [32]. While the first ones focus on subparts of an action, the second ones are intended to represent the complete action. This work proposes to leverage the complementary clip-level and video-level information to learn representations on the target domain within a contrastive learning framework. Specifically, we resort to data augmentation on the target domain and, inspired by recent works on video representation learning [39], we define two self-supervised contrastive loss terms, one at video-level and the other at clip-level.

At the video-level, we use the output of the projection head ω_v, which produces the projected video-level features $\boldsymbol{z^v}$, and define the loss:

$$\mathcal{L}^i_{\mathcal{C}_v} = -\sum_{j=1}^{2M^T} \mathbb{1}_{\varphi_{i,j}} \cdot \log \frac{s^{\boldsymbol{z_i^v},\boldsymbol{z_j^v}}}{\sum_{p=1}^{2M^T} \mathbb{1}_{i\neq p} \cdot s^{\boldsymbol{z_i^v},\boldsymbol{z_p^v}}}. \qquad (1)$$

where $s^{\boldsymbol{z}_i,\boldsymbol{z}_j} = \exp(\frac{\boldsymbol{z}_i \cdot \boldsymbol{z}_j / \tau}{||\boldsymbol{z}_i|| \cdot ||\boldsymbol{z}_j||})$, $\tau > 0$ is a temperature parameter, $\mathbb{1}$ is an indicator function which is 1 if its argument is true or 0 otherwise, and $\varphi_{i,j}$ is true if i and j are two different augmentations of the same video. In practice, this loss has the effect of pulling together representations in the embedding space of augmented versions of the same target video (positive samples), while pushing away those associated with different videos in the same mini-batch (negative samples).

Similarly, at clip-level, we use the output of the projection head ω_c, which produces the projected clip-level features $\boldsymbol{z^{c_j}}$, and define the loss:

$$\mathcal{L}^i_{\mathcal{C}_c} = -\frac{1}{K} \sum_{j=1}^{2M^T} \sum_{u=1}^{K} \mathbb{1}_{\varphi_{i,j}} \log \frac{s^{\boldsymbol{z_i^{cu}},\boldsymbol{z_j^{cu}}}}{s^{\boldsymbol{z_i^{cu}},\boldsymbol{z_j^{cu}}} + \mathrm{Neg}_{i,u}}, \qquad (2)$$

where $\mathrm{Neg}_{i,u} = \sum_{p=1}^{2M^T} \mathbb{1}_{p\neq i, p\neq j} \sum_{v=1}^{K} s^{\boldsymbol{z_i^{cu}},\boldsymbol{z_p^{cv}}}$ is the set of negatives for instance i and clip u. In practice, this loss considers different augmentations of the same clip as positives and clips from different videos as negatives. Selecting different clips from the same video to form the set of negatives, in fact, could be harmful, since an action may span over multiple clips. The final self-supervised contrastive loss on target data is:

$$\mathcal{L}^i_{\mathcal{C}} = \mathcal{L}^i_{\mathcal{C}_c} + \mathcal{L}^i_{\mathcal{C}_v} \qquad (3)$$

It is worth noting that the two proposed losses are complementary since they use different notions of positives and negatives and operate at different temporal resolutions. Our experimental results (Sec. 5) demonstrate the benefit of our multi-scale self-supervised video representation learning strategy.

Contrastive Domain Alignment. To specifically address the domain shift problem and perform feature alignment of the source and target data, we introduce a novel inter-domain contrastive loss. Previous works on supervised contrastive learning [22] introduced a loss that has the effect of pulling together in the embedding space samples belonging to the same class while pushing far apart samples from different classes. In practice, in [22], positive and negative samples are obtained by only considering label information. In this work, we propose to revisit this idea by jointly combining samples from the two domains. However, while each source video V^S has an associated label Y^S, for the unsupervised target videos, we compute pseudo-labels $\tilde{Y}^T = \mathrm{argmax}\ \sigma(\boldsymbol{l}^T)$, where σ denotes the softmax operator. We employed a simple pseudo-labelling strategy, but other more complex strategies could be used. Therefore, we propose to consider as positives, instances from different domains that share the same label/pseudo-label, while instances with different labels/pseudo-labels and different domains are regarded as negatives. In this way, feature alignment among different domains is realised, while also taking into account semantic information. Formally, our proposed inter-domain contrastive loss is defined as:

$$\mathcal{L}^i_{\mathcal{IDC}} = -\frac{1}{\gamma_i} \sum_{j=1}^{3M} \mathbb{1}_{\rho_{i,j}} \cdot \mathbb{1}_{\tilde{Y}_i = \tilde{Y}_j} \cdot \log \frac{s^{\boldsymbol{z_i^v},\boldsymbol{z_j^v}}}{\sum_{p=1}^{3M} \mathbb{1}_{\rho_{i,p}} \cdot s^{\boldsymbol{z_i^v},\boldsymbol{z_p^v}}}, \qquad (4)$$

where $\mathbb{1}_{\rho_{a,b}} = \mathbb{1}_{a\notin\Omega} \mathbb{1}_{b\notin\Omega} \mathbb{1}_{D(a)\neq D(b)}$, γ_i is the number of positives for instance i and $D(\cdot)$ is a function that returns the domain of an instance. Ω denotes the set of target instances for which pseudo-labels are not considered reliable. Note that alternative losses such as those based on domain discrepancy minimisation and adversarial learning are designed to encourage the network to produce domain-agnostic features, whereas our proposed inter-domain contrastive loss encourages the network to produce tight representations and exploits label/pseudo-label information to push together features belonging to the same class and pull apart those belonging to different classes. To limit the effect of noise that is typically present in pseudo-labels, we propose to employ a simple sample filtering procedure: target instances are added to Ω when $H(\sigma(\boldsymbol{l}_i^T)) > log(n_{classes})/\eta$, where H is an entropy function and η a user-defined parameter that represents the percentage of the maximum allowed entropy.

Two-headed network. As discussed above, we design an architecture with two different heads. Each head is supervised with different losses. The first head, $h_{\mathcal{CE}}$, is mainly trained with a cross-entropy loss on source instances, *i.e.*:

$$\mathcal{L}^{i}_{\mathcal{CE}} = -\sum Y^{S}_{i} \log \sigma(\boldsymbol{l_i}). \tag{5}$$

Differently, the second head, $h_{\mathcal{CT}}$, is trained using the contrastive losses. Besides the previously described $\mathcal{L}^{i}_{\mathcal{C}_v}$ and $\mathcal{L}^{i}_{\mathcal{IDC}}$, we also introduce a supervised contrastive loss [22]:

$$\mathcal{L}^{i}_{\mathcal{SC}} = -\frac{1}{\gamma_i} \sum_{j=1}^{M} \mathbb{1}_{i \neq j} \cdot \mathbb{1}_{\tilde{Y}_i = \tilde{Y}_j} \cdot \log \frac{s^{\boldsymbol{z^v}_i, \boldsymbol{z^v}_j}}{\sum_{p=1}^{M^S} \mathbb{1}_{i \neq p} \cdot s^{\boldsymbol{z^v}_i, \boldsymbol{z^v}_p}}. \tag{6}$$

By leveraging from source supervision in a different way, the two heads promote the learning of different feature representations. To maximally benefit from this dual head structure, we introduce an additional loss term that enforces coherence between the predictions of the two heads. Besides stabilising the training of both heads, it makes additional information flow directly from the $h_{\mathcal{CT}}$ to $h_{\mathcal{CE}}$ and vice versa. As the contrastive head operates on instance pairs, we propose to define this coherence loss considering pairwise predictions associated to pairs of source and target videos in the minibatch, as follows:

$$\mathcal{L}^{i,j}_{\mathcal{ST}} = q_{i,j} \log(p_{i,j}) + (1 - q_{i,j}) \log(1 - p_{i,j}) \tag{7}$$

Here $p_{i,j} = \boldsymbol{l_i} \boldsymbol{l_j}^{\top}$ denotes the pairwise predictions computed on source video i and target video j using of the logits produced by $h_{\mathcal{CE}}$. Similarly, $q_{i,j}$ indicates the binary prediction label which is computed though $h_{\mathcal{CT}}$ as:

$$q_{i,j} = \begin{cases} 1, & \text{if } \cos(\bar{\boldsymbol{v}}_i, \bar{\boldsymbol{v}}_j) > \theta \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $\cos(\cdot, \cdot)$ indicates the cosine similarity between two vectors and θ is a threshold that we set equal to 0.5.

Overall Loss. The whole model is trained by combining the losses and weighting them accordingly as follows:

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} w_{ce} \mathcal{L}^{i}_{\mathcal{CE}} + w_{sc} \mathcal{L}^{i}_{\mathcal{SC}} + \frac{1}{\mu} \sum_{i=1}^{3M} w_{idc} \mathcal{L}^{i}_{\mathcal{IDC}} + \\ \frac{1}{2M} \sum_{i=1}^{2M} w_{c} \mathcal{L}^{i}_{\mathcal{C}} + \frac{1}{2M^2} \sum_{i=1}^{2M} \sum_{j=1}^{M} w_{st} \mathcal{L}^{i,j}_{\mathcal{ST}}, \tag{9}$$

where μ is the number of instances with at least one positive.

Inference. At inference time, the projection heads ω_c and ω_v, and $h_{\mathcal{CT}}$ are discarded. Data is only forwarded through the shared backbone and $h_{\mathcal{CE}}$, producing the logits $\boldsymbol{l}$ that are further normalised by a softmax function to generate the classes probabilities. It is worth noting that while at training time the addition of the double head implies an increase in terms of parameters, during inference, since the ω_c, ω_v and $h_{\mathcal{CT}}$ are discarded, the number of parameters is the same as if we used a single head architecture.

5. Experimental Results

5.1. Setup

Datasets. We conduct experiments on three standard UDA benchmarks for action recognition: *UCF↔HMDB* [5], *UCF↔Olympic Sports* [5], *Kinetics→NEC-Drone* [9], and on our newly proposed *Mixamo→Kinetics*. In *UCF↔HMDB* and *UCF↔Olympic Sports*, the domain shift is caused by varying visual appearance, lighting, camera viewpoint, etc. However, source and target domains are both associated with videos depicting real scenes. In the *Kinetics→NEC-Drone*, the domain shift is large as data of the source domain consist of Youtube videos, while the target domain comprises videos taken from a camera installed on a drone. Lastly, the *Mixamo→Kinetics* dataset presents the most severe domain shift, comprising synthetically generated video sequences in the source domain and Youtube videos in the target.

Baselines. We compare with three state-of-the-art UDA methods for action recognition: (i) TA^{3}N [5], considering both the original implementation of the 2D encoder (from [59]) and the adapted 3D version (similar to [10]) using the I3D [4] backbone; (ii) TCoN [35], considering the Resnet101 architecture as backbone, and (iii) SAVA [10], which employs I3D as clip feature extractor. We did not compare with [45] because their approach combines RGB with optical flow information, whereas ours solely uses RGB. Results are also reported for each backbone considering the following settings: (i) *supervised source only*, when the network is trained only with supervised source data, and (ii) *supervised target only*, when the network is trained (fine-tuned) with supervised target data. These settings correspond respectively to a lower and an upper bound for UDA methods. Note that while code for TA^{3}N [5] is publicly available, we did not find implementations for TCoN [35] and SAVA [10]. The associated results are taken from the original papers. Methods are compared in terms of *Top-1 Accuracy*.

Implementation details. We employ an I3D architecture as backbone network to be comparable with our closest competitor [10]. $\phi(\cdot)$ since . The I3D is pretrained on Kinetics for all datasets, except on the *Mixamo→Kinetics* where it is initialised by inflating the weights from an Imagenet-pretrained Inception-v1 network, as in [4], and fine-tuned on Mixamo labelled data. We implemented κ as MLP with architecture *Linear/ReLU/Linear/Sigmoid* that receives as input $K = 4$ clips, following [10]. The two heads $h_{\mathcal{CE}}(\cdot)$ and $h_{\mathcal{CT}}(\cdot)$ are both implemented as 2-layers MLPs with ReLU activation and without BatchNorm layers, with the only difference that a linear classifier is appended to $h_{\mathcal{CE}}(\cdot)$. Both take as input the video-level features, but $h_{\mathcal{CE}}(\cdot)$ outputs a vector of size equal to the number of classes and $h_{\mathcal{CT}}(\cdot)$ a vector of size 256. The projection heads ω_v and ω_c are both implemented as *Linear/ReLU/Linear* with output 128. The input space of ω_v is 256, whereas in ω_c it is

Table 2: Results on *UCF↔HMDB*

Method	Encoder	U→H	H→U
Supervised source only [5]	Resnet101-TRN	71.7	73.9
DANN [14]		75.2	76.3
JAN [29]		74.7	79.6
AdaBN [25]		72.2	77.4
MCD [41]		73.8	79.3
TA^{3}N [5]		78.3	81.8
Supervised target only [5]		82.8	94.9
Supervised source only [7]	I3D-based TRN	80.6	88.8
TA^{3}N [5]		81.4	90.5
Supervised target only [5]		93.1	97.0
Supervised source only [10]	I3D	80.3	88.8
SAVA [10]		82.2	*91.2*
Supervised target only [10]		95.0	96.8
TCoN [35]	2D/3D CNN	*87.2*	89.1
CO^{2}A	I3D	**87.8**	**95.8**

1024.

We perform video-based data augmentations on target data. Following [39], given a video, the same transformation is applied to all frames coherently. Colour, spatial and random horizontal flip augmentations are considered. Additional details about augmentations are reported in the supplementary material.

Hyper-parameters selection is performed following a common protocol in UDA literature [43], *i.e.* by selecting a subset (here, 5 annotated videos per class) in the target training set and by using them as validation set. On *HMDB↔UCF* and *UCF↔Olimpic*, the losses weights are $w_{ce} = 1$, $w_{sc} = 1$, $w_{idc} = 1.5$, $w_c = 0.2$ and $w_{st} = 0.02$; for *Kinetics→NEC-Drone* we set $w_{idc} = 0.2$, $w_c = 1.2$ and $w_{st} = 0.01$ and for *Mixamo→Kinetics* we set $w_{idc} = 2$, $w_c = 0.2$ and $w_{st} = 0.02$. All values were found using grid-search. We trained our network with SGD with learning rate 0.02, momentum of 0.9 and weight decay of 1^{-9}. We set $\eta = 6$ for *Mixamo→Kinetics* and $\eta = 4$ for the other experiments. Experiments were carried out on 4 Nvidia RTX 5000 GPUs for around 1 hour *HMDB↔UCF* and *UCF↔Olimpic*, 3 hours for *Kinetics→NEC-Drone* and 4 hours for *Mixamo→Kinetics*.

5.2. Results

Comparison with state of the art. We first report the results obtained comparing our approach with state-of-the-art methods. Tables 2, 3 and 4 show the results of our experiments on *HMDB↔UCF*, *UCF↔Olympic Sports* and *Kinetics→NEC-Drone*, respectively. In all tables, the best results are indicated in bold and the second-best in italic.

As shown in Table 2, our approach outperforms all previous methods for the *HMDB↔UCF* setting. In particular, it achieves an accuracy of 87.8% for U→H and 95.8% for H→U, outperforming its best competitor, SAVA [10] with the same I3D backbone, by 5% and almost 4%, respectively. Notably, all recent UDA methods specifically designed for action recognition, *i.e.* TA^{3}N [5], TCoN [35], SAVA [10] and our method, significantly outperform traditional image-based UDA approaches, *i.e.* DANN [14], JAN [29], AdaBN [25] and MCD [41].

Table 3: Results on *UCF↔Olympic Sports*

Method	U→OS	OS→U
W. Sultani *et al.* [46]	33.3	47.9
T. Xu *et al.* [55]	87.0	75.0
AMLS (GFK) [20]	84.6	86.4
AMLS (SA) [20]	83.9	86.0
DAAA [20]	91.6	89.9
TA^{3}N [5] (Resnet101-TRN)	*98.2*	92.9
TCoN [35] (Resnet101-TRN)	96.8	*96.7*
SAVA [10] (I3D)	98.1	*96.7*
CO^{2}A (I3D)	**100**	**97.5**

Similar observations can be made looking at results in Table 3. Our method outperforms the best competing methods, *i.e.* TA^{3}N [5] on *UCF→Olympic Sports* and TCoN [35] and SAVA [10] on *Olympic Sports→UCF*. Again, modern UDA methods for action recognition, , *i.e.* TA^{3}N [5], TCoN [35], SAVA [10] and CO^{2}A, are significantly more accurate than traditional techniques [20, 46, 55].

Finally, Table 4 reports the results obtained in the more challenging *Kinetics→NEC-Drone* setting. The gap in performance between *supervised source only* (lower bound) and *supervised target only* (upper bound) indicates a domain shift that is significantly more pronounced than that observed in the *HMDB↔UCF* and the *UCF↔Olympic Sports* datasets. Even in this challenging setting, our approach outperforms state of the art methods. In particular, the accuracy of CO^{2}A is 1.6% higher than its best competitor SAVA [10].

Results on the *Mixamo→Kinetics* dataset. Table 5 shows the results obtained on our newly proposed *Mixamo→Kinetics* dataset. This setting is much more challenging than previous ones, not only due to the large domain gap but also because it contains more action categories than *Kinetics→NEC-Drone* (14 classes versus 7) and previous benchmarks. For this dataset, we only consider baseline methods for which the code is publicly available, *i.e.* TA^{3}N [5]. Additionally, we run a previous image-based UDA approach, *i.e.* ADDA [49]. Due to the intrinsic difficulty of the *Mixamo→Kinetics* dataset, it is not surprising that all the methods achieve lower performance than in other settings. Still, our approach sets the state-of-the-art, outperforming its best competitor TA^{3}N [5].

The table also reports the performance of CO^{2}A and baselines considering a weakly supervised setting, *i.e.* assuming that annotations are available for 5 randomly selected target instances per class. As shown in the table, our method again outperforms the competitors. The table additionally reports, as upper bound, the score of the *supervised target only* method, which considers annotations available on the entire target training set. The large gap between the performance of UDA methods and the upper bound encour-

Table 4: Results on *Kinetics→NEC-Drone*.

Method	**Encoder**	**Top-1 acc**
Supervised source only [5]	ResNet-101-based TRN	15.8
TA^{3}N [5]	ResNet-101-based TRN	25.0
Supervised source only [5]	I3D-based TRN	15.8
TA^{3}N [5]	I3D-based TRN	28.1
Supervised source only [10]	I3D	17.2
DANN [14]	I3D	22.3
ADDA [49]	I3D	23.7
Choi et al. [9] (on val set)	I3D	15.1
SAVA [10]	I3D	*31.6*
Supervised target only	I3D	81.7
CO^{2}A	I3D	**33.2**

Table 5: Results on *Mixamo→Kinetics*.

Method	**Weak Supervision**	**Encoder**	**Top-1 acc**
Supervised source only		I3D	11.2
ADDA [49]		I3D	11.0
TA^{3}N [5]		Resnet101-TRN	7.0
TA^{3}N [5]		I3D-based TRN	10.0
ADDA [49]	✓	I3D	17.0
TA^{3}N [5]	✓	Resnet101-TRN	13.0
TA^{3}N [5]	✓	I3D-based TRN	19.1
Supervised target only	✓	I3D	79.3
CO^{2}A		I3D	**16.4**
CO^{2}A	✓	I3D	**20.1**

ages further research on this challenging synthetic-to-real UDA setting that we introduced with this paper.

Ablation Study. We also perform an ablation study to assess and empirically demonstrate the importance of our technical contributions. Table 6 reports the results of a set of experiments conducted to analyze the role of the distinct losses employed in our framework. The ablation experiments consider the *HMDB↔UCF* and *Kinetics→NEC-Drone* datasets and report results obtained by disabling one loss at the time. Looking at Table 6 the following observations can be made: (i) The scores obtained with the full model (last line of the table) show that the different losses are complementary and the model achieves the best results when combining all of them; (ii) the inter-domain loss $\mathcal{L}_{\mathcal{IDC}}$ is beneficial in all the settings, enabling to reduce the domain shift by promoting domain distribution alignment; (iii) the heads consistency loss $\mathcal{L}_{\mathcal{ST}}$ provides a significant benefit in term of performance in the most challenging setting, *i.e.* in the *Kinetics→NEC-Drone* setting: the accuracy drops by 6% when this loss is disabled; (iv) disabling the clip-level and video-level losses is also detrimental for performance, with different performance among datasets. This suggests that both video-level and clip-level information are important to describe action videos. Lastly, (vi) disabling the supervised contrastive loss greatly reduces the performance on *HMDB→UCF* and *Kinetics→NEC-Drone*, which is related to the fact that the second head is not performing on par on source data.

Figure 3 shows the results of a sensitivity analysis of our model concerning the weights associated to $\mathcal{L}_{\mathcal{IDC}}$ and $\mathcal{L}_{\mathcal{SC}}$.The sensitivity analysis for the weights of $\mathcal{L}_{\mathcal{C}_c}$, $\mathcal{L}_{\mathcal{C}_v}$ and $\mathcal{L}_{\mathcal{ST}}$ are provided in the supplementary material due to lack of space. We considered the *HMDB↔UCF* setting and we show that both losses are beneficial for the final score when the optimal values of their weights are set. Figure 3 (left) shows that a value of w_{idc} equal to zero corresponds to the worst performance since no domain distribution alignment takes place. Figure 3 (right) shows that a value of w_{sc} equal to zero, corresponding to no supervision provided on the contrastive head, is suboptimal and performance can be improved if supervision on both network heads is provided. Similarly, using a high weight for $\mathcal{L}_{\mathcal{SC}}$ implies relying too strongly on source data and losing the benefit of the losses applied to the target data. Additional results are provided in the supplementary material, due to lack of space.

Table 6: Ablation study on *HMDB↔UCF* and *Kinetics→NEC-Drone*: importance of different losses.

Method	**H→U**	**U→H**	**K→N-D.**
CO^{2}A w/o $\mathcal{L}_{\mathcal{IDC}}$	92.5	87.5	30.9
CO^{2}A w/o $\mathcal{L}_{\mathcal{ST}}$	94.4	82.4	27.0
CO^{2}A w/o $\mathcal{L}_{\mathcal{C}_c}$	91.9	85.5	29.6
CO^{2}A w/o $\mathcal{L}_{\mathcal{C}_v}$	**95.8**	81.5	24.8
CO^{2}A w/o $\mathcal{L}_{\mathcal{SC}}$	91.5	86.9	28.1
CO^{2}A (full)	**95.8**	**87.8**	**33.2**

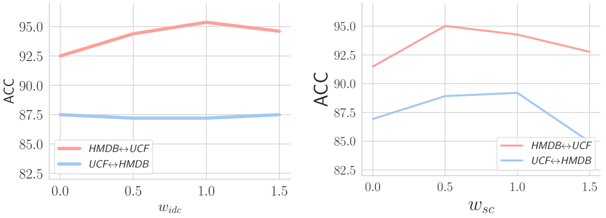

Figure 3: Sensitivity analysis of the weights of the losses $\mathcal{L}_{\mathcal{IDC}}$ and $\mathcal{L}_{\mathcal{SC}}$.

6. Conclusions

We presented CO^{2}A, a novel UDA approach for video action recognition that explores conditional feature alignment across domains within a contrastive learning framework. Our approach achieved state-of-the-art performance on three publicly available UDA benchmarks. Moreover, we introduced the novel large-scale dataset *Mixamo→Kinetics*. This new benchmark will foster future research in domain adaptation from synthetic to real video sequences. In the future, we plan to improve our approach by integrating more sophisticated methods for obtaining reliable pseudo-labels. Additionally, we plan to extend *Mixamo* to include videos generated with random light source position and strength and with moving cameras.

Acknowledgements: This work was supported by the Huawei Technologies project BONSAI, the EU H2020 AI4Media No. 951911 project, the Italy-Israel collaboration project AI4CH, the EUREGIO project OLIVER and the National Council for Scientific and Technological Development (CNPq), Brazil.

References

[1] Silvia Bucci, Antonio D'Innocente, and Tatiana Tommasi. Tackling partial domain adaptation with self-supervision. In *International Conference on Image Analysis and Processing*. Springer, 2019.

[2] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. AutoDIAL: Automatic Domain Alignment Layers. In *ICCV*, 2017.

[3] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *CoRR*, abs/1907.06987, 2019.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[5] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *ICCV*, 2019.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[7] Yu Chen, Chunhua Shen, Hao Chen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial Learning of Structure-Aware Fully Convolutional Networks for Landmark Localization. *IEEE T-PAMI*, 2019.

[8] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No More Discrimination: Cross City Adaptation of Road Scene Segmenters. In *ICCV*, 2017.

[9] Jinwoo Choi, Gaurav Sharma, Manmohan Chandraker, and Jia-Bin Huang. Unsupervised and semi-supervised domain adaptation for action recognition from drones. In *WACV*, 2020.

[10] Jinwoo Choi, Gaurav Sharma, S. Schulter, and J. Huang. Shuffle and attend: Video domain adaptation. In *ECCV*, 2020.

[11] Gabriela Csurka. *Domain adaptation in computer vision applications*, volume 2. Springer, 2017.

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

[13] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[14] Yaroslav Ganin and Victor Lempitsky. Unsupervised Domain Adaptation by Backpropagation. *ICML*, 2015.

[15] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *ICML*, 2018.

[18] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. *arXiv:1612.02649*, 2016.

[19] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional Generative Adversarial Network for Structured Domain Adaptation. In *CVPR*, 2018.

[20] Arshad Jamal, Vinay P Namboodiri, Dipti Deodhare, and KS Venkatesh. Deep domain adaptation in action space. In *BMVC*, 2018.

[21] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *CVPR*, 2019.

[22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

[23] Donghyun Kim, Kuniaki Saito, Tae-Hyun Oh, Bryan A Plummer, Stan Sclaroff, and Kate Saenko. Cross-domain self-supervised learning for domain adaptation with few source labels. *arXiv preprint arXiv:2003.08264*, 2020.

[24] W. Li, Fuyu Li, Yongkang Luo, and Peng Wang. Deep domain adaptive object detection: a survey. *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1808–1813, 2020.

[25] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *ICLR Workshop*, 2017.

[26] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[27] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional Adversarial Domain Adaptation. In *Proc. NeurIPS*, 2018.

[28] Mingsheng Long and Jianmin Wang. Learning Transferable Features with Deep Adaptation Networks. In *ICML*, 2015.

[29] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

[30] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. In *CVPR*, 2019.

[31] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Boosting Domain Adaptation by Discovering Latent Domains. In *CVPR*, 2018.

[32] Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Alex Andonian, Tom Yan, Kandan Ramakrishnan, Lisa M. Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding. *CoRR*, abs/1801.03150, 2018.

[33] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, 2017.

[34] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *CVPR*, 2020.

[35] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In *AAAI*, 2020.

[36] Changhwa Park, Jonghyun Lee, Jaeyoon Yoo, Minhoe Hur, and Sungroh Yoon. Joint contrastive learning for unsupervised domain adaptation. *arXiv preprint arXiv:2006.10297*, 2020.

[37] Mandela Patrick, Yuki M Asano, Polina Kuznetsova, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020.

[38] Albert Pumarola, Jordi Sanchez, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In *ICCV*, 2019.

[39] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020.

[40] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulò, Nicu Sebe, and Elisa Ricci. Unsupervised Domain Adaptation using Feature-Whitening and Consensus Loss. In *CVPR*, 2019.

[41] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[42] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.

[43] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[44] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018.

[45] Xiaolin Song, Sicheng Zhao, Jingyu Yang, Huanjing Yue, Pengfei Xu, Runbo Hu, and Hua Chai. Spatio-temporal contrastive domain adaptation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9787–9795, June 2021.

[46] Waqas Sultani and Imran Saleemi. Human action recognition across datasets by foreground-weighted histogram decomposition. In *CVPR*, 2014.

[47] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

[48] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[49] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[50] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[51] Gül Varol, Ivan Laptev, Cordelia Schmid, and Andrew Zisserman. Synthetic humans for action recognition from unseen viewpoints. *CoRR*, abs/1912.04070, 2019.

[52] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[53] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *CVPR*, 2018.

[54] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. *ECCV*, 2020.

[55] Tiantian Xu, Fan Zhu, Edward K Wong, and Yi Fang. Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition. *Image and Vision Computing*, 55, 2016.

[56] Xitong Yang, Xiaodong Yang, Sifei Liu, Deqing Sun, Larry Davis, and Jan Kautz. Hierarchical contrastive motion learning for video action recognition. *arXiv preprint arXiv:2007.10321*, 2020.

[57] Yang Zhang, Philip David, and Boqing Gong. Curriculum Domain Adaptation for Semantic Segmentation of Urban Scenes. In *ICCV*, 2017.

[58] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.

[59] Xingyi Zhou, Arjun Karpur, Chuang Gan, Linjie Luo, and Qixing Huang. Unsupervised Domain Adaptation for 3D Keypoint Estimation via View Consistency. In *ECCV*, 2018.