Pattern Recognition 161 (2025) 111328

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Source-free video domain adaptation by learning from noisy labels

Avijit Dasgupta [a],*, C.V. Jawahar [a], Karteek Alahari [b]

[a] CVIT, IIIT Hyderabad, India
[b] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, Grenoble, France

ARTICLE INFO

Keywords:
Action recognition
Domain adaptation
Transfer learning
And self-training

ABSTRACT

Despite the progress seen in classification methods, current approaches for handling videos with distribution shifts in source and target domains remain source-dependent as they require access to the source data during the adaptation stage. In this paper, we present a self-training based *source-free* video domain adaptation approach to address this challenge by bridging the gap between the source and the target domains. We use the source pre-trained model to generate pseudo-labels for the target domain samples, which are inevitably noisy. Thus, we treat the problem of source-free video domain adaptation as learning from noisy labels and argue that the samples with correct pseudo-labels can help us in adaptation. To this end, we leverage the cross-entropy loss as an indicator of the correctness of the pseudo-labels and use the resulting small-loss samples from the target domain for fine-tuning the model. We further enhance the adaptation performance by implementing a teacher–student (TS) framework, in which the teacher, which is updated gradually, produces reliable pseudo-labels. Meanwhile, the student undergoes fine-tuning on the target domain videos using these generated pseudo-labels to improve its performance. Extensive experimental evaluations show that our methods, termed as *CleanAdapt, CleanAdapt + TS*, achieve state-of-the-art results, outperforming the existing approaches on various open datasets. Our source code is publicly available at https://avijit9.github.io/CleanAdapt.

1. Introduction

The availability of large-scale action recognition datasets, coupled with the rise of deep neural networks, have significantly advanced the field of video understanding [1]. Similar to other machine learning models, these action recognition models often encounter new domains with *distribution-shift* when deployed in real-world scenarios where the data distribution of training (source domain) and test (target domain) data is different, resulting in degraded performance. A trivial solution to alleviate this problem is fine-tuning the models with *labeled* target domain data, which is not always feasible due to expensive target domain annotations. Unsupervised domain adaptation (UDA) tackles this problem by transferring knowledge from the labeled source domain data to the *unlabeled* target domain, thus eliminating the need for comprehensive annotations for the target domain [2]. Source-free UDA takes this approach one step further, where we assume that the source domain data is unavailable during the adaptation stage. This setting is more realistic than the source-dependent one primarily due to (a) privacy concerns, it is not always possible to transfer data between the vendor (source) and the client (target), (b) storage constraints to transfer the source data to the client side (e.g., Sports-1M is about 1 TB), and (c) source-free models reduce computation time and thus cost by not using the source domain data during the adaptation stage.

There has been a recent surge of interest in source-dependent unsupervised domain adaptation for videos [3–5]. These approaches either propose to directly extend the adversarial learning framework [6] from image-based methods [2] or couple it with some temporal attention weights [7,8] and self-supervised pretext tasks [8,9] to align the segment-level features between the domains. However, these strategies produce only a modest $\sim$ 2% gain over the source-only model (see Fig. 1). Recently, there has been a paradigm shift from adversarial to contrastive learning framework [3,4,10] for video domain adaptation. As shown in Fig. 1, CoMix [10] achieves 6.4% and 5.1% gain over the source-only model on the UCF $\rightarrow$ HMDB and HMDB $\rightarrow$ UCF datasets, respectively. However, all these methods are inherently complex and use source domain videos during the adaptation stage, which is untenable in several scenarios [11], as discussed earlier. Due to its practical relevance, source-free domain adaptation is a well-known problem for different computer vision tasks such as image classification [11], and semantic segmentation [12] but relatively under-explored in the context of the video classification task. Therefore, there is a need to investigate source-free domain adaptation for video classification tasks in order to improve the practicality and efficiency of today's approaches.

* Corresponding author.
E-mail addresses: avijit.dasgupta@research.iiit.ac.in (A. Dasgupta), jawahar@iiit.ac.in (C.V. Jawahar), karteek.alahari@inria.fr (K. Alahari).

https://doi.org/10.1016/j.patcog.2024.111328
Received 27 April 2024; Received in revised form 4 October 2024; Accepted 26 December 2024
Available online 2 January 2025

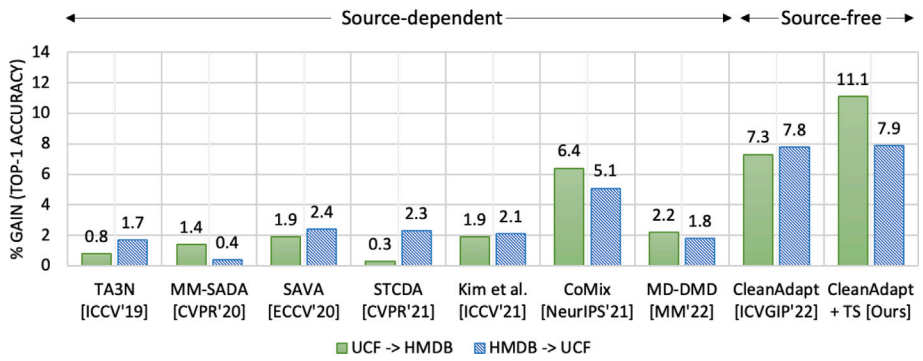


Fig. 1. Existing approaches have a *source-dependent* adaptation stage achieving marginal performance gain over the source-pretrained models. On the other hand, our proposed methods CleanAdapt and CleanAdapt + TS achieve significant performance improvements over the source-only model while being *source-free* (*i.e.*, the adaptation stage does not require videos from the source domain). (Best viewed in color.).

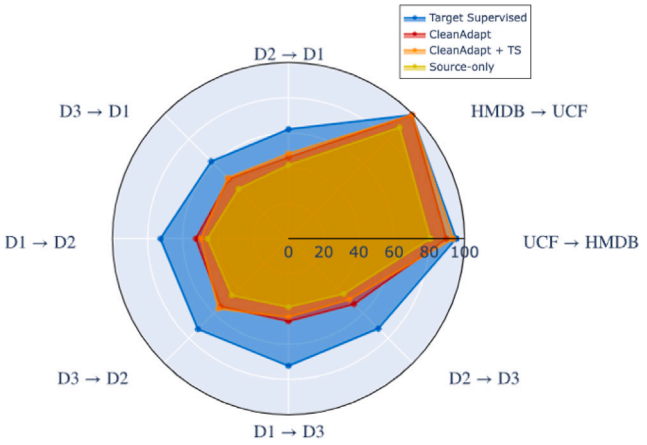


Fig. 2. The radar plot illustrates the performance improvements of our proposed methods, CleanAdapt and CleanAdapt + TS (shown in orange and red, respectively), compared to the source-only model on multiple benchmarks. The source-only model (shown in yellow), trained on the source domain and tested on the target domain, serves as the lower bound of adaptation performance, while the target-supervised model (shown in blue), trained and tested on target domain videos, represents the upper bound. (Best viewed in color.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In this work, we present an effective approach that leverages the self-training framework for source-free video UDA, where we do not have access to source-domain videos during the adaptation stage. We generate *pseudo-labels* for the unlabeled target domain videos using a source pre-trained model. These pseudo-labels are indeed noisy due to the existing domain gap. Finetuning the source pre-trained model with these noisy pseudo-labels is a sub-optimal solution as the presence of incorrect pseudo-labels hinders the adaptation stage, as discussed in Section 4.4. However, we observe that these pseudo-labeled target domain videos are not completely unusable, and in fact, there is a substantial number of target domain videos with correct pseudo-labels. For example, in the case of HMDB → UCF, the HMDB pre-trained model produces pseudo-labels with ∼ 90% accuracy on the UCF dataset, and we experimentally show that this amount of data is sufficient for adaptation. Throughout this paper, we refer to these samples with correct pseudo-labels as *clean*, whereas the samples with incorrect pseudo-labels are termed *noisy*.

We treat the problem of source-free domain adaptation as learning from noisy labels and propose a self-training based approach that selects clean samples from the noisy pseudo-labeled target domain samples to re-train the model for gradually adapting to the target domain in an iterative manner. We observe that deep neural networks tend to learn from the clean samples first before memorizing the noisy samples in the later stage of training according to the deep memorization effect [13]. In Fig. 3, we validate this effect empirically for both appearance (RGB) and motion (flow) modalities. From Fig. 3, it is clear that the model produces small-loss for the clean samples. In contrast to that, the noisy samples have high loss values for both the modalities. In our paper, we exploit this connection between small-loss and clean instances and propose an approach for source-free video domain adaptation. We discuss this in detail in Section 3.3. This method progressively adapts the model to the target domain in an iterative fashion. Thus, we name our approach as *CleanAdapt*. Additionally, we employ a teacher–student network [14] to produce more resilient pseudo-labels, where the teacher network is continuously updated by incorporating the temporal ensemble of student networks. This approach generates more consistent pseudo-labels, thus aiding the enhancement of the student network. We refer to this version of our method as *CleanAdapt + TS*. Our proposed methods surpass all other source-dependent state-of-the-art methods by a large margin on UCF ↔ HMDB and EPIC-Kitchens datasets, despite being source-free (see Figs. 1 and 2).

An earlier version of our approach was published as a conference paper [15]. This manuscript proposes the following significant improvements.

1. We comprehensively survey existing video domain adaptation approaches for different domain adaptation setups in Section 2.
2. We propose a strategy to filter out the incorrect pseudo-labeled (noisy) samples based on the deep memorization effect [13] and utilize the clean samples in the adaptation stage.
3. In Section 3, we propose an improved version of our pseudo-label generation process using a teacher–student framework. This modification yields reliable pseudo-label generation, thereby enhancing the overall performance.
4. We conduct a comprehensive analysis of the results obtained with the popular UCF ↔ HMDB and EPIC-Kitchens datasets and detailed in Section 4.
5. We also include evaluations and comparisons in Section 4 with recently developed source-free video domain adaptation techniques like [16–18]. Our CleanAdapt and CleanAdapt + TS outperform all the existing video domain adaptation approaches.

The paper is organized as follows. In Section 2, we review some of the notable and recent works in this domain. Section 3 describes the CleanAdapt and CleanAdapt + TS methods. In Section 4, we present the experimental analyses and finally conclude in Section 5.

2. Related work

Supervised action recognition. Convolutional neural networks (CNNs) are now the de facto solution for action recognition tasks. Various efforts have been made in this context to capture spatio-temporal

Table 1
Summary of domain adaptation for action recognition methods. "N/A" denotes the unavailability of source code.

Methods	Venue	Code Link	Backbone	Core Components	Types of Videos
Unsupervised Video Domain Adaptation					
AMLS [6]	BMVC'18	N/A	C3D	Subspace alignment	Third person
DAAAA [6]	BMVC'18	N/A	C3D	Domain invariant feature learning; Adversarial learning with domain discriminator	Third person
TA3N [7]	ICCV'19	PyTorch	ResNet-101	Attention alignment; Temporal discrepancy; Adversarial alignment	Third person
TCoN [20]	AAAI'20	N/A	BN-Inception, C3D	Cross-domain co-attention; Adversarial learning for temporal adaptation	Third person
MM-SADA [9]	CVPR'20	Tensorflow	I3D	Self-supervised cross-modal alignment; Adversarial Learning	First person
Choi *et al.* [21]	WACV'20	N/A	I3D	Adversarial learning	Third-person
SAVA [8]	ECCV'20	N/A	I3D	Align important clips; Self-supervised clip-order prediction	Third-person
STCDA [4]	CVPR'21	N/A	BN-Inception, I3D	Spatio-temporal contrastive learning; pseudo-labeling	First and Third person
Kim *et al.* [22]	ICCV'21	N/A	I3D	Contrastive learning; cross-modal and cross-domain alignment	First and Third person
CoMix [10]	NeurIPS'21	PyTorch	I3D	Temporal contrastive learning; background mixing; pseudo-labeling	First and Third person
CO2A [23]	WACV'22	PyTorch	I3D	Dual head contrastive network; Synthetic data	Third person
MA^{2}LTD [24]	WACV'22	PyTorch	ResNet-101	Multi-level temporal features; Multiple domain discriminators	Third person
CIA [25]	CVPR'22	N/A	I3D	Cross-modal complementarity and consensus	First and Third person
TranSVAE [26]	NeurIPS'23	PyTorch	I3D	Disentanglement of domain-related and semantic-related information	First and Third person
MD-DMD *et al.* [17]	MM'22	N/A	I3D	Dynamic modal distillation	First and Third person
Broome *et al.* [27]	WACV'23	N/A	SlowFast (video), Resnet-18 (audio)	Audio-adaptive encoder	First and Third person
CTAN [5]	TCSVT'23	PyTorch	I3D	Channel-temporal attention network	First person
Source-free Video Domain Adaptation					
CleanAdapt [15]	ICVGIP'22	PyTorch	I3D	Pseudo-labeling; Learning from noisy labels	First and Third person
ATCoN [16]	ECCV'22	PyTorch	ResNet-50	Temporal consistency network	Third person
MTRAN [18]	MM'22	N/A	I3D, Transformer	Temporal relative alignment; Mix-up	First and Third person

information in videos, starting from two-stream networks with 2D [19] to 3D CNNs [1]. However, a common limitation of existing methods is their dependence on training data that closely matches the distribution of the test data. When there is a subtle difference in the distribution between the training and the test domains, these models struggle to generalize effectively. Consequently, fine-tuning with a large amount of labeled data from the target domain is often required, which can be both time-consuming and expensive. To address this issue, our focus is on unsupervised video domain adaptation, aiming to overcome the need for labeled target domain data.

Domain adaptation for action recognition. Early works [7–9] on video UDA are inspired by the adversarial framework [2] for image-based UDA tasks. Jamal *et al.* [6] proposes to align the source and the target domains using a subspace alignment technique and outperform all the previous shallow methods. Chen *et al.* [7] show the efficacy of attending to the temporal dynamics of video for domain adaptation. TCoN [20] is a cross-domain co-attention module for matching the source and the target domain features with appearance and motion streams. Munro *et al.* [9] were among the first to show the effectiveness of learning multi-modal correspondence for video domain adaptation. SAVA [8] is an attention-augmented model with a clip order prediction task to re-validate the effectiveness of self-supervised learning for video domain adaptation, as shown in [9]. Overall, the adversarial methods are complex and sensitive to the choice of hyperparameters [10].

There has been a recent shift from adversarial to contrastive learning-based methods for the video UDA task. Song *et al.* [4] propose to bridge the domain gap using a self-supervised contrastive framework named cross-modal alignment. In a similar direction, Kim *et al.* [3] use a cross-modal feature alignment loss for learning a domain adaptive feature representation. CoMix [10] represents videos as graphs and uses temporal-contrastive learning over graph representations for transferable feature learning. Additionally, these methods [3,4,10] generate pseudo-labels from the source pre-trained model for the target domain videos and use only the target domain samples with high-confident pseudo-labels in their contrastive loss in each iteration. However, the source-only model often makes incorrect predictions with high confidence due to the distribution shift for target domain videos, which can hinder adaptation. To address this, we treat target pseudo-labels as noisy and formulate the domain adaptation problem as learning from noisy labels. Moreover, the adaptation stage in these methods [3,4,10] is *source-dependent*. This is an impractical requirement as source data transfer during the deployment phase of the model is often infeasible.

Recently, ATCoN [16] and our conference paper CleanAdapt [15] have addressed this issue of source data dependency. These methods introduce a source-free adaptation approach, *i.e.*, it does not rely on source domain videos during the adaptation stage. In Table 1, we provide an overview of existing methods for unsupervised and source-free video domain adaptation. Xu *et al.* [28] provide an extensive survey of video domain adaptation, encompassing a variety of setups.

Learning from noisy-labels. Self-training based methods with careful design choices may still produce over-confident, incorrect predictions. To alleviate this issue, we resort to learning from label-noise literature. One of the popular approaches to reducing the effect of noisy-labels is to design noise-robust losses [29]. However, these methods fail to handle real-world noise [30]. According to [13], deep neural networks produce small loss values for samples with correct pseudo-labels. Thus, a popular direction for handling label-noise is to use the cross-entropy loss to indicate label correctness [31] and leverage these small-loss samples for re-training the networks. In this work, we demonstrate that the small-loss samples are potentially clean samples and are effective in helping our source pre-trained model adapt to the target domain if these samples are used for fine-tuning. Therefore, our proposed approach is simpler that the existing approaches, requiring solely pseudo-labeled samples from the target domain.

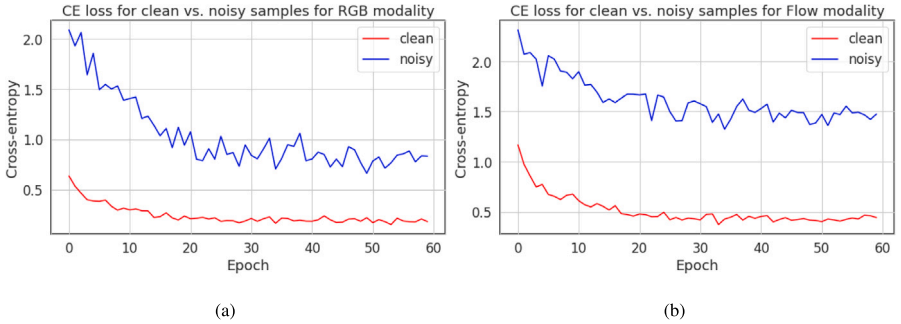


Fig. 3. Average cross-entropy loss per epoch of training with pseudo-labeled target domain videos for clean vs. noisy samples with (a) RGB modality and (b) Flow modality. We term the target domain samples with correct pseudo-labels as *clean* samples and with incorrect pseudo-labels as *noisy* samples. Note that, the groundtruth labels are only used to identify the clean vs. the noisy samples for visualization purposes and not used for training the model. Deep neural networks learn the clean samples first before memorizing the noisy samples according to the deep memorization effect as proposed in [13]. In our proposed approach CleanAdapt, we exploit this connection to select the clean samples for fine-tuning the model to adapt to the target domain. (Best viewed in color.)

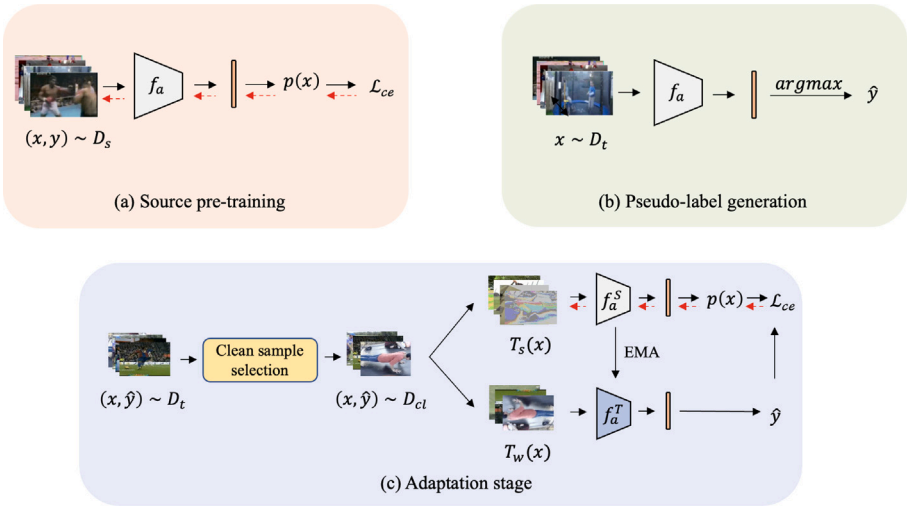


Fig. 4. Overview of the three stages of our CleanAdapt + TS framework for source-free video domain adaptation, which has three stages. (a) The model (f_a) is first pre-trained on the *labeled* source domain videos from D_s. For brevity, only the single-stream model is shown here. (b) This source pre-trained model is then used to generate pseudo-labels $\hat{y}$ for the *unlabeled* target domain videos from D_t. Inevitably, these pseudo-labels are noisy due to the domain shift between the source and the target domains. (c) A *clean sample selection* module is used to select a set D_{cl} of small-loss samples as potential clean samples. The source pre-trained model is finetuned on these clean samples from D_{cl} using their corresponding pseudo-labels $\hat{y}$. We repeat this step multiple times. See Section 4 for implementation details. (Best viewed in color.).

3. Approach

3.1. Problem definition

In the *source-free* UDA task for videos, we are given a labeled source domain dataset of videos $D_s = \{(x_s, y_s) : x_s \sim P\}$, where P is the source data distribution and y_s is the corresponding label of x_s. We are also given an unlabeled target domain dataset $D_t = \{x_t : x_t \sim Q\}$, where Q is the target distribution that is different from the source distribution P. We assume that the source and the target domains share the same label-set C, *i.e.*, closed-set domain setup.

For a video clip x from any domain, we consider two modalities, $x = \{x_a, x_m\}$, where x_a is the appearance (RGB) stream and x_m is the motion (optical flow) stream. We use two 3D CNN backbones f_a and f_m, one for each modality that classifies a video into one of the $|C|$ classes. We aim to adapt the 3D CNNs (f_a and f_m) to the target domain. We also note that the source domain videos are only available during the pre-training stage and we do not use this dataset D_s during the adaptation stage as we are interested in the more realistic source-free setup. We show an overview of the proposed method in Fig. 4.

3.2. Self-training based domain adaptation

In contrast to the adversarial learning based approaches [7–9], we take the path of self-training primarily due to its simplicity in the adaptation stage. First, we pre-train the 3D CNN models using the labeled source videos from D_s. Second, we generate pseudo-labels for the unlabeled target dataset D_t using the source pre-trained model referred to as pseudo-labels. Third, we retrain the networks f_a and f_m using the pseudo-labeled target domain videos from D_t for adaptation. One of the possibilities is to use all the samples with their corresponding pseudo-labels to retrain the networks. However, pseudo-labels are noisy due to the domain gap between the source domain D_s and the target domain D_t. Retraining f_a and f_m with all these pseudo-labeled samples from D_t leads to a sub-optimal result, as discussed in Section 4. We aim to answer the following question in this paper: How do we choose the pseudo-labeled samples from D_t that can help the model in adaptation?

3.3. Clean samples are all you need

The pseudo-labels contain a large number of samples with correct pseudo-labels (clean samples). For example, there are ~90% samples with correct pseudo-labels in the UCF dataset when generated using the HMDB pre-trained networks. Thus, if we can filter out the noisy

Fig. 5. The *clean sample selection* module. The pseudo-labeled target domain videos from D_t are grouped according to their pseudo-labels $\hat{y}$ and sorted in ascending order of the loss generated by the model against their pseudo-labels. The *keep-rate* τ ($\tau = 0.6$ in this example) decides the number of samples to be selected for adaptation, having small-loss values for each class. For simplicity, we have used only four classes here. We show the videos with the correct pseudo-labels inside green border, whereas the videos with incorrect pseudo-labels are inside the red border solely for visualization purposes. (Best viewed in color.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

samples and keep only the clean ones, we can finetune our networks (f_a and f_m) using these clean samples and their corresponding pseudo-labels. Thus, we argue that these clean samples are the ones that can help us in domain adaptation. Now, the important question here is how to separate the clean samples from the noisy ones.

In Fig. 3, we observe that deep neural networks learn from clean samples easily and have a hard time learning from noisy samples due to the memorization effect [13]. Thus, samples with *low-loss* values are the potential clean samples and can be filtered out using the loss as an indicator. In this work, we design two approaches without bells and whistles, *CleanAdapt* and *CleanAdapt + TS*, aiming to select the clean samples based on the loss generated by the model against their corresponding pseudo-labels for adaptation. In each epoch of the adaptation stage, we select these clean samples from the target domain and use them to re-train the source pre-trained models f_a and f_m.

There are three key advantages to this: (1) we do not need to modify the overall training regime (*e.g.*, contrastive learning for domain alignment [3,4,10]) during adaptation, (2) we do not need to make any domain adaptation specific design choices (*e.g.*, background mixing [10]), and (3) we implicitly design an adaptation method that *does not* need any source data during the adaptation stage (refer Fig. 4).

3.4. Source pre-training

In the source pre-training stage, we train the 3D CNNs f_a and f_m using the labeled source domain dataset D_s, and we refer to these as *source-only* models. For a sample $(x, y) \in D_s$, we average the logits obtained from $f_a(x)$ and $f_m(x)$ to compute the final score $p(x)$ as follows:

$$p(x) = \sigma(f_a(x) + f_m(x)). \tag{1}$$

We use the conventional cross-entropy loss between the predicted class probabilities $p(x)$ and the one-hot encoded ground-truth label y as the loss function for training:

$$\mathcal{L}_{ce}(x) = -\sum_{c=1}^{|C|} y^c \log(p^c(x)), \tag{2}$$

where y^c and p^c represent the cth element of y and $p(x)$ respectively for class c. The main goal for this pre-training step is to equip our model with the initial knowledge of the classes present in the source dataset D_s. Fig. 4(a) depicts this step.

3.5. Pseudo-label generation

The next step, as illustrated in Fig. 4(b), is to generate the pseudo-labels for the unlabeled target domain samples. Once the model is pre-trained on the source domain videos, we use the learned notion of the class semantics of the model to generate labels for the target domain data. Note that these generated labels are not the actual labels for the target domain videos. Thus, we term these source-only model-generated labels as *pseudo-labels* $\hat{y}$. Formally,

$$\hat{y}(x) = \arg\max_c p^c(x), \tag{3}$$

where $x \in D_t$. Due to the domain shift between the source and the target, these pseudo-labels $\hat{y}$ are noisy.

3.6. CleanAdapt + TS : A strong video adaptation method

Once the pseudo-labels are obtained for the target domain videos, we use them for adaptation, as shown in Fig. 4(c). As discussed earlier, the pseudo-labels are noisy, and we aim to extract samples with the correct pseudo-labels (clean samples) for adaptation. Each epoch of the adaptation stage has two key steps in our framework: (a) clean sample selection and (b) fine-tuning the models f_a and f_m using these clean samples.

Clean sample selection. To filter out the target domain videos with noisy pseudo-labels, we start with the pseudo-labels generated in Section 3.5 and exploit the relation between the small-loss and the clean samples. We use the source pre-trained models (f_a, f_m) to select the clean samples reliably. In each epoch, the videos are first grouped into $|C|$ classes based on their pseudo-labels generated by the model and sorted in ascending order of their cross-entropy loss values computed using the *prediction* made by the models ($p(x) = \sigma(f_a(x) + f_m(x))$) and their corresponding *pseudo-labels* ($\hat{y}$):

$$D_{cl}, D_{no} \leftarrow \mathcal{L}_{ce}(p(x), \hat{y}(x)), \tag{4}$$

where $\sigma(.)$, D_{cl}, and D_{no} represent the softmax function, sets of clean samples and noisy samples, respectively. If the pseudo-labels are correct, the model is likely to produce a small loss, and thus, there is a high possibility that the sample belongs to D_{cl}. Inspired by [31], we define a hyper-parameter *keep-rate* τ. For each group, we select τ proportion of the total number of samples with small losses (See Fig. 5). We call this updated dataset of small-loss samples as $D_{cl} \subset D_t$ and discard the samples in D_{no}. We update the pseudo-labels as follows:

$$\hat{y}(x) \leftarrow \arg\max_c p^c(x). \tag{5}$$

This version of our proposed model is referred to as CleanAdapt.

Inspired by the success of the teacher–student framework [14,32], we adopt it along with our small-loss based clean sample selection to combat the label-noise, and we call this model as *CleanAdapt + TS*. The teacher and student networks share the same architecture and are initialized with the source pre-trained weights. We use temporally averaged teacher models to select reliable pseudo-labeled target domain videos. To accomplish this, we generate two copies of the models f_a and f_m: one operates as the teacher model (f_a^T, f_m^T), while the other functions as the student model (f_a^S, f_m^S) in their respective modalities. The parameters of the teacher models (θ_a^T and θ_m^T) are not updated through loss back-propagation, rather they are an exponential moving average of student parameters θ_a^S and θ_m^S respectively. To this end, we create two different versions of the same video x in each modality using two transformations. Let $T_w(x)$ and $T_s(x)$ denote the weakly and strongly augmented versions of the same video $x \in D_t$ as defined below:

Weak augmentations. We refer to common geometric transformations like flipping and shifting as weak augmentations. In particular, we incorporate a random horizontal flip, applied universally to the video, with a 50% probability.

Strong augmentations. To achieve robust augmentation, we implement RandAugment [33] for each video x. This method randomly selects three augmentations from a predefined list and applies them to the video x.

The teacher networks, denoted as f_a^T and f_m^T, use the weakly augmented version of the video x to produce pseudo-labels. In contrast, the student networks, represented by f_a^S and f_m^S, undergo fine-tuning using the strongly augmented version of the video x. This fine-tuning process enhances their resilience to noise.

As for CleanAdapt, here also the samples are divided into the clean and the noisy sets as follows:

$$D_{cl}, D_{no} \leftarrow \mathcal{L}_{ce}(p^T(x), \hat{y}(x)), \tag{6}$$

where $p^T(x) = \sigma(f_a^T(T_w(x)) + f_m^T(T_w(x)))$. We also update the pseudo-labels using the teacher model, as shown below.

$$\hat{y}(x) \leftarrow \arg\max_c p^{T,c}(T_w(x)), \tag{7}$$

where $P^{T,c}$ denotes the probability of the sample x belonging to class c as predicted by the teacher model.

Fine-tuning. In this step, the student networks f_a^S and f_m^S are re-trained using the strongly augmented samples $T_s(x)$ and their corresponding pseudo-label $\hat{y(x)}$ from D_{cl} using the cross-entropy loss as shown below.

$$\mathcal{L}_{ce}(x) = -\sum_{c=1}^{|C|} \hat{y}^c \log(p^{S,c}(T_s(x))). \tag{8}$$

where $(x, \hat{y}) \in D_{cl}$ and $P^{S,c}$ denotes the probability of the sample x belonging to class c as determined by the student model. The parameters of the teacher networks are updated using the exponential moving average of the updated student networks as follows:

$$\begin{aligned} \theta_a^T &\leftarrow \epsilon\theta_a^T + (1-\epsilon)\theta_a^S, \\ \theta_m^T &\leftarrow \epsilon\theta_m^T + (1-\epsilon)\theta_m^S, \end{aligned} \tag{9}$$

where ϵ is the momentum parameter. We repeat these two steps in an iterative manner until the networks converge.

4. Results and analysis

4.1. Datasets and metrics

We consider both first-person and third-person videos for benchmarking our proposed approach. Following [4,10], we use publicly available UCF101 [34] and HMDB51 [35] for third-person and EPIC-Kitchens [36] for first-person videos. We show experimentally that our approach adapts well to both of these scenarios.

UCF ↔ HMDB. We use the official split released by Chen *et al.* [7] for UCF ↔ HMDB to evaluate our CleanAdapt on video domain adaptation. In total, this dataset has 3209 third-person videos with 12 action classes. Specifically, all the videos are a subset of the original UCF101 [34] and HMDB51 [35] datasets with 12 classes common between them. Following [7], we use two settings: UCF101 → HMDB51, and HMDB51 → UCF101.

UCF ↔ HMDB$_{\text{small}}$. This dataset has 5 shared classes from UCF101 and HMDB51 datasets with a total of 1271 videos.

EPIC-Kitchens. This is the largest video domain adaptation dataset, which contains egocentric videos of fine-grained actions recorded in different kitchens. We follow the official split provided by Munro *et al.* [9]. This dataset contains videos from the three largest kitchens, *i.e.*, D1, D2, and D3, with 8 common action categories. EPIC-Kitchens has more class imbalance than UCF ↔ HMDB, making it more challenging [9].

Ego2Exo. We examine and evaluate the effectiveness of our proposed approach for cross-view transfer in videos using the challenging Ego2Exo [37] dataset. This dataset includes videos sourced from the Ego-Exo4D [38] dataset, leveraging their keystep annotations for action labels (*e.g.*, *make dough*, *prepare skillet*, *etc.*). In total, it comprises 4,100 ego videos and 4,986 exo videos for training, while the validation set consists of 3,168 samples from each view.

Metrics. We follow the standard protocol defined by [7,9] to compare our approach with state-of-the-art unsupervised domain adaptation methods [3,4,10] in terms of top-1 accuracy. We perform cross-domain retrieval experiments to evaluate the feature space learned by our model before and after adaptation. We report retrieval performance in terms of Recall at k (R@k), implying that if k closest nearest neighbors contain one video of the same class, the retrieval is considered correct.

4.2. Implementation details

We use the Inception I3D [1] network as the backbone for both RGB and Flow modalities. Following the prior video domain adaptation works [3,4,8,9], we use the Kinetics [1] pre-trained weights to initialize the I3D network. During training, we randomly sample 16 consecutive frames and perform the same data augmentation used in [3,8,9] for all our steps. We set the batch size to 48 for both UCF ↔ HMDB and EPIC-Kitchens datasets. We pre-compute optical flow using the TV-L1 algorithm.

Source pretraining stage. We train the model on the source dataset for 40 and 100 epochs with learning rates $1e-2$, and $2e-2$ for UCF ↔ HMDB and EPIC-Kitchens datasets, respectively. We reduce the learning rate by a factor of 10 after 10, 20 epochs for UCF ↔ HMDB. For EPIC-Kitchens, we decrease the learning rate by 10 after 50 epochs. We follow [8] for other hyperparameters.

Adaptation stage. We use the source pre-trained weights during the adaptation stage to initialize the I3D [1] network. The network is trained for 100 epochs with learning rates $1e-2$ and $2e-3$ for UCF ↔ HMDB and EPIC-Kitchens, respectively. The learning rate is reduced by 10 after 20, 40 epochs for UCF ↔ HMDB. In the case of EPIC-Kitchens, we reduce the learning rate by 10 after 10, 20 for EPIC-Kitchens. We set the values of momentum parameters ϵ as 0.99 in all experiments.

Our entire framework is implemented in PyTorch and uses 4 NVIDIA 2080Ti GPUs. On average, training takes around 1 h for UCF ↔ HMDB and approximately 7 h for EPIC-Kitchens datasets.

We now first provide a detailed comparison of our proposed approaches with state-of-the-art video domain adaptation methods on UCF ↔ HMDB, UCF ↔ HMDB$_{\text{small}}$, and EPIC-Kitchens datasets in Section 4.3. We provide some discussions to understand the effect of high-loss samples and over-confident pseudo-labels on the adaptation stage. In Section 4.3, we also show and compare the heatmaps generated by our CleanAdapt and source pre-trained model. In Section 4.4, we illustrate the impact of the hyperparameter τ and explore considerations for selecting an appropriate value for it. We also experimentally show the retrieval performance of our proposed approach as well as the source pre-trained model in Section 4.6. In Section 4.5, we discuss the impact of the teacher–student framework in detail.

4.3. Comparisons to the state-of-the-art methods

UCF ↔ HMDB. We present the quantitative results of both of our approaches CleanAdapt and CleanAdapt + TS for UCF ↔ HMDB dataset in Table 2 and compare our results with the state-of-the-art unsupervised source-free video domain adaptation approaches. For each approach in Table 2, we also report *source-only* and *target-supervised* results for fair comparisons wherever applicable. The source-only method refers to the f_a and/or f_m models trained only on the `train` split of the source dataset as described in Section 3.4 and tested directly on the `validation` split of the target dataset, which serves as a lower

Table 2
Performance comparison with state-of-the-art video domain adaptation methods on UCF101↔ HMDB51. Result for MM-SADA [9] is taken from Kim *et al.* [3]. The results for our methods are highlighted in gray. The average gains over the source-only model for the first and second best source-free approaches are highlighted in green and red, respectively.

Method	Venue	Two-stream?	Source-free?	Backbone	Datasets	
					UCF → HMDB	HMDB → UCF
Source only [7]				I3D	80.6	88.8
TA3N [7]	ICCV'19	✗	✗	I3D	81.4	90.5
Target supervised [7]				I3D	93.1	97.0
Source only [8]				I3D	80.3	88.8
SAVA [8]	ECCV'20	✗	✗	I3D	82.2	91.2
Target supervised [8]				I3D	95.0	96.8
Source only [9]				I3D	82.8	90.7
MM-SADA [9]	CVPR'20	✓	✗	I3D	84.2	91.1
Target supervised [9]				I3D	98.8	95.0
Source only [4]				I3D	82.8	89.8
STCDA [4]	CVPR'21	✓	✗	I3D	83.1	92.1
Target supervised [4]				I3D	95.8	97.7
Source only [3]				I3D	82.8	90.7
Kim *et al.* [3]	ICCV'21	✓	✗	I3D	84.7	92.8
Target supervised [3]				I3D	98.8	95.0
Source only [10]				I3D	80.3	88.8
CoMix [10]	NeurIPS'21	✗	✗	I3D	86.7	93.9
Target supervised [10]				I3D	95.0	96.8
Source only [16]				TRN	72.8	72.2
ATCoN [16]	ECCV'22	✗	✓	TRN	79.7 ▲ +6.9	85.3 ▲ +13.1
Source only [17]				I3D	80.8	91.0
MD-DMD [17]	MM'22	✓	✗	I3D	82.2	92.8
Target supervised [17]				I3D	98.8	95.0
Source only [18]				Transformer	81.1	86.8
MTRAN [18]	MM'22	✓	✓	Transformer	92.2 ▲ +11.1	95.3 ▲ +8.5
Costa *et al.* [23]	WACV'22	✗	✗	I3D	87.8	95.8
Source only				ResNet-101	76.4	78.1
MA^2LTD [24]	WACV'22	✗	✗	ResNet-101	85.0	86.6
Source only [25]				I3D	85.8	93.5
CIA [25]	CVPR'22	✓	✗	I3D-TRN	91.9	94.6
Target supervised [25]				I3D	96.8	99.1
Source only [26]				I3D	86.1	92.5
TranSVAE [26]	NeurIPS'23	✓	✗	I3D-TRN	87.8	99.0
Source only				I3D	80.6	89.3
CleanAdapt	Ours	✗	✓	I3D	86.1 ▲ +5.5	96.1 ▲ +6.8
CleanAdapt + TS	Ours	✗	✓	I3D	88.6 ▲ +8.0	96.7 ▲ +7.4
Target supervised				I3D	93.6	98.4
Source only				I3D	82.5	91.4
CleanAdapt	Ours	✓	✓	I3D	89.8 ▲ +7.3	99.2 ▲ +7.8
CleanAdapt + TS	Ours	✓	✓	I3D	93.6 ▲ +11.1	99.3 ▲ +7.9
Target supervised				I3D	95.3	99.3

bound of the adaptation performance. The target-supervised model is trained and tested on the `train` and `validation` split of the target dataset, respectively. This serves as an upper bound to the adaptation performance.

Next, we shift our focus towards comparing our method with the most advanced unsupervised video domain adaptation techniques available. TA3N [7], SAVA [8], CoMix [10], ATCoN [16], MA^2LTD [24] and Costa *et al.* [23] use only appearance stream in their methods. In contrast to these methods, STCDA [4], MM-SADA [9], and Kim *et al.* [3], MD-DMD [17], MTRAN [18], CIA [25], and TransVAE [26] leverage both appearance and motion streams. We show the results for both single-stream and two-stream versions of our model.

To show that the efficacy of our proposed approach is not solely due to the addition of the motion stream with appearance, we show our adaptation results for both single-stream (appearance only) and two-stream (appearance and motion) models. Our single-stream model achieves **86.1%** and **96.1%** top-1 accuracy with a gain of **5.5%** and **6.8%** over the source-only model for UCF → HMDB and HMDB → UCF datasets respectively. Further improvement in the adaptation performance is observed when we couple our method CleanAdapt with the teacher–student framework [14,32] resulting in CleanAdapt + TS. The teacher network serves as a regularization mechanism by producing consistent pseudo-labels, which, in turn, incentivizes the student model to make more confident predictions. This improved method CleanAdapt + TS achieves **88.6%** and **96.7%** top-1 accuracy, resulting in **8.0%** and **7.4%** gains over the source-only models, respectively.

In comparison, the best performing earlier existing model CoMix [10], which uses a temporal contrastive learning framework with background mixing, gives 6.4% and 5.1% gain for these two datasets, respectively. Note that all of these methods use the source data along with the target data during adaptation, whereas we use only target data in our approach and attain similar gains. Although the source-free video domain adaptation approaches such as ATCoN [16] and MTRAN [18] achieve better performance for HMDB→ UCF than our proposed approaches CleanAdapt and CleanAdapt + TS, it is not consistent across all the datasets as shown in Tables 2 and 5.

Similarly, our two-stream model CleanAdapt achieves state-of-the-art performance on both UCF → HMDB and HMDB → UCF datasets in terms of top-1 accuracy with the values of **89.8%** and **99.2%**, respectively. This is a significant gain of **7.3%** for UCF → HMDB and **7.8%** for HMDB → UCF over the source-only model without using any source-domain data which is much higher than the other source-dependent adaptation models. Our improved method CleanAdapt + TS achieves a further gain of **11.1%** for UCF → HMDB and **7.9%**. This affirms the assertion that in source-free, unsupervised video domain adaptation, utilizing the low-loss samples from the target domain during the adaptation phase is justified. It also highlights the efficacy of employing a slowly updated teacher network for generating pseudo-labels to fine-tune the student network using strongly augmented target domain videos.

We now aim to show the effect of using high-loss samples for adaptation and discuss if overconfident pseudo-labels can affect adaptation performance.

Table 3
Comparison with state-of-the-art image-based source-free domain adaptation techniques.

Method	Backbone	U → H	H → U
Source only	TRN	72.7	72.2
Kim *et al.* [22]	TRN	69.9	74.9
Li *et al.* [39]	TRN	74.4	67.3
Yang *et al.* [40]	TRN	75.3	76.3
Qiu *et al.* [41]	TRN	75.8	68.2
Source only	I3D	80.6	89.3
Yang *et al.* [11]	I3D	86.6	91.4
Liang *et al.* [42]	I3D	82.5	91.9
CleanAdapt	I3D	86.1	96.1
CleanAdapt + TS	I3D	**88.6**	**96.7**

Table 4
Performance comparisons with state-of-the-art video domain adaptation methods on UCF ↔ $\text{HMDB}_{\text{small}}$.

Method	U → H	H → U
Source only	97.3	96.8
SHOT [42]	99.3	99.5
3C-GAN [39]	98.3	99.5
SFDA [22]	98.0	99.3
HCL [43]	99.3	99.5
MTRAN [18]	100	100
Source only	98.3	97.8
CleanAdapt + TS	100	100
Target Supervised	100	100

What happens if we use only high-loss samples for adaptation? We trained our two-stream network with the high-loss samples instead of the proposed low-loss samples. For UCF → HMDB, we obtained 84.7% accuracy after adaptation with the high-loss samples, which is 5.1% less when adapted with the low-loss samples. We observe a similar drop for HMDB → UCF. This difference is even more significant when the noisy pseudo-labels are dominant (e.g., more than 12% on Epic-Kitchens). Nevertheless, these outcomes are in line with expectations, as demonstrated in Fig. 3, where it is evident that the noisy samples typically exhibit elevated loss values, thereby detrimentally affecting the fine-tuning performance when incorporated during the adaptation stage.

Do the overconfident pseudo-labels trigger error accumulation? A potential question arising here is whether the degradation of models and the further decline in the quality of pseudo-labels can occur due to the presence of *overconfident yet incorrect* pseudo-labels. Although error accumulation could be a possibility, we have found error accumulation to be negligible in practice. For example, the UCF pre-trained model selects low-loss samples with ∼98% accuracy in each epoch of the adaptation stage from HMDB.

Comparisons with self-training based methods. In Table 2, we compare our approach with the other self-training approaches [3,4,10]. Our method re-purposed the learning from noisy labels based pseudo-label selection method that shows better performance than all these.

Comparisons with image-based source-free methods. In Table 3, we compare our approach with state-of-the-art image-based source-free methods. For [22,39–41], we report the values with TRN [48] as their backbone network. Our model CleanAdapt + TS achieves higher gain over their corresponding source-only model than all these image-based source-free methods. We have also adopted the frameworks proposed by [11,42] with our 3D backbone network. Liang *et al.* [42] perform marginally better than the source-only model. Yang *et al.* [11] performance is comparable to ours on UCF → HMDB, but significantly worse on HMDB→UCF.

UCF ↔ $\text{HMDB}_{\text{small}}$. To provide a thorough evaluation and compare fairly with existing methods, we compare the performance of our method CleanAdapt + TS on the UCF ↔ $\text{HMDB}_{\text{small}}$ dataset in Table 4. The dataset size is very limited, and thus CleanAdapt + TS achieves scores comparable to the target-supervised baseline for both the UCF → $\text{HMDB}_{\text{small}}$ and HMDB ↔ $\text{UCF}_{\text{small}}$. We compare our approach CleanAdapt + TS with state-of-the-art techniques such as [18,22,39,42,43] and achieves superior performance.

EPIC-Kitchens. In Table 5, we compare the results of our approach with state-of-the-art image-based methods extended for videos as well as video-based domain adaptation. We implement our model to replicate the source-only and target-supervised performance as reported in [3]. Note that there is a minor difference (∼2.7% and ∼3.5%) in the performance of the source-only model reported in MM-SADA [9] and both of our models. Comparable distinctions to those observed in [9] can be identified in [10], primarily attributable to the non-deterministic operations associated with CUDA. However, such minor differences of source-only accuracy is not a concern for evaluating domain adaptation performance. The most important metric here is the gain achieved after adaptation over the source-only model. On average, the source-free methods demonstrate a maximum improvement of 3% over the source-only model. In contrast, our improved method, CleanAdapt + TS, achieves 7.6% improvement despite being simple.

MM-SADA [9] is the first to report domain adaptation results on the EPIC-Kitchens dataset, achieving an average of 4.8% gain on top of their source-only model followed by Song *et al.* [4] reporting an average gain of 5.7%. Kim *et al.* [3] show an improvement of 5.5% averaged over 6 datasets. CTAN [5] achieves a modest gain of 1.5% over the source-only models. However, all of these methods use the source dataset for adaptation. In contrast to these prior approaches, our simple yet powerful source-free approaches, CleanAdapt and CleanAdapt + TS, achieve an average of **7.5%** and **7.6%** gain over the source-only model, respectively. The primary source of the performance boost achieved by our methods, despite their simplicity, can be attributed to the abundance of clean samples with low-loss values compared to the noisy ones.

Ego2Exo. Following [37], we use the pre-extracted Omnivore Swin-L [51] features in the Ego-Exo4D [38] dataset. We train a 2-layer linear classifier on top of this features keeping all other hyper-parameters as described above.

As shown in Table 6, TA3N [7] and TransVAE [26] achieve average improvements of 5.41% and 5.67%, respectively, over the source-only model on the Ego2Exo dataset. In contrast, the zero-shot baselines, EgoVLP [52] and LaViLA [50], deliver more modest gains of 0.59% and 0.47%, respectively. Although LaGTran [37] achieves a 9.52% improvement over the source-only model, it relies on text data associated with the videos, which requires manual effort and contradicts our assumption. In contrast, our proposed method, CleanAdapt, achieves a 9.76% improvement over the source-only model without needing any text data in the target domain.

Visualization. In Fig. 6, we show the Class Activation MAP (CAM) visualizations of our adapted model and compare them with the source-only model. The visualization shows that the source-only model attends to the background of the scene and makes incorrect predictions, while the adapted model focuses on the action component of the video to make correct predictions.

Comparisons with Zero-Shot Vision-Language Models. In addition to the state-of-the-art video domain adaptation approaches, we also compare our methods CleanAdapt and CleanAdapt + TS with the following pre-trained vision-language models.

Video-LLaVA [53]. These baselines integrate visual representations into the language feature space, contributing to the development of unified Large Vision-Language Models (LVLMs). We prompt this LVLM with the following text - "*USER: < video > Pick the action being performed in the video from the list below: ['take', 'put', 'open', 'close', 'wash', 'cut', 'mix', 'pour'] ASSISTANT:*".

EgoVLP [49]. A zero-shot baseline leverages the video-language pre-trained backbone from the Ego4D [54] dataset. We compute the embeddings for the *class names* using the *distilbert-base-uncased* model,

Table 5
Performance comparison with state-of-the-art video domain adaptation methods on EPIC-Kitchens dataset. Results with single-stream models are highlighted in cyan whereas the results with two-stream networks are highlighted in gray. The average gains over the source-only model for the first and second best source-free approaches are highlighted in green and red, respectively. See [44–47].

Method	Venue	Source-free?	Backbone	D2→D1	D3→D1	D1→D2	D3→D2	D1→D3	D2→D3	Mean
Source only			I3D	42.5	44.3	42.0	56.3	41.2	46.5	45.5
MMD [44]	ICML'15	✗	I3D	43.1	48.3	46.6	55.2	39.2	48.5	46.8
AdaBN [45]	PR'18	✗	I3D	44.6	47.8	47.0	54.7	40.3	48.8	47.2
MCD [46]	CVPR'18	✗	I3D	42.1	47.9	46.5	52.7	43.5	51.0	47.3
MM-SADA [9]	CVPR'20	✗	I3D	48.2	50.9	49.5	56.1	44.1	52.7	50.3
STCDA [4]	CVPR'21	✗	I3D	49.0	52.6	52.0	55.6	45.5	52.5	51.2
Kim *et al.* [3]	ICCV'21	✗	I3D	49.5	51.5	50.3	56.3	46.3	52.0	51.0
MD-DMD [17]	MM'22	✗	I3D	50.3	51.0	56.0	54.7	47.3	52.4	52.0
CIA [25]	CVPR'22	✗	I3D	52.5	47.8	49.8	53.2	52.2	57.6	52.2
Target Supervised			I3D	62.8	62.8	71.7	71.7	74.0	74.0	69.5
Source only			I3D	35.5	38.1	39.4	40.5	32.0	39.2	37.5
CTAN [5]	TCSVT'23	✗	I3D	36.6	39.3	41.3	41.3	35.0	40.6	39.0
Target Supervised			I3D	60.2	60.2	64.7	64.7	52.8	52.8	59.2
Source only			I3D	35.4	34.6	32.8	35.8	34.1	39.1	35.3
DANN [2]	ICML'15	✗	I3D	38.3	38.8	37.7	42.1	36.6	41.9	39.2
ADDA [47]	CVPR'17	✗	I3D	36.3	36.1	35.4	41.4	34.9	40.8	37.4
TA3N [7]	ICCV'19	✗	I3D	40.9	39.9	34.2	44.2	37.4	42.8	39.9
CoMix [10]	NeurIPS'21	✗	I3D	38.6	42.3	42.9	49.2	40.9	45.2	43.2
Target Supervised			I3D	57.0	57.0	64.0	64.0	63.7	63.7	61.5
Source only			Transformer	43.7	51.1	40.5	36.2	48.9	45.2	44.2
Liang *et al.* [42]	ICML'20	✓	Transformer	44.1	53.9	40.8	36.5	49.0	45.3	44.9
Li *et al.* [39]	CVPR'20	✓	Transformer	44.7	54.3	41.0	36.7	49.9	45.4	45.4
Kim *et al.* [22]	TAI'21	✓	Transformer	44.4	54.9	41.3	37.2	49.8	45.2	45.5
HCL [43]	NeurIPS'21	✓	Transformer	45.1	55.6	41.5	36.9	50.2	45.7	45.8
MTRAN [18]	MM'22	✓	Transformer	46.3	58.2	42.2	38.1	52.3	46.1	47.2 ▲ +3.0
Source only			I3D	40.9	38.6	39.3	41.3	37.3	42.4	39.9
CleanAdapt	Ours	✓	I3D	44.6 ▲ +3.7	40.7 ▲ +2.1	44.5 ▲ +5.2	47.1 ▲ +5.8	40.9 ▲ +3.6	45.7 ▲ +3.3	43.9 ▲ +4.0
Target Supervised			I3D	60.5	60.5	68.4	68.4	68.8	68.8	65.9
Source only			I3D	41.8	40.0	46.0	45.6	38.9	44.4	42.8
CleanAdapt	Ours	✓	I3D	46.2 ▲ +4.4	47.8 ▲ +7.8	52.7 ▲ +6.7	54.4 ▲ +8.8	47.0 ▲ +8.1	52.7 ▲ +8.3	50.3 ▲ +7.5
Target Supervised			I3D	62.1	62.1	72.8	72.8	72.3	72.3	69.1
Source only			I3D	41.8	41.1	41.9	46.1	37.3	43.9	42.0
CleanAdapt + TS	Ours	✓	I3D	48.3▲ +6.5	48.7 ▲ +7.6	49.9 ▲ +8.0	56.3 ▲ +10.2	44.6 ▲ +7.3	48.9 ▲ +5.0	49.6 ▲ +7.6
Target Supervised			I3D	62.3	62.3	72.7	72.7	71.1	71.1	68.4

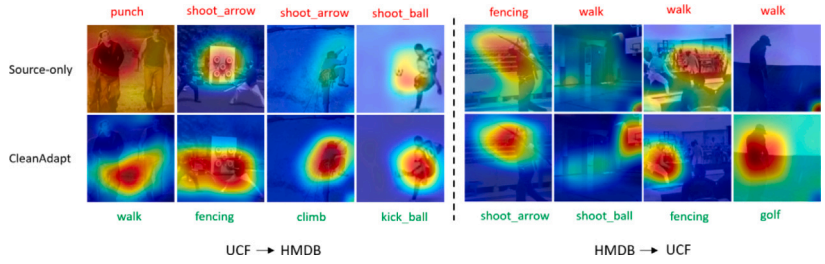


Fig. 6. Class activation map (CAM) on target-domain videos of the UCF ↔ HMDB dataset. The actions in green are correct predictions, while actions in red are incorrect. It is worth noting that the adapted model in the bottom row emphasizes the action component rather than the contextual scene aspect. (Best viewed in color.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Performance comparison of state-of-the-art video domain adaptation method on Ego2Exo [37] dataset.

	Ego→Exo	Exo→Ego	Avg.
Unsupervised Adaptation			
Source Only	8.39	15.66	12.03
TA3N [7]	6.92	27.95	17.44
TransVAE [26]	12.06	23.34	17.70
Zero-shot Video Recognition			
EgoVLP [49]	5.89	19.35	12.62
LaViLA [50]	5.86	19.13	12.50
LaGTran [37]	12.34	30.76	21.55
Target Sup.	17.91	33.19	25.55
Source-free Unsupervised Adaptation			
Source-only	18.08	40.02	29.05
CleanAdapt ($\tau = 0.5$)	27.14	50.47	38.81
Target Sup.	30.99	52.05	41.52

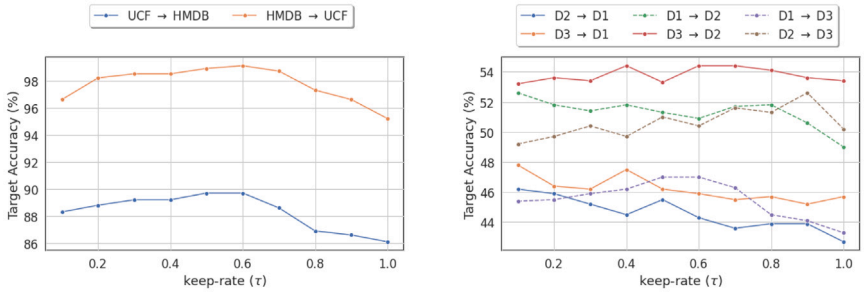
and similarly, we extract video embeddings using the Ego4D [54] pre-trained backbone. Finally, the class name with the highest similarity is selected.

LaViLA [50]. This zero-shot baseline leverages the video-language pre-training by making a large-language model (LLM) conditioned on the visual inputs using the cross-attention layer on the Ego4D [54] dataset. Following [50], we compute the embeddings for the *class names* using the *distilbert-base-uncased* model, and similarly, we extract video embeddings using the Ego4D [54] pre-trained backbone. Finally, the class name with the highest similarity is selected.

The results of the zero-shot baselines on the EPIC-Kitchens dataset are presented in Table 7. From the table, it is evident that zero-shot vision-language models do not perform particularly well on egocentric video recognition tasks. For instance, Video-LLaVA [53], which is trained on generic third-person videos, achieves only 33.76% average top-1 accuracy. In contrast, EgoVLP [49], and LaViLA [50], pre-trained on egocentric videos from the Ego4D [54] dataset, outperform Video-LLaVA but still fall short by 22.4% and 1.53% when compared with our proposed method CleanAdapt + TS.

Table 7
Performance comparison with zero-shot vision-language models on EPIC-Kitchens dataset. Results with zero-shot vision-language models are highlighted in cyan whereas the results our proposed model are highlighted in gray.

Method	D2→D1	D3→D1	D1→D2	D3→D2	D1→D3	D2→D3	Mean
Zero-shot Baselines							
Video-LLaVA [53]	32.64	32.64	33.73	33.73	34.90	34.90	33.76
EgoVLP [49]	23.60	23.60	28.00	28.00	30.08	30.08	27.22
LaViLA [50]	46.44	46.44	50.13	50.13	47.64	47.64	48.07
Source-free Domain Adaptation							
Source only	41.8	41.1	41.9	46.1	37.3	43.9	42.0
CleanAdapt + TS	48.3▲ +6.5	48.7 ▲ +7.6	49.9 ▲ +8.0	56.3 ▲ +10.2	44.6 ▲ +7.3	48.9 ▲ +5.0	49.6 ▲ +7.6
Target Supervised	62.3	62.3	72.7	72.7	71.1	71.1	68.4



(a) UCF ↔ HMDB (b) EPIC-Kitchens

Fig. 7. Hyperparameter search for the value of keep-rate τ for UCF101↔ HMDB51 and EPIC-Kitchens dataset. The keep-rate τ controls the number of samples to be selected as clean due to low-loss values computed against the pseudo-labels. The results reported here are for the two-stream network. (Best viewed in color.)

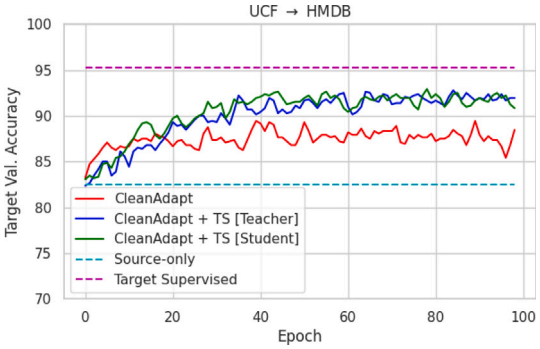


Fig. 8. Analysis of the effect of CleanAdapt + TS on the target validation accuracy as compared to source-only, CleanAdapt, and target-supervised methods. (Best viewed in color.).

4.4. Hyperparameter search

The only hyperparameter our model introduces is the keep-rate τ. It controls the number of target domain samples to be chosen from each class with low loss values in the adaptation stage. Fig. 7 shows the ablation results of varying τ in terms of validation accuracy for the target domain.

Empirically, we verify that the choice of keep-rate τ is important. As mentioned earlier, the samples from the target domain `train` set pseudo-labeled by the source-only model have inherently noisy labels. The choice of keep-rate $\tau = 1$ is equivalent to choosing all the samples for retraining the model on the target domain. However, the noisy pseudo-labels lead to a sub-optimal adaptation performance for all the datasets. For example, the adapted model gives top-1 accuracy of 86.1% on UCF → HMDB and 95.2% on HMDB → UCF respectively when τ is set to 1. However, when keep-rate τ is set to 0.6 gives top-1 accuracy of 89.8% and 99.2% on UCF → HMDB and HMDB → UCF respectively.

Table 8
Cross-domain video retrieval results on UCF ↔ HMDB dataset. Given queries from the target domain, we evaluate retrieved videos from the source domain in terms of R@k, where $k \in \{1, 5, 10\}$. Note that, all models reported here are two-stream networks and we average the similarity score from each modality to retrieve the source videos.

Method	UCF → HMDB			HMDB → UCF		
	R@1	R@5	R@10	R@1	R@5	R@10
Source Only	0.82	0.87	0.90	0.88	0.94	0.95
CleanAdapt	**0.92**	**0.97**	**0.99**	**0.91**	**0.97**	**0.98**

4.5. Impact of teacher–student framework

To assess the effectiveness of the teacher–student framework, we generate a plot of target validation accuracy for each epoch during the training phase. We compare the performance of the source-only model, CleanAdapt, and CleanAdapt + TS, and target-supervised models. Fig. 8 clearly demonstrates that the teacher–student framework contributes to stable pseudo-label generation, resulting in improved adaptation performance. To be precise, the teacher model serves as a regularizing factor for the student model, accomplishing this by producing consistent pseudo-labels. Consequently, the student model is guided to make gradual changes rather than abrupt ones. These pseudo-labels are treated by the student model as actual labels and take strongly augmented videos to generate robust features.

4.6. Cross-domain video retrievals

We examine the feature space learned by our adapted model CleanAdapt to gain insights into its predictions through cross-domain video retrieval performance. Given a query video of a particular class from the target domain, we aim to retrieve videos from the source domain with the same semantic category. We show the results for the two-stream networks, where we first compute the similarity scores for the individual modalities and average them for final retrieval. We evaluate both the source-only and the proposed method CleanAdapt quantitatively as well as qualitatively.

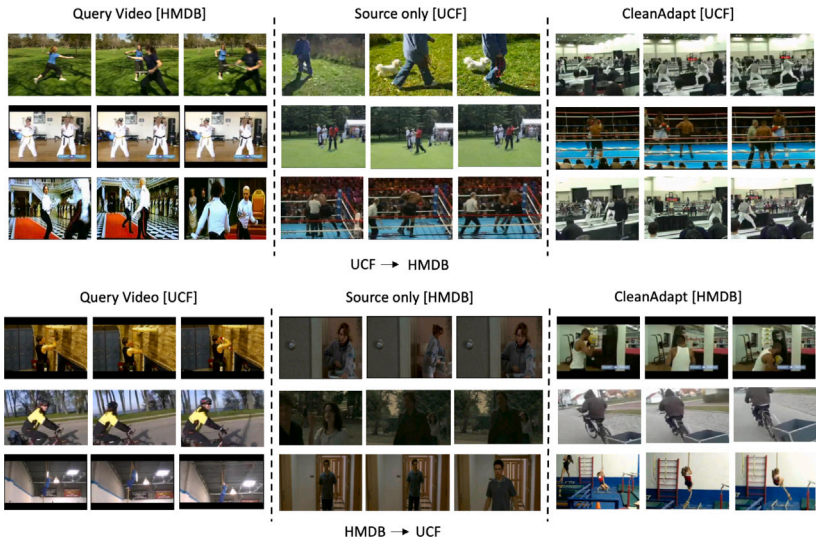


Fig. 9. Nearest neighbor retrieval results for the UCF → HMDB and the HMDB → UCF dataset. The left column shows the query videos from the target domain. The middle column shows the retrieved source videos using the source-only model, and the right column shows the source videos retrieved using our proposed model. (Best viewed in color.).

Table 9
Cross-domain video retrieval results on the EPIC-Kitchens dataset. Given queries from the target domain, we evaluate retrieved videos from the source domain in terms of R@k, where $k \in \{1, 5, 10\}$. All the models reported here are two-stream networks and we average the similarity score from each modality to retrieve the source videos.

Method	D2 → D1			D3 → D1			D1 → D2			D3 → D2			D1 → D3			D2 → D3		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Source only	0.35	0.65	0.77	**0.38**	0.68	0.79	0.35	**0.75**	0.86	0.41	**0.77**	0.84	0.34	0.68	0.82	**0.42**	**0.74**	**0.84**
CleanAdapt	**0.42**	**0.68**	**0.80**	0.37	**0.75**	**0.83**	**0.42**	0.74	**0.87**	**0.46**	**0.77**	**0.85**	**0.35**	**0.69**	**0.83**	0.40	0.70	0.82

In Table 8, we show the quantitative results for the cross-domain video retrieval task for the UCF ↔ HMDB dataset. Our model retrieves better source videos from the target queries with R@1 of 0.92 and 0.91 as compared to the source-only model, which achieves only 0.82 and 0.88 on UCF → HMDB and HMDB → UCF datasets respectively. In Fig. 9, we show qualitative retrieval results for the UCF → HMDB. Our model can correctly retrieve the source videos of the same semantic categories as the target query videos.

As shown in Table 9, our proposed approach achieves better retrieval performance in most of the cases than the source-only model for the EPIC-Kitchen dataset. Only for D2 → D3, our model under-performs the source-only model. This can be attributed to the fact that our model does not use source data during the adaptation stage, and thus, the model might start forgetting some attributes of the source dataset.

4.7. Limitations and future work

In our model, we do not explicitly incorporate the spatiotemporal relationships inherent in videos. This information is crucial for capturing how objects and actions evolve over time, which can significantly enhance the model's ability to understand complex video content. Without modeling these relationships, the model may struggle to effectively adapt to more challenging tasks such as video object segmentation, where it is essential to accurately track and delineate objects across frames.

A key assumption of our method is that each sample in both the source and target domains is associated with a single label and that they share a common label space C. While this is a standard setup in the literature, it may not always apply. For instance, CharadesEgo [55] contains both first- and third-person videos, with each video having multiple labels. Our proposed approaches cannot be directly applied to such scenarios, and extending them to handle multi-label settings is left as future work.

However, we believe that the fundamental concept of selecting clean samples from noisy pseudo-labeled data will still be advantageous for tasks beyond video domain adaptation.

5. Conclusion

In this work, we address the relatively under-explored problem of source-free video domain adaptation and propose two simple yet effective approaches: CleanAdapt and CleanAdapt + TS. Our framework is based on self-training in which we generate noisy pseudo-labels for the target domain data using the source pre-trained model. Moreover, if we can filter out the noisy samples with varying τ using our proposed approach and use only the clean samples for fine-tuning, we achieve state-of-the-art performance without any video-specific modeling. To mitigate this issue of noisy pseudo-labels impeding the adaptation performance, we leverage the deep memorization effect [13] to identify and select the clean samples. Furthermore, we demonstrate that the quality of the clean samples can be improved by introducing a teacher–student framework, which in turn enhances the overall reliability of the adaptation training process and results in further performance improvements. Our methods consistently outperform recent image-based and video-based UDA methods without any source domain videos, thus establishing a new state of the art across several benchmarks.

While it is true that we get approximately 90% accurate pseudo-labels for the HMDB → UCF task, we want to emphasize that this is not always the case. For instance, in the EPIC-Kitchens dataset, the source-only model generates pseudo-labels with only about 42.0% *clean samples*. As demonstrated in Table 5 and Fig. 7, training with all pseudo-labels without filtering the clean samples leads to sub-optimal performance. This underscores the importance of selecting an appropriate keep-rate (τ). We would like to emphasize that the success of our method relies on the noise level in the pseudo-labeled target domain samples and the network's ability to avoid memorizing them. These might fail when the amount of noise is too much (e.g., 90%

noisy samples). Nonetheless, as demonstrated empirically, our method is effective even with a reasonable noise level.

CRediT authorship contribution statement

Avijit Dasgupta: Visualization, Validation, Software, Methodology, Conceptualization. **C.V. Jawahar:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization. **Karteek Alahari:** Writing – review & editing, Writing – original draft, Supervision, Conceptualization.

Declaration of competing interest

We have declared that we do not have any financial and personal relationships with other people or organizations that could inappropriately influence (bias) our work.

Acknowledgments

Avijit Dasgupta is supported by a Google Ph.D. India Fellowship. Karteek Alahari is supported in part by the ANR grant AVENUE (ANR-18-CE23-0011).

Data availability

The data used in our manuscript is publicly available and not owned by us. But it can be easily downloaded and preprocessed.

References

[1] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in: CVPR, 2017.
[2] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: ICML, 2015.
[3] D. Kim, Y.-H. Tsai, B. Zhuang, X. Yu, S. Sclaroff, K. Saenko, M. Chandraker, Learning cross-modal contrastive features for video domain adaptation, in: ICCV, 2021.
[4] X. Song, S. Zhao, J. Yang, H. Yue, P. Xu, R. Hu, H. Chai, Spatio-temporal contrastive domain adaptation for action recognition, in: CVPR, 2021.
[5] X. Liu, S. Zhou, T. Lei, P. Jiang, Z. Chen, H. Lu, First-person video domain adaptation with multi-scene cross-site datasets and attention-based methods, in: IEEE Transactions on Circuits and Systems for Video Technology, 2023.
[6] A. Jamal, V.P. Namboodiri, D. Deodhare, K. Venkatesh, Deep domain adaptation in action space, in: BMVC, 2018.
[7] M.-H. Chen, Z. Kira, G. AlRegib, J. Yoo, R. Chen, J. Zheng, Temporal attentive alignment for large-scale video domain adaptation, in: ICCV, 2019.
[8] J. Choi, G. Sharma, S. Schulter, J.-B. Huang, Shuffle and attend: Video domain adaptation, in: ECCV, 2020.
[9] J. Munro, D. Damen, Multi-modal domain adaptation for fine-grained action recognition, in: CVPR, 2020.
[10] A. Sahoo, R. Shah, R. Panda, K. Saenko, A. Das, Contrast and mix: Temporal contrastive video domain adaptation with background mixing, in: NeurIPS, 2021.
[11] S. Yang, J. van de Weijer, L. Herranz, S. Jui, Exploiting the intrinsic neighborhood structure for source-free domain adaptation, in: NeurIPS, 2021.
[12] D. Guan, J. Huang, S. Lu, A. Xiao, Scale variance minimization for unsupervised domain adaptation in image segmentation, in: Pattern Recognition, 2021.
[13] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M.S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, S. Lacoste-Julien, A closer look at memorization in deep networks, in: ICML, 2017.
[14] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C.A. Raffel, E.D. Cubuk, A. Kurakin, C.-L. Li, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, in: NeurIPS, 2020.
[15] A. Dasgupta, C.V. Jawahar, K. Alahari, Overcoming label noise for source-free unsupervised video domain adaptation, in: ICVGIP, 2022.
[16] Y. Xu, J. Yang, H. Cao, K. Wu, M. Wu, Z. Chen, Source-free video domain adaptation by learning temporal consistency for action recognition, in: ECCV, 2022.
[17] Y. Yin, B. Zhu, J. Chen, L. Cheng, Y.-G. Jiang, Mix-DANN and dynamic-modal-distillation for video domain adaptation, in: MM, 2022.
[18] Y. Huang, X. Yang, J. Zhang, C. Xu, Relative alignment network for source-free multimodal video domain adaptation, in: MM, 2022.
[19] B. Zhou, A. Andonian, A. Oliva, A. Torralba, Temporal relational reasoning in videos, in: ECCV, 2018.
[20] B. Pan, Z. Cao, E. Adeli, J.C. Niebles, Adversarial cross-domain action recognition with co-attention, in: AAAI, 2020.
[21] J. Choi, G. Sharma, M. Chandraker, J.-B. Huang, Unsupervised and semi-supervised domain adaptation for action recognition from drones, in: WACV, 2020.
[22] Y. Kim, D. Cho, K. Han, P. Panda, S. Hong, Domain adaptation without source data, in: IEEE Transactions on Artificial Intelligence, 2021.
[23] V.G.T. da Costa, G. Zara, P. Rota, T. Oliveira-Santos, N. Sebe, V. Murino, E. Ricci, Dual-head contrastive domain adaptation for video action recognition, in: WACV, 2022.
[24] P. Chen, Y. Gao, A.J. Ma, Multi-level attentive adversarial learning with temporal dilation for unsupervised video domain adaptation, in: WACV, 2022.
[25] L. Yang, Y. Huang, Y. Sugano, Y. Sato, Interact before align: Leveraging cross-modal knowledge for domain adaptive action recognition, in: CVPR, 2022.
[26] P. Wei, L. Kong, X. Qu, X. Yin, Z. Xu, J. Jiang, Z. Ma, Unsupervised video domain adaptation: A disentanglement perspective, in: NeurIPS, 2023.
[27] S. Broomé, E. Pokropek, B. Li, H. Kjellström, Recur, attend or convolve? On whether temporal modeling matters for cross-domain robustness in action recognition, in: WACV, 2023.
[28] Y. Xu, H. Cao, Z. Chen, X. Li, L. Xie, J. Yang, Video unsupervised domain adaptation with deep learning: A comprehensive survey, 2022, arXiv preprint arXiv:2211.10412.
[29] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, B. An, Can cross entropy loss be robust to label noise? in: IJCAI, 2021.
[30] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, F. Wen, Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation, in: CVPR, 2021.
[31] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, M. Sugiyama, Co-teaching: Robust training of deep neural networks with extremely noisy labels, in: NeurIPS, 2018.
[32] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: NeurIPS, 2017.
[33] E.D. Cubuk, B. Zoph, J. Shlens, Q.V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: CVPRW, 2020.
[34] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, in: arXiv preprint , 2012, arXiv:1212.0402.
[35] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: ICCV, 2011.
[36] D. Damen, H. Doughty, G.M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, M. Wray, Scaling egocentric vision: The EPIC-KITCHENS dataset, in: ECCV, 2018.
[37] T. Kalluri, B. Majumder, M. Chandraker, Tell, don't show! language guidance eases transfer across domains in images and videos, in: ICML, 2024.
[38] K. Grauman, A. Westbury, L. Torresani, K. Kitani, J. Malik, T. Afouras, K. Ashutosh, V. Baiyya, S. Bansal, B. Boote, et al., Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives, in: CVPR, 2024.
[39] R. Li, Q. Jiao, W. Cao, H.-S. Wong, S. Wu, Model adaptation: Unsupervised domain adaptation without source data, in: CVPR, 2020.
[40] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, S. Jui, Unsupervised domain adaptation without source data by casting a bait, in: Computer Vision and Image Understanding, 2023.
[41] Z. Qiu, Y. Zhang, H. Lin, S. Niu, Y. Liu, Q. Du, M. Tan, Source-free domain adaptation via avatar prototype generation and adaptation, in: IJCAI, 2021.
[42] J. Liang, D. Hu, J. Feng, Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, in: ICML, 2020.
[43] J. Huang, D. Guan, A. Xiao, S. Lu, Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data, in: NeurIPS, 2021.
[44] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: ICML, 2015.
[45] Y. Li, N. Wang, J. Shi, X. Hou, J. Liu, Adaptive batch normalization for practical domain adaptation, in: Pattern Recognition, 2018.
[46] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: CVPR, 2018.
[47] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: CVPR, 2017.
[48] B. Zhou, A. Andonian, A. Oliva, A. Torralba, Temporal relational reasoning in videos, in: ECCV, 2018.
[49] K.Q. Lin, J. Wang, M. Soldan, M. Wray, R. Yan, E.Z. Xu, D. Gao, R.-C. Tu, W. Zhao, W. Kong, et al., Egocentric video-language pretraining, in: NeurIPS, 2022.
[50] Y. Zhao, I. Misra, P. Krähenbühl, R. Girdhar, Learning video representations from large language models, in: CVPR, 2023.
[51] R. Girdhar, M. Singh, N. Ravi, L. Van Der Maaten, A. Joulin, I. Misra, Omnivore: A single model for many visual modalities, in: CVPR, 2022.
[52] W. Lin, M.J. Mirza, M. Kozinski, H. Possegger, H. Kuehne, H. Bischof, Video test-time adaptation for action recognition, in: CVPR, 2023.
[53] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, L. Yuan, Video-llava: Learning united visual representation by alignment before projection, 2023, arXiv preprint arXiv:2311.10122.

[54] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al., Ego4d: Around the world in 3,000 hours of egocentric video, in: CVPR, 2022.

[55] G.A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, K. Alahari, Charades-ego: A large-scale dataset of paired third and first person videos, 2018, arXiv preprint arXiv:1804.09626.

Avijit Dasgupta is a Ph.D. student at IIIT Hyderabad, where he works with CVIT lab. His research focuses on developing video understanding models with the ability to generalize effectively across previously unseen domains. He was the recepient of the Google India PhD Fellowship.

C. V. Jawahar received the PhD degree from IIT Kharagpur. Since December 2000, he has been with IIIT Hyderabad, where he is currently a professor. His research interests include computer vision, machine learning, and multimedia systems.

Karteek Alahari is a research director at Inria in the Thoth research team, deputy scientific director for AI at Inria, and co-director of the PEPR IA program. His research focuses on visual understanding in large-scale datasets, emphasizing robust visual representations through incremental learning, weak supervision, and adversarial training. He completed his Ph.D. in 2010 under Philip Torr and has affiliations with ENS and the Visual Geometry Group at Oxford.