The Unreasonable Effectiveness of Large Language-Vision Models for Source-free Video Domain Adaptation

Giacomo Zara[1*], Alessandro Conti[1*], Subhankar Roy[3], Stéphane Lathuilière[3], Paolo Rota[1], Elisa Ricci[1,2]

[1]University of Trento, Italy [2]Fondazione Bruno Kessler, Italy

[3]LTCI, Télécom Paris, Institut polytechnique de Paris, France

{giacomo.zara,alessandro.conti-1}@unitn.it

Abstract

*Source-Free Video Unsupervised Domain Adaptation (SFVUDA) task consists in adapting an action recognition model, trained on a labelled source dataset, to an unlabelled target dataset, without accessing the actual source data. The previous approaches have attempted to address SFVUDA by leveraging self-supervision (e.g., enforcing temporal consistency) derived from the target data itself. In this work, we take an orthogonal approach by exploiting "web-supervision" from Large Language-Vision Models (LLVMs), driven by the rationale that LLVMs contain a rich world prior surprisingly robust to domain-shift. We showcase the unreasonable effectiveness of integrating LLVMs for SFVUDA by devising an intuitive and parameter-efficient method, which we name **D**omain **A**daptation with **L**arge **L**anguage-Vision models (DALL-V), that distills the world prior and complementary source model information into a student network tailored for the target. Despite the simplicity, DALL-V [1] achieves significant improvement over state-of-the-art SFVUDA methods.*

1. Introduction

Video analysis tasks, such as action recognition, have long been investigated in computer vision, due to the numerous applications, ranging from video surveillance to social robotics [50, 14, 29]. Major progress has been made in the last decade with the development of specialized deep architectures, such as 3D CNNs [4, 3] and Video Transformers [36], trained on large-scale annotated datasets. However, obtaining sufficient labelled training videos for real-world scenarios can be very costly and time-consuming. In order to alleviate the burden of annotating large scale datasets Video-based Unsupervised Domain
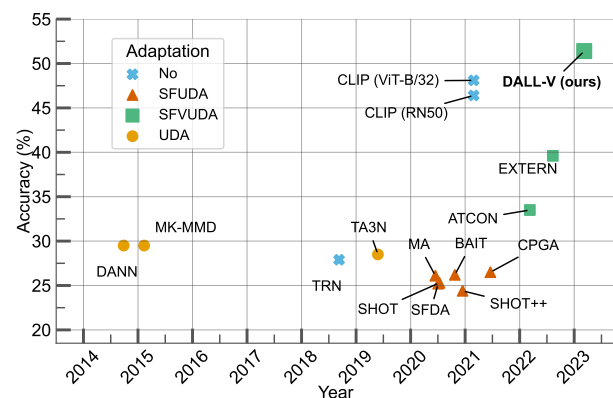

Figure 1: Performance over time by various methods on the **_Daily-DA_** video benchmark. A pre-trained LLVM (*e.g.*, CLIP [31]) is surprisingly better when compared with Unsupervised Domain Adaptation (UDA), Source Free UDA (SFUDA) and Source Free Video-based UDA (SFVUDA) methods. Our proposed DALL-V that is built on top of LLVM successfully outperforms all existing baselines.

Adaptation (VUDA) [1, 28, 2] have been introduced. The VUDA methods are derived from the common idea of transferring knowledge from a labelled *source* domain to an unlabelled *target* domain. While considering different strategies for adaptation, these approaches have all shown significant improvements in the robustness of learnt video models without requiring annotated target data.

In the last few years, the field of computer vision has also witnessed the emergence of a new generation of powerful deep architectures, trained on mammoth *internet-scale* image-text datasets [35]. These models, commonly known as *foundation models* [32, 37, 31, 10] or Large Language-Vision Models (LLVM), have achieved outstanding performance, and have become a cornerstone of modern computer vision research. In particular, recent works such as CLIP [31] or FLAVA [37] have shown that rich visual representations for images can be learned from natural language descriptions as a form of supervision. These pre-trained
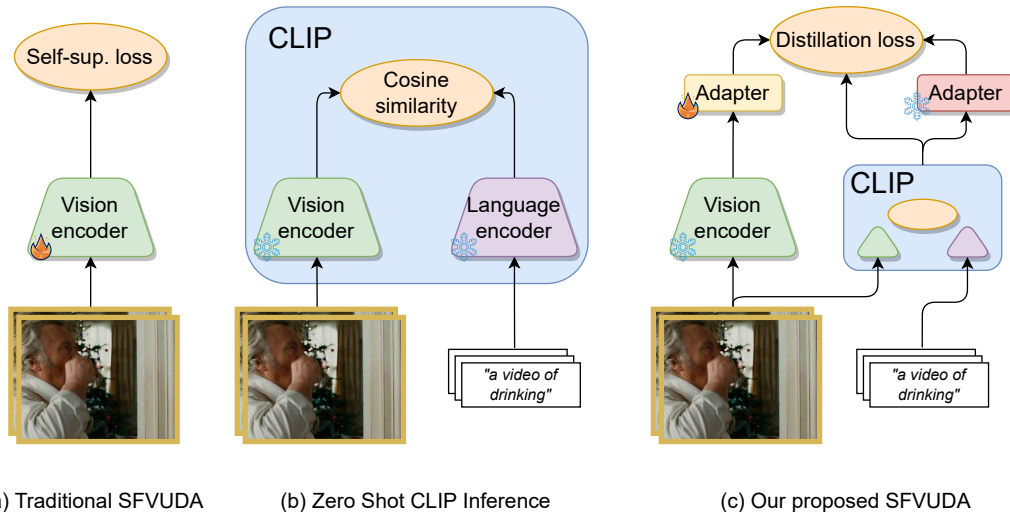
Figure 2: (a) In traditional SFVUDA a source-trained model is adapted to the *unlabelled* target dataset using self-supervision such as temporal consistency [44, 46] among frames. (b) LLVM (*e.g.*, CLIP [31]) uses cosine similarity between the feature and language representation to predict the most probable class in a zero-shot (ZS) manner. (c) Our proposed SFVUDA solution *distills* the ZS-CLIP and source model predictions, while introducing very few learnable parameters, namely adapters. The 🔥 means the network is trainable, while the ❄ means the network is frozen.

LLVMs are now publicly available and can be easily integrated into any recognition system. Lately, researchers have also shown that the LLVMs can be applied with success to the video domain and specifically to the supervised action recognition task [40, 41].

Despite the significant progress in the VUDA literature, to date, the role of LLVM in the context of action recognition has been overlooked. We argue that the LLVMs have a lot to offer to the progress of VUDA methods, which is currently untapped. Our work starts from a preliminary analysis demonstrating that current sophisticated VUDA methods can be largely outperformed by *off-the-shelf* publicly available LLVMs like CLIP, without needing explicit adaptation (see Fig. 1). This observation highlights the need for a paradigm shift among VUDA methods. The benefit of LLVM models is so significant that it raises a pertinent question: *If explicit alignment methodologies, traditionally pursued in VUDA, is truly the way forward?* Instead, the primary research question becomes: *How can we efficiently integrate the prior knowledge derived from LLVMs to adapt a model to the target domain for VUDA?*

This paper presents the first approach in the VUDA literature, which attempts to address this question specifically. In particular, we consider the challenging scenario of Source-Free Video Unsupervised Domain Adaptation (SFVUDA) [44, 46], which consists in the task of adapting an action recognition model trained on a labelled source domain to an unlabelled target domain, without accessing the actual source data. The primary motivation for exploiting LLVMs in SFVUDA is that we expect the generalization capabilities of these models to effectively serve the purpose of mitigating the effects of the *domain shift* existing between the data distributions of the source and the target domain. In particular, since in SFVUDA we do not have access to the source data but *only* the source pre-trained model, direct target adaptation may potentially lead to significantly poor performance, especially when the domain shift is large. On the contrary, the employment of LLVMs can efficiently counteract this negative effect owing to the wide range of visual domains observed during their training process. An additional motivation behind our approach lies in the fact that the rich visual representations derived from LLVMs can efficiently compensate when additional modalities (*e.g.*, optical-flow), which are known to help reduce domain gap in VUDA frameworks [27, 33], are unavailable.

Our approach, which we name as **D**omain **A**daptation with **L**arge **L**anguage-**V**ision models (or DALL-V in short), is a simple yet effective technique to combine the knowledge from the visual representation of a pre-trained LLVM with that derived from the source model and the target data (see Fig. 2c). DALL-V involves a two-stage pipeline: (**i**) in the first stage pseudo-labels are extracted from the CLIP model and are subsequently used for adapting on the target dataset, and (**ii**) in the second stage an ensemble of CLIP, source and target models are used to distill information into a student network. DALL-V introduces very few trainable parameters on top of CLIP and is realized with the help of domain-specific adapters. Despite its simplicity, our approach outperforms SFVUDA and even VUDA methods by a significant margin (+11.8% w.r.t. the best competitor).

In summary, our **contributions** are: (**i**) We show, for the first time in the literature that straightforward zero-shot

methods based on LLVMs vastly outperform the state-of-the-art approaches for SFVUDA. (**ii**) Building upon this observation, we propose DALL-V, a simple approach for SFVUDA which optimally integrates information derived from LLVMs, from a pretrained source model and from unlabelled videos of the target domain. (**iii**) We perform extensive experiments on a total of 20 domain adaptation settings, demonstrating the effectiveness of DALL-V in SFVUDA.

2. Related Work

Video Unsupervised Domain Adaptation. While domain adaptation techniques have been primarily studied in the context of image-level representations and fall under the umbrella of Unsupervised Domain Adaptation (UDA) methods [7, 47, 22, 39, 1, 42], addressing the domain shift problem is even more challenging in the case of videos, due to the additional temporal dimension. Researchers have proposed different strategies. For instance, Chen *et al.* introduced TA^{3}N [1], a discrepancy-based method that aligns domains on the temporal axis by learning a temporal relationship across video sequences. TCoN was introduced in [28], a deep architecture which employs a cross-domain module to compute temporally-aligned source and target feature representations. CO2A [2] proposed an approach based on contrastive learning to align source and target video representations. One major limitation with both UDA and VUDA methods is the fact that they require access to source data. This limits their applicability in real-world scenarios where data may not be accessible due to privacy reasons. Differently, our approach focuses in the more realistic setting where we only have access to the source model.

Source-Free Video Unsupervised Domain Adaptation. Due to the growing concerns related to privacy and data sharing, recently researchers have begun exploring methods Source-Free Unsupervised Domain Adaptation (SFUDA). One of the first works in this direction is SHOT [20], a deep architecture which employs an entropy loss and a classification loss on pseudo-labelled data to adapt the source pretrained network focusing solely on the target data. In a subsequent work [21], the authors introduced an additional auxiliary head to solve the relative rotation task, further improving prediction accuracy. Other works, such as 3C-GAN [18], SFIT [9], and SDDA [16], framed the problem of SFUDA as an image translation task.

In the literature, only a handful of works addressed the problem of Source-Free Video Unsupervised Domain Adaptation (SFVUDA). In particular, ATCoN [44] presented an approach which models temporal consistency across the video sequences. EXTERN [46] proposed to exploit mask-to-mix strategies and video-tailored regularizations for SFVUDA. Our approach radically departs from these previous works as we proposed to exploit LLVMs to adapt to the target video data.

Large Language-Vision Models. The availability of vast web-scale datasets containing image-text pairs [35, 34, 37] have enabled the emergence of novel large multi-modal neural networks which learn joint visual-text embedding spaces [31, 10, 37]. These approaches, commonly referred as Large Language-Vision Models, utilise a separate encoder for each modality and employ a contrastive loss to align the data representations in the feature space [31, 10]. CLIP [31] is a prominent example of such an approach. Despite its simplicity, CLIP have been shown to achieve impressive zero-shot image recognition capabilities. More recently, LLVMs have been also extended with success to the video domains and particularly to the action recognition task [40]. However, we are not aware of previous works which have specifically used LLVMs to address the problem of domain shift in video action recognition.

3. The Unreasonable Effectiveness of LLVM

In this work our goal is to develop a Source-free Video Unsupervised Domain Adaptation (SFVUDA) method that can adapt a source trained model to a target domain of interest. Departing from the traditional SFVUDA approaches, where a multitude of self-supervised losses are optimized on the target domain [44], we pursue an orthogonal and unconventional approach. Our key idea is to leverage a LLVM (*e.g.*, CLIP [31]), which has been pre-trained on web-scale image-text pairs, as a tool for bridging the domain gap.

As a part of a preliminary study we evaluate CLIP in a *zero-shot* manner (see Fig. 2b) on the target dataset. In other words, we simply run inference and evaluate the performance on the **Daily-DA** [44] benchmark using a pre-trained CLIP (see Sec. 4.2 for details on zero-shot inference), and compare with the recent state-of-the-art SFVUDA approaches. The evaluation is carried out on the four target domains of the **Daily-DA**, each of which has eight semantic categories. As showcased in Fig. 1, much to our surprise, CLIP (ViT-B/32) outperforms the best performing SFVUDA method EXTERN [46] by a huge margin of +8.5%, despite CLIP never been explicitly fine-tuned on the target frames, let alone videos. These telling observations from our preliminary study compels us to explore and exploit the *unreasonable effectiveness* of the LLVMs, which is orthogonal to the existing SFVUDA literature.

The implications of undertaking such a pathway to bridge the domain gap in SFVUDA can be several: (**i**) the LLVMs being publicly available, it will help democratize more efficient target adaptation, even if the source data is withheld due to privacy reasons, (**ii**) it disposes off the need to balance and tune complex training objectives, a *de facto* practice in SFVUDA (*e.g.*, ATCoN [44] requires to jointly optimize up to 6 loss functions), (**iii**) it prevents the necessity of ad hoc and specialized network architectures and training objectives to solve the task at hand. Encouraged by

the flexibility offered by LLVMs, we propose DALL-V for SFVUDA that can effectively exploit the *world prior* from the LLVMs and integrate it with complementary sources of information, such as the source trained model. Before we describe our proposed method, we formalize the SFVUDA task and introduce some preliminaries.

4. Methods

4.1. Problem definition and notations

In SFVUDA we are given an *unlabelled* target dataset $\mathcal{D}^{\mathrm{T}} = \{\mathbf{X}_i^{\mathrm{T}}\}_{i=1}^m$, containing m video sequences, and access to a source model $F^{\mathrm{S}}(\cdot)$, trained on a *labelled* source dataset $\mathcal{D}^{\mathrm{S}} = \{\mathbf{X}_i^{\mathrm{S}}, y_i^{\mathrm{S}}\}_{i=1}^n$, which is not available during adaptation on the target. We assume that the target dataset $\mathcal{D}^{\mathrm{T}}$ contains video sequences of actions that share the same label space as the source, *i.e.*, $\mathcal{Y}^S = \mathcal{Y}^{\mathrm{T}} = \{1, ..., C\}$, with C being the number of semantic action categories. We also assume that the source and target marginal distributions are not the same, leading to the so-called *domain-shift* [44].

The goal of any SFVUDA method is to utilize the target dataset and the source-trained model to learn a mapping function (typically using a neural network) that can correctly predict the target samples, *without* needing any access to the source dataset.

4.2. Preliminaries

CLIP [31], which stands for Contrastive Language-Image Pre-training, is composed of two encoders: a vision encoder $G_{\mathrm{V}}(\cdot)$ and a language (or text) encoder $G_{\mathrm{L}}(\cdot)$ (see Fig. 2b). To classify, it associates labels to visual inputs by computing their similarity with a set of textual descriptions. More precisely, given the names of the classes in a dataset (*i.e.*, *drinking*, *walking*, etc.), we construct prompts containing each class name, *e.g.*, "a video of a person [CLS]", where "[CLS]" denotes a class name.

The language model $G_{\mathrm{L}}(\cdot)$ projects the class names into embeddings, denoted as $\{z_c^{\mathrm{L}}\}_{c=1}^C$. On the other hand, assuming a test video of K frames $\mathbf{X} = \{\mathbf{x}_k\}_{k=1}^K$, we extract the visual feature representation for each frame, *i.e.* $\mathbf{Z}^{\mathrm{V}} = \{z_k^{\mathrm{V}}\}_{k=1}^K$ using the vision encoder $G_{\mathrm{V}}(\cdot)$. CLIP outputs a probability distribution, where the probability of the k-th frame to belong to the c-th class is given by:

$$p_c(\mathbf{x}_k) = \frac{\exp(< z_c^{\mathrm{L}}, z_k^{\mathrm{V}} > /\tau)}{\sum_{c'=1}^C \exp(< z_{c'}^{\mathrm{L}}, z_k^{\mathrm{V}} > /\tau)} \quad (1)$$

where τ is a temperature and $< \cdot, \cdot >$ is the inner-product (or cosine similarity) operator. The frame-wise predictions are then aggregated by simply averaging the class probabilities estimated for each frame.

With respect to inference with CLIP, it is regarded as *zero-shot* inference if a given dataset has not been explicitly used for training the CLIP model. In other words,

zero-shot inference refers to the generalization to unseen datasets [31], and must not be confused with generalization to strictly unseen objects [17].

4.3. Domain Adaptation with Large Language-Vision models (DALL-V)

In this work, we propose Domain Adaptation with Large Language-Vision models (DALL-V) as a SFVUDA method. As the name suggests, it leverages the rich world prior from the LLVMs, CLIP in particular, to help mitigate the domain gap. Recall that CLIP has been trained on *image-text* pairs and not *video-caption* pairs. On the other hand, the source model can offer complementary information as it has been trained *supervisedly* on the source videos. Thus, we aspire to jointly leverage these two reservoirs of information in a simple yet effective manner.

Our DALL-V mainly operates in two stages. The first stage, called **Target adaptation** (see Sec. 4.3.1), consists in pseudo-labelling the unlabelled target videos with CLIP and training a target-specific model with the pseudo-labelled data. In this way the target-specific model can benefit from the rich general purpose knowledge of CLIP and at the same time discover patterns that are inherent to the target domain. The second stage, called **Ensemble distillation** (see Sec. 4.3.2), focuses on *ensembling* the source model and target model into a smaller network, with knowledge distillation [8], which is then finally used during inference. Before we dive into the details of our proposed method, we describe how we carry out source training.

Source Pre-Training. Opposed to the previous work AT-CoN, which fine-tunes all the source model weights using the source dataset, we propose an alternative approach. To prevent the source model to be biased towards the source dataset, we propose to initialize the source model with CLIP pre-trained weights, and fine-tune only a small set of parameters. To fine-tune CLIP on the source data, we introduce an adapter network [6], a small multi-layer perceptron network, that is appended on top of the vision encoder of CLIP. Source pre-training essentially consists in updating the weights of the adapter using a supervised training objective, while keeping the CLIP vision encoder frozen.

Formally, the vision encoder of CLIP $G_{\mathrm{V}}(\cdot)$ is extended with an adapter $A(\cdot) : \mathcal{R}^d \to \mathcal{R}^d$ on top, where d is the output dimension of $G_{\mathrm{V}}(\cdot)$. We construct textual descriptions t for the video sequences using a set of 16 manually-designed prompts [40], containing templates such as "a video of [CLS]", "can you recognize the action of [CLS]?", and so on, with y being the class name [CLS] (refer to the Supplement for the complete list of prompts).

We adopt the training objective of ActionCLIP [40] to train the source model. In details, we first compute the ground-truth similarity scores $q(\mathbf{x})$ for each input frame $\mathbf{x}$,
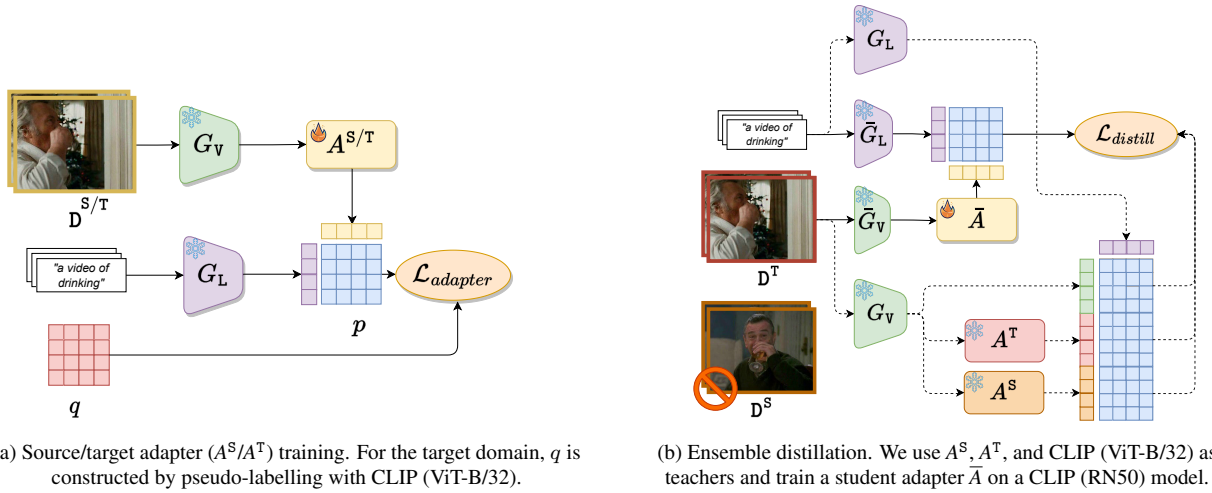
(a) Source/target adapter ($A^{\text{S}}/A^{\text{T}}$) training. For the target domain, q is constructed by pseudo-labelling with CLIP (ViT-B/32).

(b) Ensemble distillation. We use A^{S}, A^{T}, and CLIP (ViT-B/32) as teachers and train a student adapter $\bar{A}$ on a CLIP (RN50) model.

Figure 3: Overview of the pipeline of DALL-V. The 🔥 means the network is trainable, while the ❄ means the network is frozen. We denote with $\overline{bar}$ the student components to distinguish them from the teacher.

with 0 in the positions corresponding to negative pairs in the mini-batch and 1 for positive pairs, where a pair denotes the *frame-prompt* tuple. To train the network (essentially just the adapter A^{S}; language and vision encoders being frozen) we use the Kullback-Leibler (KL) divergence loss between $\mathbf{p}(\mathbf{X})$ (see Eq. (1)) and $\mathbf{q}(\mathbf{X})$:

$$\mathcal{L}^{\text{S}}_{\text{adapter}} = \mathbb{E}_{(\mathbf{X},t)\sim\mathcal{D}^{\text{S}}}[\text{KL}(\mathbf{p}(\mathbf{X}), \mathbf{q}(\mathbf{X}))] \qquad (2)$$

4.3.1 Target adaptation

To adapt CLIP on the target domain, we adopt a similar approach to that outlined for the source domain, with one key difference: the absence of ground-truth labels for the target domain. While it is a common practice among the source-free methods to utilize the source model to pseudo-label the unlabelled target data [20], we refrain from such an approach, and instead utilize the original CLIP model (*i.e.*, *without* any source adapter A^{S}) for pseudo-labelling. Despite learning fewer parameters with the adapter, the source model can still be biased towards the source, and hence, yielding noisy pseudo-labels.

To pseudo-label (PL), we utilize zero-shot CLIP inference (described in Sec. 4.2), with the same collection of textual descriptions used during the source fine-tuning stage, to assign a PL to each sample in the target domain. To this end, we use a confidence threshold for filtering out unreliable PLs (see Supplement for details). After the thresholding, we end up with a subset $\tilde{\mathcal{D}}^{\text{T}} = \{\mathbf{X}^{\text{T}}_i\}^{m'}_{i=1} \subset \mathcal{D}^{\text{T}}$, containing PLs as $\{\tilde{y}_i\}^{m'}_{i=1}$, with m' denoting the filtered samples.

Following the source fine-tuning procedure, we append an adapter $A^{\text{T}}(\cdot) : \mathcal{R}^d \to \mathcal{R}^d$ to the vision encoder of CLIP and minimize the following loss, while keeping CLIP frozen:

$$\mathcal{L}^{\text{T}}_{\text{adapter}} = \mathbb{E}_{(\mathbf{X},t)\sim\tilde{\mathcal{D}}^{\text{T}}}[\text{KL}(\mathbf{p}(\mathbf{X}), \mathbf{q}(\mathbf{X}))] \qquad (3)$$

4.3.2 Ensemble distillation

With three different models at our disposal: the source, target and the original CLIP, we can distill information from these three models into a single network. Knowledge Distillation [8] presents two important advantages: (**i**) allows to generate more reliable and informative pseudo-labels, and (**ii**) distillation into a smaller network reduces inference time without sacrificing much performance. For this second point, we explicitly use CLIP ViT-B/32 as teacher (and as backbone for the source and target adapters) and CLIP RN50 as student. Since we use an ensemble of three models to distill information, we call this step *ensemble distillation*. Now, with predictions available from three different models, the question is how to distill them together. A simple approach could be to average the three probability distributions as follows:

$$\mathbf{p}^{\text{ens}}(\mathbf{X}) = \frac{1}{3}(\underbrace{\mathbf{p}(\mathbf{X})}_{\text{ZS CLIP}} + \underbrace{\mathbf{p}^{\text{S}}(\mathbf{X})}_{\text{Source adapter}} + \underbrace{\mathbf{p}^{\text{T}}(\mathbf{X})}_{\text{Target adapter}}) \qquad (4)$$

Whereas, another approach could consist in carrying out majority voting, where each of the three heads votes for a class, and the class with two or more votes becomes the new PL. Following [8], we utilize both kinds of PLs for distillation, using a standard multi-class classification loss (or cross-entropy loss $\mathcal{L}_{\text{CE}}$) and a discrepancy loss. The discrepancy loss corresponding to the *ensembled* PL in Eq. (4) is given as:

$$\mathcal{L}_{\text{discrepancy}} = \mathbb{E}_{(\mathbf{X},t)\sim\mathcal{D}^{\text{T}}}[\text{KL}(\frac{\mathbf{p}^{\text{student}}(\mathbf{X})}{\tau}, \frac{\mathbf{p}^{\text{ens}}(\mathbf{X})}{\tau})] \qquad (5)$$

where $\mathbf{p}^{\text{student}}$ denotes the student network's probability distribution, which is matched with that of $\mathbf{p}^{\text{ens}}$.

We then balance the contribution of the two losses as:

$$\mathcal{L}_{distill} = \alpha * \mathcal{L}_{ce} + (1 - \alpha) * \mathcal{L}_d \quad (6)$$

where α is set according to the value proposed in [8]. Note that, we do not fine-tune all the student weights, but instead append an adapter on top of the frozen backbone and fine-tune only the adapter $\overline{A}$ using Eq. (6).

5. Experiments

5.1. Experimental setup

Datasets and settings. We present an extensive experimental evaluation on three standard benchmarks for SFVUDA. In particular, we report results on **Daily-DA** [44], which comprises 18,949 videos from 8 classes, and it is built from 4 original video action recognition datasets, namely HMDB51 [15], ARID [43], MIT [26] and Kinetics [12]. Additionally, we evaluate our framework on **UCF-HMDB**$_{full}$ [1], a benchmark comprising 3,209 videos divided in 12 action categories from the HMDB51 [15] and UCF101 [38] action recognition datasets. Note that the first benchmark, **Daily-DA**, poses a more significant challenge when compared to the latter since it comprises videos with very different lighting conditions across domains. Last, we test our method on **Sports-DA** [45], a benchmark consisting of three datasets (*i.e.*, UCF101 [38], Sports-1M [11], and Kinetics [12]), with 40,718 videos and 23 classes.

Implementation details. We implement our adapters as two-layers perceptrons, following [6]. In accordance with [40], we use AdamW optimizer [24] with a learning rate of 0.01 and a weight decay of 0.2. The temperature τ is set to 2.0 for all datasets. We trained all our models for 30 epochs. For a single experiment, we have used either 4 Tesla V100 or 2 RTX A6000 GPUs. Further implementation details can be found in the Supplement.

Baselines and competitors. We compare our method with a selection of standard baseline methods for VUDA and SFVUDA, which include TRN [49], DANN [5], MK-MDD [23], TA^{3}N [1], SFDA [13], SHOT [20], SHOT++ [21], MA [19], BAIT [48] and CPGA [30]. Additionally, we report the scores of the state-of-the-art competitors ATCoN [44] and EXTERN [46]. We also report the results for a set of CLIP-based baselines of our design to provide a more solid context for assessing our performance. In particular, CLIP (RN50) and CLIP (ViT-B/32) indicate the scores obtained in a zero-shot fashion with the CLIP model, with ResNet-50 and ViT-B/32 as backbones, respectively. Finally, we report the lower and upper bounds obtained by the source-only and target-supervised models, respectively.

5.2. Comparison with the State-of-the-Art

We report our experimental validation scores in Tab. 1, Tab. 2, and Tab. 3, obtained on the **Daily-DA**, the **UCF-HMDB**$_{full}$, and the **Sports-DA** benchmarks respectively. For all the benchmarks, the lower bound refers to the *out-of-the-box* performance of our source-trained model on the target data, *i.e.*, the CLIP backbone and the source adapter A^S. Analogously, the upper bound is the oracle performance obtained after training *supervisedly* on the target data, *i.e.*, assuming knowledge of the target labels.

Regarding the more challenging **Daily-DA** dataset, it is possible to observe in Tab. 1 that our proposed distillation method DALL-V (RN50) obtains the best average score across the 12 settings among the competitors and baselines sharing the same backbone, with a 5% gain on the second best score (CLIP (RN50)). It is also visible that the best score is achieved in 9 out 12 total settings, proving our proposed distillation-based method to be significantly effective when addressing the SFVUDA task. Notably, it emerges that the scores achieved when addressing the ARID dataset ($\rightarrow$A) as target domain are visibly lower than those reported for the other settings. We associate this behavior to the challenging visual features of this particular benchmark, characterized by being shot in a low illumination environment.

As for the small-scale **UCF-HMDB**$_{full}$ dataset, we can observe significantly higher absolute accuracy scores on both the involved SFVUDA directions, indicating that the benchmark is significantly closer to saturation for this task. Nonetheless, it emerges from the table that our proposed method achieves a state-of-the-art average accuracy score across the two datasets (+0.6% on the best competitor) and the best score for the HMDB$\rightarrow$UCF direction (+1.2% on the second best).

Finally, as shown in Tab. 3, our method achieves comparable results with EXTERN [46] on **Sports-DA**, surpassing it on three settings, and achieving the same results on another. On the two settings with UCF101 as target (*i.e.*, K$\rightarrow$U and S$\rightarrow$U), our method falls behind by approximately 6 percentage points, resulting on an average score across the six scenarios slightly lower than the state-of-the-art (*i.e.*, -0.9%). However, DALL-V still achieves an average accuracy of +8.5% w.r.t. ATCoN, further demonstrating the effectiveness of our approach for SFVUDA. Interestingly, on **Sports-DA**, our lower bound already achieves competitive results with other SFVUDA methods, surpassing ATCoN by +2.7% on average. The table additionally shows that CLIP zero-shot is an extremely strong baseline on the benchmark, achieving comparable results with EXTERN, i.e., -2.0% with RN50, and $+2.6\%$ with the ViT-B backbone.

	Method	Accuracy (%)											
		K→A	K→H	K→M	M→A	M→H	M→K	H→A	H→M	H→K	A→H	A→M	A→K	Avg.
	TRN [49]	20.9	36.7	29.0	22.1	43.7	53.1	13.8	22.0	37.1	17.2	14.7	24.4	27.9
	Lower bound	15.6	47.9	35.7	34.7	44.6	61.6	17.5	25.5	45.1	14.6	15.5	17.8	31.3
ZS	CLIP (RN50) [31]	<u>30.5</u>	50.0	42.2	<u>30.5</u>	50.0	62.9	<u>30.5</u>	42.2	62.9	50.0	42.2	62.9	46.4
ZS	CLIP (ViT-B/32) [31]	**31.3**	49.6	46.0	**31.3**	49.6	65.4	**31.3**	46.0	65.4	49.6	**46.0**	65.4	48.1
UDA	DANN [5]	21.2	37.5	21.7	22.8	43.3	58.8	14.2	29.5	38.2	20.1	19.7	27.0	29.5
UDA	MK-MDD [23]	21.7	36.2	24.0	21.0	50.4	58.5	20.3	25.7	33.8	18.7	18.0	26.1	29.5
UDA	TA³N [1]	19.9	37.7	31.5	21.6	43.0	55.5	14.4	25.7	38.4	14.9	15.6	23.4	28.5
SFDA	SFDA [13]	12.6	44.9	27.5	16.0	35.2	49.2	13.1	24.2	24.9	16.3	13.2	25.2	25.2
SFDA	SHOT [20]	12.0	44.6	29.5	15.3	36.7	51.0	13.6	24.2	21.2	17.1	14.0	24.3	25.3
SFDA	SHOT++ [21]	12.6	40.8	28.7	14.9	41.7	46.3	16.0	22.2	33.1	15.4	12.5	21.8	24.4
SFDA	MA [19]	12.8	45.8	30.0	17.7	37.4	53.5	12.9	25.0	22.2	16.7	15.2	24.3	26.1
SFDA	BAIT [48]	12.7	45.7	30.0	16.9	39.6	53.0	13.6	25.5	21.2	15.7	14.5	25.5	26.2
SFDA	CPGA [30]	13.1	46.0	30.7	18.1	39.2	55.1	13.1	26.2	25.5	19.2	16.5	26.7	26.5
SFVUDA	ATCoN [44]	17.2	48.2	32.5	27.2	47.3	57.7	17.9	30.7	48.5	26.7	17.2	31.0	33.5
SFVUDA	EXTERN [46]	23.9	<u>55.8</u>	35.2	18.1	53.7	68.1	26.2	40.7	57.6	26.2	18.2	51.4	39.6
SFVUDA	DALL-V (ours)	24.0	52.5	**47.0**	24.0	**65.4**	**78.1**	24.0	**47.0**	**76.7**	**57.9**	**45.7**	**75.0**	**51.4**
	Upper bound	26.9	70.4	61.5	26.9	70.4	88.9	26.9	61.5	88.9	70.4	61.5	88.9	61.9

Table 1: Validation accuracy for ***Daily-DA***. **Bold** indicates best, <u>underline</u> represents best with same backbone as baseline (*i.e.* ResNet50). Lower bound indicates a source adapter and Upper bound a target adapter, both trained supervised on CLIP (RN50).

	Method	Accuracy (%)		
		H→U	U→H	Avg.
	TRN [49]	72.8	72.1	72.4
	Lower bound	71.6	76.1	73.8
ZS	CLIP (RN50) [31]	81.0	86.0	83.5
ZS	CLIP (ViT-B/32) [31]	90.3	**89.1**	89.7
UDA	DANN [5]	74.4	75.1	74.8
UDA	MK-MMD [23]	74.7	79.7	77.2
UDA	TA³N [1]	78.1	84.8	81.5
SFDA	SFDA [13]	69.8	75.0	72.4
SFDA	SHOT [20]	74.4	74.4	74.4
SFDA	SHOT++ [21]	71.1	68.1	69.6
SFDA	MA [19]	74.4	67.3	70.9
SFDA	BAIT [48]	75.3	76.3	75.8
SFDA	CPGA [30]	75.8	68.1	72.0
SFVUDA	ATCoN [44]	85.3	79.7	82.5
SFVUDA	EXTERN [46]	91.9	<u>88.9</u>	90.4
SFVUDA	DALL-V (ours)	<u>**93.1**</u>	88.9	<u>**91.0**</u>
	Upper bound	93.7	91.4	92.6

Table 2: Validation accuracy for ***UCF-HMDB***$_{full}$. **Bold** indicates best, <u>underline</u> represents best with same backbone as baseline (*i.e.* ResNet50). Lower bound indicates a source adapter and Upper bound a target adapter, both trained supervised on CLIP (RN50).

5.3. Ablation analysis

We report in Tab. 4 the intermediate accuracy scores achieved at different steps of our proposed source-free pipeline for the ***Daily-DA*** and ***UCF-HMDB***$_{full}$. In particular, we report the performance of the *source supervised* model and the *target unsupervised* one, also reporting the *final* model and the *zero-shot* version of CLIP for comparison. Finally, we report the scores obtained by ensembling the predictions of the three aforementioned models. The A^S and A^T notation indicates the usage of the source and target adapters on top of the frozen CLIP model, that are trained on the respective domains as described in detail in 4.3. Next we discuss the individual contributions. Additional ablation analyses are available in the Supplement.

Effectiveness of finetuning LLVM on the target. Tab. 4 clearly shows that fine-tuning our model in an unsupervised fashion on the target domain is significantly effective with respect to the target validation accuracy. The tables show indeed a gain of 2.0% on ***Daily-DA*** and 1.1% on ***UCF-HMDB***$_{full}$. Additionally, it is possible to observe a further gain on ***Daily-DA*** when ensembling the target model (which gains +2.7% by itself) with CLIP, achieving a +3.6%. From these results, it emerges that fine-tuning LLVMs on the target domain, even in an unsupervised manner, is effective for the SFVUDA task.

Effectiveness of ensembles on LLVM and domain-specific networks. In the last line of Tab. 4, it is possible to observe that ensembling predictions from the three distinct models turns out to yield the best average target accuracy score on both the considered benchmarks, resulting in a gain of +5.7% on ***Daily-DA*** and of +2.9% on ***UCF-HMDB***$_{full}$ w.r.t. the CLIP baseline. These results and those reported in the previous two paragraphs represent further proof that the three models are effectively complementary, and each contributes with additional discriminative knowledge to the resulting model. Conversely, CLIP positively contributes to

	Method	Accuracy (%)						
		K→U	K→S	S→U	S→K	U→K	U→S	**Avg.**
	TRN [49]	86.4	66.9	85.3	71.0	49.3	43.3	67.0
	Lower bound	85.4	79.5	84.4	78.2	67.2	64.3	76.5
ZS	CLIP (RN50) [31]	83.4	<u>79.9</u>	83.4	80.4	80.4	<u>79.9</u>	81.2
ZS	CLIP (ViT-B/32) [31]	90.0	**82.4**	90.0	**85.1**	**85.1**	**82.4**	**85.8**
UDA	DANN [5]	88.0	75.0	85.7	73.4	65.9	55.1	73.8
UDA	MK-MMD [23]	90.2	67.9	90.9	73.6	66.1	55.6	74.0
UDA	TA³N [1]	90.3	68.6	93.0	72.6	63.6	54.1	73.7
SFUDA	SFDA [13]	86.1	60.0	85.4	68.0	55.8	43.6	66.5
SFUDA	SHOT [20]	91.2	64.9	88.8	72.0	53.9	43.6	69.1
SFUDA	SHOT++ [21]	90.0	63.1	88.0	70.3	44.7	40.9	66.2
SFUDA	MA [19]	91.0	65.9	87.8	71.9	60.7	39.4	69.5
SFUDA	BAIT [48]	92.3	66.6	88.3	72.8	57.2	44.7	70.3
SFUDA	CPGA [30]	89.4	66.3	86.5	72.5	55.2	44.5	69.1
SFVUDA	ATCoN [44]	93.6	69.7	90.6	76.0	65.2	47.9	73.8
SFVUDA	EXTERN [46]	<u>93.7</u>	73.8	<u>95.4</u>	82.2	<u>81.2</u>	72.7	<u>83.2</u>
SFVUDA	DALL-V (ours)	88.0	77.7	88.8	<u>82.3</u>	<u>81.2</u>	75.9	82.3
	Upper bound	93.4	88.3	93.4	85.6	85.6	88.3	89.1

Table 3: Validation accuracy for ***Sports-DA***. **Bold** indicates best, <u>underline</u> represents best with same backbone as baseline (*i.e.* ResNet50). Lower bound indicates a source adapter and Upper bound a target adapter, both trained supervised on CLIP (RN50).

	Accuracy (%)							
Method	Daily-DA					UCF-HMDB$_{full}$		
	K→Any	M→Any	H→Any	A→Any	**Avg.**	H→U	U→H	**Avg.**
CLIP	42.3	48.7	47.6	53.7	48.1	90.3	89.1	89.7
A^S	40.0 [-2.3]	55.7 [+7.0]	35.3 [-12.4]	35.0 [-18.7]	41.5 [-6.6]	91.0 [+0.7]	80.5 [-8.6]	85.8 [-3.9]
A^T	44.4 [+2.1]	51.6 [+2.9]	48.5 [+0.9]	58.4 [+4.7]	50.8 [+2.7]	90.5 [+0.2]	91.1 [+2.0]	90.8 [+1.1]
CLIP + A^S	45.1 [+2.8]	**57.8** [+9.1]	40.0 [-7.6]	50.8 [-2.9]	48.4 [+0.3]	93.1 [+2.8]	87.2 [-1.9]	90.2 [+0.5]
CLIP + A^T	43.8 [+1.5]	50.9 [+2.2]	48.2 [+0.6]	57.4 [+3.7]	51.7 [+3.6]	89.3 [-1.0]	**91.9** [+2.8]	90.6 [+0.9]
CLIP + A^S + A^T	**46.2** [+3.9]	57.3 [+8.6]	**49.5** [+1.9]	**62.1** [+8.4]	**53.8** [+5.7]	**94.9** [+4.6]	90.3 [+1.2]	**92.6** [+2.9]

Table 4: Ablation on the relative improvement of the different modules of our proposed DALL-V framework w.r.t. CLIP (ViT-B/32) on both considered benchmarks. **Bold** indicates best. The + operator indicates ensembling. Due to limited space, results are aggregated by source dataset.

the final score thanks to its well-known ability to generalize to unseen domains in a zero-shot fashion; on the other hand, the tuning process of domain-specific adapters on the source and target domains, although not sufficient by itself, proves beneficial thanks to its ability to capture specific features of the involved domains and the associated label sets. Note that we still opted for distilling this knowledge to a ResNet-based model in order to be directly comparable with the competitors.

Effectiveness of using multiple templates. As a further insight into the individual contributions of the different components of our framework, we propose in Fig. 5, a simple analysis on how the number of templates for the text input at inference time impacts on the accuracy score on the target validation set of the two considered benchmarks. We observe that increasing the number of templates from just

a single one to 12 (following ActionCLIP [40]) is significantly beneficial with respect to the final accuracy score. For even higher number of templates, the performance of our model reaches a plateau. Note that for the choice of the number of template we opted for following [40].

Modeling the feature space. We plot in Fig. 4 an UMAP [25] chart of the features produced by the zero-shot CLIP (RN50) model, the source model, and our DALL-V model, on the ***UCF-HMDB***$_{full}$ benchmark, showing the features belonging to different classes in different colors. The chart clearly highlights the ability of the proposed DALL-V method to produce well-separated clusters for different action categories, thanks to the effective combination of the complementary knowledge of intermediate models.
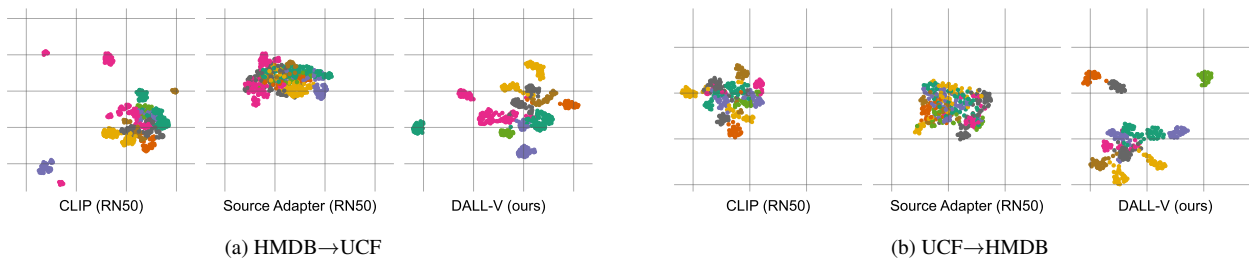
(a) HMDB→UCF

(b) UCF→HMDB

Figure 4: UMAP visualisation of the feature space on *UCF-HMDB*$_{full}$ for the zero-shot CLIP (RN50), source and final DALL-V models. The chart shows the ability of our proposed DALL-V to efficiently cluster the action categories in the features space by combining knowledge from the other two models.
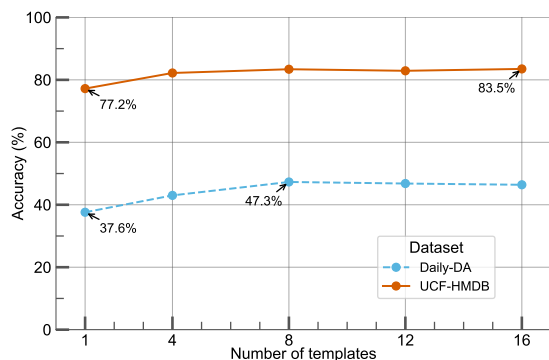

Figure 5: Sensitivity study on the number of templates used for the text input w.r.t. the accuracy score, on *Daily-DA* and *UCF-HMDB*$_{full}$.

Limitations

Note that CLIP has been trained on web-crawled image-text pairs that are not made publicly available by OpenAI [35]. Due to its black-box nature, equally trusting CLIP predictions, as much as the source model, may elicit unpredictable behaviour, especially in safety critical applications. Thus, appropriate caution must be exercised. Moreover, ours being an empirical work, we can not provide any theoretical guarantee if using LLVMs, such as CLIP, will always result in better performance over traditional SFVUDA approaches. Thus, as future work, we plan to extensively evaluate using more open-source alternatives.

6. Conclusions

We presented DALL-V, a SFVUDA method based on a simple but novel approach to SFVUDA driven by the intuition of combining the complementary information derived from domain-specific simple models and the powerful CLIP-based LLVMs trained on world knowledge. We provided motivation for such approach, and reported an extensive evaluation on three standard benchmarks for VUDA, purposed in our case for the source-free scenario. We compare our performance with existing methods, as well as with a selection of CLIP-based baselines, showing that our proposed frameworks achieves state-of-the-art performance on both considered benchmarks.

References

[1] 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. 1, 3, 6, 7, 8

[2] Victor G Turrisi da Costa, Giacomo Zara, Paolo Rota, Thiago Oliveira-Santos, Nicu Sebe, Vittorio Murino, and Elisa Ricci. Dual-head contrastive domain adaptation for video action recognition. In *WACV*, 2022. 1, 3

[3] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *CVPR*, 2017. 1

[4] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 1

[5] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2015. 6, 7, 8

[6] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv*, 2021. 4, 6

[7] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016. 3

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, 2015. 4, 5, 6

[9] Yunzhong Hou and Liang Zheng. Visualizing adapted knowledge in domain transfer. In *CVPR*, 2021. 3

[10] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 3

[11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 6

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 6

[13] Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. Domain adaptation without source data, 2020. 6, 7, 8

[14] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *IJCV*, 2022. 1

[15] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, 2011. 6

[16] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *WACV*, 2021. 3

[17] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 4

[18] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 3

[19] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, 2020. 6, 7, 8

[20] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020. 3, 5, 6, 7, 8

[21] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *TPAMI*, 44(11):8602–8617, 2021. 3, 6, 7, 8

[22] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 3

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks, 2015. 6, 7, 8

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[25] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. 8

[26] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding. *TPAMI*, 42(2):502–508, 2019. 6

[27] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *CVPR*, 2020. 2

[28] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In *AAAI CAI*, 2020. 1, 3

[29] Preksha Pareek and Ankit Thakkar. A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review*, 2021. 1

[30] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation, 2021. 6, 7, 8

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 4, 7, 8

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[33] Aadarsh Sahoo, Rutav Shah, Rameswar Panda, Kate Saenko, and Abir Das. Contrast and mix: Temporal contrastive video domain adaptation with background mixing. *NeurIPS*, 2021. 2

[34] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv*, 2022. 3

[35] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv*, 2021. 1, 3, 9

[36] Javier Selva, Anders S Johansen, Sergio Escalera, Kamal Nasrollahi, Thomas B Moeslund, and Albert Clapés. Video transformers: A survey. *arXiv*, 2022. 1

[37] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022. 1, 3

[38] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. 6

[39] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Artificial Intelligence*, volume 30, 2016. 3

[40] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv*, 2021. 2, 3, 4, 6, 8

[41] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, 2021. 2

[42] Yuecong Xu, Haozhi Cao, Kezhi Mao, Zhenghua Chen, Lihua Xie, and Jianfei Yang. Aligning correlation information for domain adaptation in action recognition. *Neural Networks and Learning Systems*, 2022. 3

[43] Yuecong Xu, Jianfei Yang, Haozhi Cao, Kezhi Mao, Jianxiong Yin, and Simon See. Arid: A new dataset for recognizing action in the dark. In *International Workshop on Deep Learning for Human Activity Recognition*, pages 70–84. Springer, 2021. 6

[44] Yuecong Xu, Jianfei Yang, Haozhi Cao, Keyu Wu, Min Wu, and Zhenghua Chen. Source-free video domain adaptation by learning temporal consistency for action recognition. In *ECCV*, 2022. 2, 3, 4, 6, 7, 8

[45] Yuecong Xu, Jianfei Yang, Haozhi Cao, Keyu Wu, Min Wu, Rui Zhao, and Zhenghua Chen. Multi-source video domain adaptation with temporal attentive moment alignment. *arXiv*, 2021. 6

[46] Yuecong Xu, Jianfei Yang, Haozhi Cao, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Leveraging endo- and exo-temporal regularization for black-box video domain adaptation, 2022. 2, 3, 6, 7, 8

[47] Jinyu Yang, Weizhi An, Sheng Wang, Xinliang Zhu, Chaochao Yan, and Junzhou Huang. Label-driven reconstruction for domain adaptation in semantic segmentation. In *ECCV*, 2020. 3

[48] Shiqi Yang, Yaxing Wang, Joost Weijer, and Luis Herranz. Unsupervised domain adaptation without source data by casting a bait, 2020. 6, 7, 8

[49] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos, 2018. 6, 7, 8

[50] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv*, 2020. 1