Author:
Munro, Jonathan P N; Damen, Dima

Title:
Unsupervised domain adaptation for fine-grained action understanding

Unsupervised Domain Adaptation for Fine-grained Action Understanding

By

JONATHAN MUNRO

Department of Computer Science
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

SEPTEMBER 2021

Word count: 46353

Abstract

Fine-grained actions are short actions that typically last a matter of seconds. However, due to the difficulty in collecting and annotating fine-grained actions, datasets contain limited variability in their collection. Many datasets are collected in a single environment, with few participants and cameras. Computer vision models trained on such datasets may not perform well on videos encountered when they are deployed. This work showcases several domains shifts in the large-scale dataset of fine-grained actions, EPIC-KITCHENS.

This thesis focuses on unsupervised domain adaptation for fine-grained action understanding. This assumes there is a domain shift between the labelled videos used for training (the source domain) and videos used for testing (the target domain). With access to unlabelled videos from the target domain, the aim is to improve the performance of fine-grained action understanding tasks. Unsupervised domain adaptation reduces the high cost of annotating fine-grained actions, which is often expensive or impractical in the target domain.

Videos depicting fine-grained actions contain both visual and motion information, as well as audio and often textual descriptions. This work explores utilising these multiple modalities to improve domain adaptation, as well as learn a representation of fine-grained actions. Some modalities will be more robust than others to different domain shifts, for example motion is more robust than RGB to environmental changes. A domain adaptation solution is proposed which improves action recognition performance by exploiting the differing level of robustness of video modalities to domain shifts. Additionally, cross-modal tasks can be used to learn discriminative information about fine-grained actions. A domain adaptation solution is proposed to adapt a text-to-video retrieval system to a novel set of uncaptioned videos.

Acknowledgements

Most importantly, I would like to express my deepest gratitude to my supervisor, Dima, for her generous support and guidance throughout my PhD. I cannot thank her enough. I would also like to thank Diane and Gabriela from NaverLabs for their guidance and mentoring.

Thanks to everyone in the VI-Lab, including, but not limited to: Will, Vangelis, Mike, Hazel, Davide, Jian, Fae, Abel, Farnoosh, Xingrui, Laurie, Sam, Jay, Erik, Richard, Toby, Zeynel, Sasha, Eduardo, Young, Ramon and Perla, for creating such a fun and friendly environment to work in.

Finally, I would like to thank Grace for her support and patience throughout my PhD, as well as my parents and my sister.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

Publications

The work described in this thesis contributed to the following publications:

1. Scaling egocentric vision: The EPIC-KITCHENS dataset
   Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, Michael Wray
   Proceedings of the European Conference on Computer Vision (ECCV)
   2018

2. Multi-modal domain adaptation for fine-grained action recognition
   Jonathan Munro, Dima Damen
   Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
   2020

3. The EPIC-KITCHENS dataset: Collection, challenges and baselines
   Dima Damen, Hazel Doughty, Giovanni Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, Michael Wray
   IEEE Transactions on Pattern Analysis & Machine Intelligence (TPAMI)
   2020

4. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100
   Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, Michael Wray
   International Journal of Computer Vision (IJCV)
   2021

Contents

List of Tables

List of Figures

Acronyms

AMT Amazon Mechanical Turk

BN Batch Normalisation

CNN Convolutional Neural Network

GMM Gaussian Mixture Module

GRL Gradient Reversal Layer

HMDB Human Motion DataBase

IoU Intersection over Union

iDT improved Dense Trajectories

RGB Red, Green Blue (colours)

RNN Recurrent Neural Network

mAP Mean Average Precision

MMD Maximum Mean Discrepancy

MLP Multi-Layer Perceptron

nDCG normalised Discounted Cumulative Gain

PCA Principle Component Analysis

RKHS Reproducing Kernel Hilbert Space

SGD Stochastic Gradient Descent

SVM Support Vector Machine

UDA Unsupervised Domain Adaptation

UMAP Uniform Manifold Approximation and Projection

Chapter 1

Introduction

Fine-grained actions describe what a human is doing over a short time-interval. These last a matter of seconds, such as “cutting an onion” or “picking up tomato”, in contrast to coarse-grained actions or activities which span longer periods of time, such as “preparing a meal”. Understanding what fine-grained actions occur in a video has a wide range of applications in assistive technologies in homes as well as in industry.

Deep learning has advanced the performance of many video understanding tasks. However, neural networks require a massive amount of annotated videos in order to train discriminative models. For activity recognition, large internet databases has enabled datasets to scale [25, 26]. The performance of recent methods on the Kinetics datasets exceed 80% accuracy [27, 28, 29], but this is only possible due to the 500,000 annotated videos available for training. Unfortunately, the collection of fine-grained actions has increased difficulty, as they are not easily searchable in online sources.

For fine-grained action understanding, datasets often collect long-untrimmed videos [11, 12, 13, 15, 30]. A single video provides many examples of fine-grained actions, but leads to a large bias towards a limited number of participants, locations and cameras [11, 12, 13, 15, 30]. In addition, participants conduct a limited number of activities, which are sometimes pre-specified before data collection [8, 9, 11, 12, 13, 31]. This is in contrast to video collected from the web, where each video is filmed in a different location with a different camera, depicting a large variety of different activities.

The assumption in classical machine learning is that the training data is representative of the test data. In practice, this is rarely the case. A domain shift occurs when the distribution of training videos is different to the videos encountered during testing. Given the large bias towards different recording conditions in datasets of fine-grained actions, a domain shift is likely to be a common problem for real-world applications. Although dataset bias has been extensively studied in the context of images [32, 33], and some focus on activities in video [34, 35], the bias within datasets of fine-grained actions has not been explored. This is surprising as datasets of fine-grained actions contain fewer and less diverse number of sources than large-scale datasets for image or activity recognition.

Neural networks trained on fine-grained actions may exhibit poor performance when evaluated under a domain shift [33]. One solution is to collect footage that better matches the recording conditions used for testing. However, the annotation of a large number of fine-grained actions is expensive. Unsupervised domain adaptation reduces the cost of annotation by exploiting pre-existing labelled data, the source domain, in addition to the unlabelled videos collected from the test distribution, the target domain. The difficulty is that videos in the target domain are sampled from a different distribution to that in the source domain.

1.1 Challenges and Research Hypothesis

The vast majority of the domain adaptation literature has focused on images, whereas video introduces additional challenges and opportunities. Video is multi-modal, containing both spacial and temporal information in addition to other modalities such as audio. Recent domain adaptation works focused on overcoming a domain shift in the temporal dimension [1, 2], however, domain adaptation is still applied to individual modalities. This thesis explores how to utilise multiple modalities in video for domain adaptation. Chapter 4, shows that the differing degrees of robustness of modalities to domain shifts can be exploited for domain adaptation.

Designing a domain adaptation benchmark to reflect real world conditions is a challenge. Current video domain adaptation benchmarks select source and target domains from different video datasets [1, 2, 36, 37, 38]. The downside is that the target domain is still collected with training computer vision models in mind. Video understanding models deployed in the real world contain will encounter actions that are irrelevant to the task, a natural long-tailed distribution of classes and untrimmed video. This is in contrast to computer vision benchmarks that carefully select videos from chosen classes, often with a similar number of videos for each class. Many fine-grained action recognition datasets, including EPIC-KITCHEN [10, 30], contain videos of continuous footage of participants over long periods of time. The unscripted nature of collecting video is more realistic scenario for domain adaptation, as collection of footage has little bias towards the sampling of actions for the target task. Chapter 5 explores some challenge of creating a domain adaptation benchmark for EPIC-KITCHENS, including how to validate and tune hyper-parameters in the absence of labelled target data.

Domain adaptation for EPIC-KITCHENS [10, 30] has additional challenges beyond other video domain adaptation benchmarks [1, 2, 36, 37, 38], including: the natural long-tailed distribution of classes, differing class proportions between domains, as well as unseen actions not present in the source domain. The most common approach for domain adaptation is to align the distributions of examples in source and target domains, however, this does not take into account the aforementioned challenges. In Chapter 6, we propose to align videos depicting the same actions across domains, instead of marginal distributions. The class balanced loss helps alleviate issues with long-tailed distributions and differing class proportions between domains. We also propose a sampling approach for target videos, to avoid aligning actions in the target domain that do not occur in the source.

The task of action recognition introduces additional challenges for domain adaptation. Verbs are generally ambiguous with large overlaps between classes [39]. A single action often has many different ways to describe it. For example, the same video with the caption *add salt to water*, could also be captioned as *season water* or *sprinkle salt into water*. This is refereed to as a semantic shift, which could lead to a domain shift for action labels. This thesis assumes there is no semantic shift between the source and target domains, and only a visual shift occurs.

Another important challenge not tackled in this thesis is that the target domain contain untrimmed video, depicting many different actions. This thesis assumes the existence of an action detector that performs well enough on target videos to segment actions into individual video snippets. Therefore, the task of thesis is to associate each unlabelled video snippet in the target domain with an action.

1.1.1 Research Hypothesis

- A domain shift between videos in the source and target domain (*e.g.* due to are collection of footage in different locations or by different participants), leads to a model trained on source to have limited generalisation capability to the target domain.

- This domain shift can be mitigated with addition of unlabelled target data during training. Aligning the distributions of source and target videos can improve the performance of both action recognition and text-video retrieval tasks on target videos.

- Multiple modalities in video have differing degrees of robustness to domain shifts, and can be exploited for domain adaptation. The presence of a more robust modality can be used to create a more robust representation for the less robust modalities.

- Learning the temporal correspondence between multiple-modalities in video in both source and unlabelled target videos can improve action recognition performance on target data.

- Collection of videos 2 years apart, even by the same participants in the same location, results in a domain shift. This is due to different objects manipulated, wear and tear in the environment and different cameras used.
- When the source domain and the target domain have differing proportions of classes, marginal alignment of the distributions of source and target videos without considering the task is insufficient to overcome the domain gap. However, alignment of individual source and target videos depicting the same action can improve domain adaptation, using a pseudo-labelling approach to approximate target actions in the absence of target labels.
- Actions can be defined as pairs of verbs and noun. While many verbs and nouns can be shared across source and target domains, it is possible only few actions are shared. Therefore, aligning visual representations of verb and nouns in video separately from actions, can improve domain adaptation.

1.2 Contributions

The contributions of this thesis are as follows:

- The previously unexplored problem of domain adaptation for fine-grained action recognition is proposed, introducing two new benchmarks in Chapters 4 and 5.
- Multiple domain shifts are showcased within the recent dataset of fine-grained actions, EPIC-KITCHENS-100 dataset [10]. Chapter 4 showcases the domain shift between videos captured by different participants in different environments. In addition, there is a domain shift between videos collected by the same participants captured two-years apart, and is the focus of Chapter 5.

- A domain adaptation method for action recognition, utilising multiple modalities, is proposed in Chapter 4. This exploits the temporal correspondence between modalities present in video (appearance, motion and sound), and improves the ability of fine-grained action recognition models to adapt to new environments.

- The unsupervised domain adaptation challenge for action recognition, defined in Chapter 5, provided a public competition for the research community. A validation strategy was proposed such that participants could select hyper-parameters of their models, without accessing action labels from the training split of the target domain.

- The problem of domain adaptation for text-to-video retrieval is defined in Chapter 6. The aim is to retrieve semantically relevant videos to textual captions describing fine-grained actions. The difficulty is that there is a domain shift between the set of videos used for retrieval, and the captioned videos used during training.

- A domain adaptation method for text-to-video retrieval is proposed in Chapter 6, which utilises a pseudo-labelling approach to overcome the lack of text captions in the target domain. This allows source and target videos that depict the same action to be aligned. The usefulness of this method is showcased on the benchmark proposed in Chapter 5.

1.2.1 Contributions in Publications

Associated with this thesis is a first author publication [40], which is described in Chapter 4, and three co-authored publications outlining the collection of the EPIC-KITCHENS dataset [10, 30, 41]. This section outlines my contributions towards these co-authored papers.

My contribution towards the initial dataset collection, EPIC-KITCHENS-55 [30, 41], was transcribing audio narrations into textual descriptions in a scalable approach using Amazon Mechanical Turk. A more detailed explanation of how EPIC-KITCHENS was collected is available in

Sec 3.1.2.

My contributions towards the extended dataset, EPIC-KITCHENS-100 [10], was to organise the collection of footage from participants, and the design of the Domain Adaptation Challenge. The domain adaptation challenge is discussed in detail in Chapter 5.

1.3 Thesis Structure

This thesis starts with background information in Chapters 2 and 3. A detailed definition of unsupervised domain adaptation is provided in Chapter 2, and summarises key works in the domain adaptation literature. Chapter 3 provides an overview of the datasets and models used for video understanding, before reviewing domain adaptation works for action recognition and cross-modal retrieval.

Chapter 4 explores the domain shift between videos collected in different locations by different participants. A multi-modal domain adaptation method is proposed, which is showcased on a new domain adaptation benchmark, based on the EPIC-KITCHENS-100 dataset [10]. The benefits of utilising multiple modalities in video for domain adaptation is shown in this chapter.

A different domain shift is explored in Chapter 5, that occurs between the videos of the same participants captured two-years apart. This has the potential to negatively impact the performance of fine-grained action recognition models, deployed over long periods of time. An unsupervised domain adaptation challenge is proposed as a public competition, to encourage the development of solutions for this domain shift. The results of the challenge are presented at the end of the chapter.

While previous chapters focused on action recognition, Chapter 6 explores the task of text-to-video retrieval. The domain adaptation method proposed in this chapter shows the effectiveness of aligning videos of similar actions, rather than the global statistics of the domains.

Finally, Chapter 7 presents a conclusion of the work presented in this thesis.

Chapter 2

Unsupervised Domain Adaptation Literature

Classical machine learning assumes that the test set is independently and identically distributed to the training set. In practice, this is rarely the case. For fine-grained video understanding, models should be able to perform well with unseen participants, camera setups and recording environments. Unsupervised Domain Adaptation attempts to overcome this problem with access to unlabelled data from the test distribution (the target domain). Unlabelled data is used to overcome the distribution mismatch between training and test, which is often cheap to collect relative to annotation.

This section provides an introduction to domain adaptation, overviewing key works in the unsupervised domain adaptation literature. The majority of domain adaptation methods focus on the task of image classification tasks, and only recently has video seen some attention by researchers [1, 2, 37, 42, 43, 44]. Domain adaptation methods tailored for video understanding are discussed in detail in Chapter 3, once the various video architectures and datasets are defined. Although, many of the domain adaption solutions for video take inspiration from the methods outlined in this chapter.

Sec. 2.1 and Sec. 2.2 provide the reader with context of domain adaptation. First, domain adaptation is defined in Sec. 2.1, outlining the different types of domain adaptation tasks and domain shifts. Next, a summary of the mainly historical, shallow domain adaptation approaches is presented in Sec. 2.2, before providing an in-depth overview of the domain adaptation methods for deep learning models, in Sec. 2.3.

A discussion of key domain adaptation methods, and how they relate to this thesis, is provided in Sec. 2.3. The main methods for domain adaptation align the marginal distributions of the source and target distributions in feature space, which are described in Sec 2.3.2, 2.3.3 and 2.3.4. It is common to apply discrepancy metrics, such as Maximum Mean Discrepancy [45], to align domains (Sec 2.3.2). More recently, adversarial training has been used to align domains (Sec 2.3.3). Alternatively, batch normalisation layers in neural networks can be re-purposed for domain adaptation(Sec 2.3.4). In Chapters 4 and 5, the alignment of source and target distributions is used for domain adaptation. In particular, Chapter 4 uses adversarial training to align multi-modal architectures. Many methods in sections 2.3.2, 2.3.3 and 2.3.4 have been adapted for video used as baselines in the technical chapters.

Marginal alignment of distributions of source and target domains has a few disadvantages. The ideal model for the target domain may not always be one where the feature spaces are aligned. In addition, marginal alignment does not promote discriminative feature for the target task. As an alternative, Sec 2.3.5 explore works which use additional self-supervised tasks trained jointly with the task specific promote domain adaptation. In Chapter 4 we propose a self-supervised method tailored for video by exploiting the temporal correspondence of multiple modalities.

Alternative approaches to combat the limitations of marginal alignment assign class labels to unlabelled target instances (pseudo-labelling) and use these to train task specific losses on target data, or align source and target distributions of individual classes. In Chapter 6 we propose the use of pseudo-labelling to overcome the inability of marginal alignment to increase target domain performance.

Finally, we discuss literature for open-set domain adaptation, where some classes are present in the target domain that are not present in the source domain (Sec 2.3.7). Our benchmark proposed in Chapter 5 is also open-set due to the unscripted nature of the dataset collection.

2.1 Definition of Domain Adaptation

A domain is defined by the joint distribution $\mathcal{D}_{X,Y}$, over an input population X represented in a particular feature space $\mathcal{X}$, and the task labels Y in label space $\mathcal{Y}$. In classical supervised learning the aim is to find a mapping from the input space to the label space $h : \mathcal{X} \mapsto \mathcal{Y}$. However, this assumes that there is a single domain for training and testing. In practice there exists multiple domains: a source, $\mathcal{S} = \{\mathcal{D}^{\mathcal{S}}_{X,Y}, X^{\mathcal{S}}, Y^{\mathcal{S}}, \mathcal{X}^{\mathcal{S}}, \mathcal{Y}^{\mathcal{S}}\}$, where a large number of labelled instances are available during training, and a target $\mathcal{T} = \{\mathcal{D}^{\mathcal{T}}_{X,Y}, X^{\mathcal{T}}, Y^{\mathcal{T}}, \mathcal{X}^{\mathcal{T}}, \mathcal{Y}^{\mathcal{T}}\}$ used for evaluation. The difficulty stems from the fact that $\mathcal{T}$ might vary with respect to: the joint distribution of inputs and labels $\mathcal{D}^{\mathcal{S}}_{X,Y} \neq \mathcal{D}^{\mathcal{T}}_{X,Y}$, the input space $\mathcal{X}^{\mathcal{S}} \neq \mathcal{X}^{\mathcal{S}}$ or the label space $\mathcal{Y}^{\mathcal{T}} \neq \mathcal{Y}^{\mathcal{S}}$.

For some applications, the input and label space can differ between domains. However, the majority of domain adaptation works assume the same input and label spaces, $\mathcal{X}^{\mathcal{S}} = \mathcal{X}^{\mathcal{T}}$ and $\mathcal{Y}^{\mathcal{S}} = \mathcal{Y}^{\mathcal{T}}$.

$\mathcal{X}^{\mathcal{S}} \neq \mathcal{X}^{\mathcal{T}}$. Different input spaces arise from the use of different modalities, such as transferring information from: text→video, depth→images. Homogeneous domain adaptation assumes that the input features share the same space, whereas in heterogeneous domain adaptation, the input features are from different inputs spaces. In this work, the homogeneous case is assumed, where the source and target domains share the same feature spaces.

$\mathcal{Y}^{\mathcal{S}} \neq \mathcal{Y}^{\mathcal{T}}$ The label space differs between the domains. Examples of domain adaptation scenarios where the label spaces differ are: partial domain adaptation where the target domain is a subset of the source domain $\mathcal{Y}^{\mathcal{T}} \subset \mathcal{Y}^{\mathcal{S}}$ and open set adaptation where $\mathcal{Y}^{\mathcal{S}} \subset \mathcal{Y}^{\mathcal{T}}$. If $\mathcal{Y}^{\mathcal{S}} = \mathcal{Y}^{\mathcal{T}}$, it is known as closed set domain adaptation.

There is often a limited amount of labelled target data available during training. The amount of labelled target data available defines different domain adaptation tasks:

Supervised Domain Adaptation Labelled data is available in both the source and target domains.

Semi-supervised Domain Adaptation There is access to a limited amount of labelled target data during training, however the majority of target data should be unlabelled.

Domain Generalisation There is no access to any data in the target domain during training. Methods are trained on one or more source domains to generalise to an unseen domain.

Unsupervised Domain Adaptation The target domain is unlabelled so no labelled target data should be used for domain adaptation. This is the main focus of this work.

2.1.1 Defining the Domain Shift

Assuming the same feature spaces across domains, if the joint distribution differs across domains, $\mathcal{D}^{\mathcal{S}}_{X,Y} \neq \mathcal{D}^{\mathcal{T}}_{X,Y}$, a learned mapping from $h : \mathcal{X} : \mathcal{Y}$ that performs well on the source data may exhibit poor performance on target data. The reason behind the joint distribution shift provides different challenges for domain adaptation.

$\mathcal{D}^{\mathcal{S}}_{X} \neq \mathcal{D}^{\mathcal{T}}_{X}$ The **co-variate shift**, or **marginal distribution shift**, is where the input features from different domains are distributed differently. This is the focus for the majority of domain adaptation problems. The domains are said to have different marginal distributions as the task labels are not considered and have been marginalised out of the distributions. There are often multiple factors behind this shift for fine-grained video understanding, including different camera setup, participants and environments across domains.

How the input distribution conditioned on a single class changes, $\mathcal{D}^{\mathcal{S}}_{X|Y} \neq \mathcal{D}^{\mathcal{T}}_{X|Y}$, is referred to as a **conditional shift**. Some methods exploit this to improve alignment.

$\mathcal{D}^{\mathcal{S}}_{Y} \neq \mathcal{D}^{\mathcal{T}}_{Y}$ The **label shift** refers to the different proportions of labels in both source and target domains. This is also linked to the class imbalance problem where at training there is a long-tailed distribution of instances per class, but at test time a uniform distribution is expected.

$\mathcal{D}^{\mathcal{S}}_{Y|X} \neq \mathcal{D}^{\mathcal{T}}_{Y|X}$ The **semantic shift** changes the meaning of labels between domains, such that the distribution over the labels conditioned on the input changes. This is difficult to overcome but can be an issue for cross-dataset action recognition due the ambiguous nature of verbs.

2.1.2 Domain Adaptation Theory

A theoretical work by Ben-David *et al.* [46] provides an upper bound of the labelling error on the target domain, $e^{\mathcal{T}}(h)$, given a hypothesis function, h. This inspired the majority of the recent unsupervised domain adaptation methods. They showed the target labelling error, $e^{\mathcal{T}}(h)$, is bounded by the labelling error on the source domain, $e^{\mathcal{S}}(h)$, and the marginal distribution discrepancy between the domains, $d_{\mathcal{H}\nabla\mathcal{H}}$.

The $\mathcal{H}$-divergence discrepancy measure, $d_{\mathcal{H}\nabla\mathcal{H}}$, is used to prove the upper bound on the target labelling error, and is defined as the supremum over a class of hypothesis functions, $\mathcal{H}$, which discriminate between source and target distributions, $\mathcal{D}^{\mathcal{S}}$ and $\mathcal{D}^{\mathcal{T}}$. $Pr_{x\sim\mathcal{D}}[h(x) \neq h'(x)]$ is the probability that two hypothesis functions, h and h', disagree on input samples from distribution, D.

$$d_{\mathcal{H}\nabla\mathcal{H}}(\mathcal{D}^{\mathcal{S}}, \mathcal{D}^{\mathcal{T}}) = 2 \sup_{h,h' \in \mathcal{H}} |Pr_{x\sim\mathcal{D}^{\mathcal{S}}}[h(x) \neq h'(x)] - Pr_{x\sim\mathcal{D}^{\mathcal{T}}}[h(x) \neq h'(x)]| \tag{2.1}$$

Theorem 1. *Given the distribution of input samples in the source and target domains, $\mathcal{D}^{\mathcal{S}}$ and $\mathcal{D}^{\mathcal{T}}$,* *the upper bound for the target error for every hypothesis $h \in \mathcal{H}$ is given by:*

$$e^{\mathcal{T}}(h) \leq e^{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\nabla\mathcal{H}}(\mathcal{D}^{\mathcal{S}}, \mathcal{D}^{\mathcal{T}}) + \lambda \tag{2.2}$$

where $\lambda = e^{\mathcal{S}}(h^*) + e^{\mathcal{T}}(h^*)$ is the combined error from the source and target domains, given the ideal joint hypothesis:

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} e^{\mathcal{S}}(h) + e^{\mathcal{T}}(h) \tag{2.3}$$

Many works assume that λ is small and focus on minimising the distribution discrepancy, $d_{\mathcal{H}\nabla\mathcal{H}}$. Ben-David *et al.* [47] prove that if there is no hypothesis function, $h^* \in \mathcal{H}$, that achieves low error on both domains, labelled data in the target domain is required to guarantee successful domain adaptation. Therefore to overcome a semantic shift between the domains, labelled target data would be needed.

2.1.3 Measuring the Domain Shift

To measure the co-variate shift, a variety of discrepancy measures such as the KL-divergence, Wasserstein metric and the Maximum Mean Discrepancy (MMD) [45] have been proposed. Theorem 1 (Sec.2.1.2) showed that minimizing the discrepancy between source and target distributions will improve the ability of a classifier trained on the source domain to generalise to the target domain. MMD is popular in the literature due to the fact that it is a non-parametric measure that does not require density estimates. MMD is referred to frequently throughout this thesis, therefore this section provides the definition of the measure.

MMD measures the distance between the mean embedding of two distributions in Reproducing Kernel Hilbert Space (RKHS). This captures the discrepancy between domains as the mean in RKHS can fully represent a single distribution. Given n inputs from the source domain, $S_n =$

$(x_1^s \dots x_n^s)$, m inputs from the target domain, $T_m = (x_1^t \dots x_m^t)$, and a mapping to a Hilbert space, $\phi : X \mapsto \mathcal{H}$, MMD is defined as:

$$MMD(S_n, T_m) = \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(x_i^s) - \frac{1}{m} \sum_{i=1}^{m} \phi(x_i^t) \right\|_{\mathcal{H}}^2 \tag{2.4}$$

Note that ϕ is not used explicity, rather a kernel trick is used. The kernel, $k(\cdot, \cdot)$, must be positive semi-definite to ensure it maps to RKHS, which is most often the radial basis function (Gaussian kernel).

$$MMD(S_n, T_m) = \left\| \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i^s, x_j^s) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(x_i^s, x_j^t) + \frac{1}{m^2} \sum_{i=1} \frac{1}{m^2} \sum_{j=1}^{m} k(x_i^t, x_j^t) \right\|^{0.5} \tag{2.5}$$

This section introduced the problem of domain adaptation, what a domain shift is and how to measure it. The next sections outline works that tackle the problem of unsupervised domain adaptation.

2.2 Shallow Domain Adaptation

These domain adaptation methods work directly in a feature space engineered for a particular task, therefore they do not combine domain adaptation with representation learning. Later sections will focus on more recent techniques, which jointly optimise domain alignment with learning a feature representation with neural networks. Therefore, this section provides a historical background for domain adaptation. A more in-depth comparison between methods in literature and the work in this thesis is provided in Sec 2.3.

Early domain adaptation approaches re-weighted source instances to match the target distribution (Sec. 2.2.1), augmenting the feature representation (Sec 2.2.2) or transforming the domains into a space where the source and the target distributions are aligned (Sec 2.2.3). Creating a representation which aligns source and target distributions inspired the majority of recent methods, described in Sec 2.3. In a similar spirit, methods used in Chapters 4,5 and 6 also learn a representation where the source and target distributions are aligned.

2.2.1 Instance-based Adaptation

To improve the performance of classifiers on target data, early works re-weighted source instances to approximate the target distribution [48, 49, 50, 51]. These assume there is no semantic shift between the source and the target distributions, $\mathcal{D}^{\mathcal{S}}_{Y|X} = \mathcal{D}^{\mathcal{T}}_{Y|X}$, in order to match the marginal distributions, $\mathcal{D}^{\mathcal{S}}_{X}$ and $\mathcal{D}^{\mathcal{T}}_{X}$. This is also referred to as importance weighting as source instances that are more representative of the target distribution are given larger weights than those far from target instances in feature space.

The motivation behind these works is shown in Eq. 2.6. The risk on the target domain is approximated by n source instances and labels, $S_n = \{x^{\mathcal{S}}, y^{\mathcal{S}}\} \sim \mathcal{D}^{\mathcal{S}}_{X,Y}$. $\mathcal{L}$ is a loss function which determines the fitness of the mapping from input space to label space, $h : \mathcal{X} : \mathcal{Y}$, with parameters θ.

$$\begin{aligned}
\theta &= \underset{\theta}{\operatorname{argmin}} \int\int \mathcal{D}^{\mathcal{T}}_{X,Y}(x,y)\mathcal{L}(y,x;\theta)dxdy \\
&= \underset{\theta}{\operatorname{argmin}} \int\int \frac{\mathcal{D}^{\mathcal{T}}_{X,Y}(x,y)}{\mathcal{D}^{\mathcal{S}}_{X,Y}(x,y)}\mathcal{D}^{\mathcal{S}}_{X,Y}(x,y)\mathcal{L}(y,x;\theta)dxdy \\
&\approx \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{x^s,y^s \in S_n} \frac{\mathcal{D}^{\mathcal{T}}_{X,Y}(x^{\mathcal{S}}_i, y^{\mathcal{S}}_i)}{\mathcal{D}^{\mathcal{S}}_{X,Y}(x^{\mathcal{S}}_i, y^{\mathcal{S}}_i)}\mathcal{L}(y^{\mathcal{S}}_i, x^{\mathcal{S}}_i;\theta) \\
&= \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{x^s,y^s \in S_n} \frac{\mathcal{D}^{\mathcal{T}}_{Y|X}(y^{\mathcal{S}}_i|x^{\mathcal{S}}_i)}{\mathcal{D}^{\mathcal{S}}_{Y|X}(y^{\mathcal{S}}_i|x^{\mathcal{S}}_i)}\frac{\mathcal{D}^{\mathcal{T}}_{X}(x^{\mathcal{S}}_i)}{\mathcal{D}^{\mathcal{S}}_{X}(x^{\mathcal{S}}_i)}\mathcal{L}(y^{\mathcal{S}}_i, x^{\mathcal{S}}_i;\theta)
\end{aligned} \tag{2.6}$$

If there is no semantic shift in the meaning of the labels, $\mathcal{D}^{\mathcal{S}}_{Y|X} = \mathcal{D}^{\mathcal{T}}_{Y|X}$, the target domain performance can be found by weighting source examples by $\beta = \frac{\mathcal{D}_X^{\mathcal{T}}(x^s)}{\mathcal{D}_X^{\mathcal{S}}(x^s)}$. A common approach to approximate β is to minimise a distribution discrepancy measure between the target and re-weighted source distributions with respect to β [48, 50]. Key works have minimised the Kuback-Libler divergence [50] or Maximum Mean Discrepancy [48, 49, 51].

A major disadvantage of these methods is that the target domain performance suffers if there is a large domain shift, where target instances are far from the source instances in input space. Therefore, more recent shallow based approaches focused on augmenting or transforming the feature representation such that the domains are close.

2.2.2 Feature Augmentation

A simple method for supervised domain adaptation augments the input features with copies of itself [52]. Eq 2.7 shows that the augmented features, $\Phi(x^{\mathcal{S}})$ and $\Phi(x^{\mathcal{T}})$, are produced from the source and target inputs, $x^{\mathcal{S}}$ and $x^{\mathcal{T}}$, and a vector of zeros of the same shape, **0**. The first set of indices of the augmented features is used to learn shared characteristics between source and target, whereas the second and third sets of indices represent domain specific features.

$$\Phi(x^{\mathcal{S}}) =< x^{\mathcal{S}}, x^{\mathcal{S}}, \mathbf{0} > \quad \Phi(x^{\mathcal{T}}) =< x^{\mathcal{T}}, \mathbf{0}, x^{\mathcal{T}} > \tag{2.7}$$

This motivated unsupervised domain adaptation methods to augment low-dimensional representations of the input space with sub-spaces sharing similar characteristics to both source and target domains. Geodesic Flow Sampling (GFS) [53] creates a low-dimensional representation of the source and target domains, and embeds them onto the Grassman manifold. The Grassman manifold provides a topological structure over all possible sub-spaces with the same dimensions. Sub-spaces are selected along the geodesic flow, the shortest path between the source and target, that provides incremental changes from source to target. As it is difficult to choose the best sub-

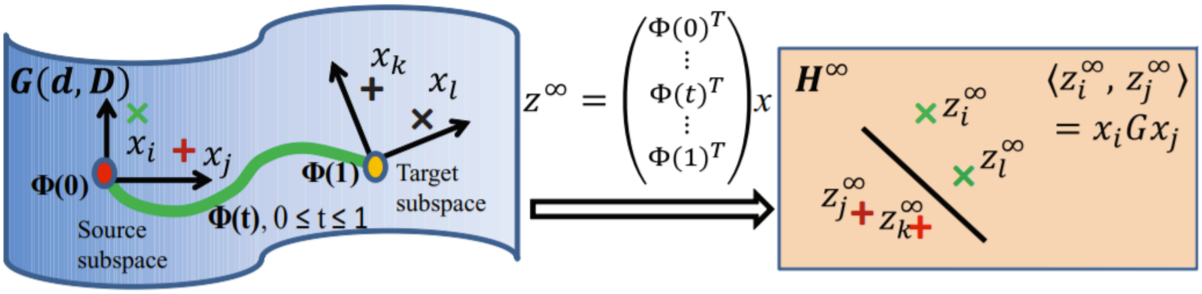

$G(d,D)$
x_k
x_l
x_i
x_j
$\Phi(0)$
$\Phi(1)$
Target subspace
Source subspace
$\Phi(t), 0 \leq t \leq 1$
$z^{\infty} = \begin{pmatrix} \Phi(0)^T \\ \vdots \\ \Phi(t)^T \\ \vdots \\ \Phi(1)^T \end{pmatrix} x$
H^{∞}
z_i^{∞}
$\langle z_i^{\infty}, z_j^{\infty} \rangle = x_i G x_j$
z_l^{∞}
z_j^{∞}
z_k^{∞}

Figure 2.1: Sampling sub-spaces along the Geodesic Flow to augment the input feature space. Figure taken from [3]

space for domain adaptation, the domains are projected into multiple spaces along the geodesic flow. The representations from each sub-space are concatenated and used to train a classifier on the source. Alternatively, Geodesic Flow Kernel (GFK) [3] uses all sub-spaces along the geodesic flow, which exploits the kernel trick to train a classifier in an inifite number of subspaces without projecting into each of the spaces explicitly.

Although feature augmentation can aid domain adaptation, the augmented source and target features do not necessarily have the same distribution. As the target error is bounded by the discrepancy between source and target input distributions (Theorem 1, Sec.2.1.2), the next section outlines works that transform the input space such that the domains are aligned.

2.2.3 Feature Transformations

These works transform the source and target features such that the marginal distributions between the transformed features are close. This transformation can be jointly learned as a projection into a low-dimensional space [54, 55], or into a space of the same cardinality [56, 57]. Methods find a common projection to map source and target domains into an aligned space [54, 55], or learn a separate projection to map each domain into a common sub-space [56]. Alternatively, the source domain can be directly mapped onto the target domain [57, 58].

Domain Alignment can be achieved jointly while learning a projection into a low-dimensional space [54, 55]. Transfer Component Analysis (TCA) [54] is inspired by MMD and learns a projection that minimizes the distance between the means of the the source and the target distributions in the transformed feature space. The alignment is optimised with an orthogonality constraint on the projection matrix in order to preserve task information. Domain Invariant Projection (DIP) [55] learns a projection that minimises the means of the distribution in RKHS with a Gaussian kernel. This better resembles MMD as domains are compared in RKHS rather than the projected low-dimensional space. However DIP has no closed form solution and uses a conjugate gradient method to find the projection. DIP improves target object detection performance upon GFK, TCA and instance-sampling methods [51, 59] on the Office-dataset [60].

Other approaches conduct alignment separately from dimensional reduction [56, 57]. Sub-space Alignment (SA) [56] first projects the domains into a low-dimentional space before alignment. The projected space is transformed into a sub-space where the Frobenius norm between the source and the target distributions is minimised. One advantage of this method is that there are no hyper-parameters that require tuning for domain alignment. Another simple but effective method which has no hyper-parameters is Correlation Alignment [57]. This transforms the source features to match the target co-variance. Co-variance matching is approximated by whitening the source data and *'re-colouring'* it with the target co-variance. The target features are not transformed. They showed improved performance over GFK [3], SA [56] and TCA [54] on the Office and extended Office-Caltech100 datasets [60] arguing that the improved performance is due to their alignment method considering both eigenvalues as well as eigenvectors.

Feature transformation can be combined with instance re-weighting [61]. Transfer Joint Matching [61] builds upon the feature transformation approach, TCA, but imposes L_1 and L_2 sparsity regularisation on the rows of the PCA projection matrix corresponding to source instances. This regularisation promotes source examples irrelevant to the target to be down-weighted in the projection which matches the means of the distributions. While maximising variance of the projected features and minimising row-sparsity of the source features, their strategy down-weights the contribution from irrelevant source examples in the low-dimensional space.

An approach quite different from the others solves an optimal transport problem to minimize the discrepancy between source and target [58, 62, 63]. The aim is to find a transport plan, γ, which is a probabilistic coupling between the source and the target distributions. The set of possible transport plans is given by : $\mathcal{B} = \{\gamma \in (\mathbb{R}^+)^{n \times m} | \gamma \mathbf{1}_m = \mu_s, \gamma^T \mathbf{1}_n = \mu_t\}$, where $\mathbf{1}$ is a column of n or m ones, and μ_s and μ_t are the discrete distributions of the source and target examples. Another interpretation is that γ states how much *mass* of each source example should be transported to each target example, ensuring that the distribution of the mapped source example matches the target distribution and visa versa. The optimal plan, γ_o, is the plan that minimises a cost matrix C between source and target examples, x^s and x^t, which is often the Fronebious norm.

$$\gamma_o = \operatorname*{argmin}_{\gamma \in \mathcal{B}} \sum_i \sum_j \gamma_{i,j} C_{i,j} \qquad where \quad C_{i,j} = \left\| x_j^s - x_j^t \right\| . \tag{2.8}$$

The transformed source examples, $\gamma_o(x^s)$, can be used to train a classifier [58], or a separate transformation can be learned to approximate γ, improving the performance on target examples not seen during training [62]. Class information can be used to ensure that source examples from the same class are not mapped to the same target examples [64], or the joint distribution of inputs and label can be aligned [63].

2.2.4 Conditional Alignment

The works in the previous section focused on matching the marginal distributions without utilising class information. However, this doesn't ensure that samples of the same class share the same distribution in both the source and the target domains. These works aim to align the conditional distributions between the source and the target, $\mathcal{D}_{\mathcal{X}|\mathcal{Y}}$. Discrepancy measures are applied per-class to align domains, using the class labels of the source domain. However, for unsupervised domain adaptation, no labels from the target domain are available during training. Instead, these methods utilise *pseudo-labels*, which are the class label prediction on target data. These method utilise an iterative strategy between training the model for domain alignment and computing the pseudo-labels.

Some methods adapt existing approaches for marginal distribution matching to consider class information. Transfer Joint Analysis [65] extended TCA to match the conditional distributions. TCA is applied individually for each class in the source and target domain. JDOT [63] incorporates class information into the optimal transport problem [58].

Metric based approaches explored using pseudo-labels to adapt a source representation to the target domain [66]. A projection is used to learn a metric space, such that source inputs from the same class are closer than instances from inputs of differing classes. Inputs are labelled by the class associated with the nearest centroid in the projection. Domain Specific Class Means (DSCM) [66] explores the multi-source setting where multiple source domains are present during training, therefore, multiple centroids are defined for each class, one for each source. They propose to adapt the metric space to better fit the target data in an interactive manner. Each iteration target inputs are pseudo-labelled, and the target instance closest to the nearest centroid of the same label are added to the source set, and the source instance furthest from the nearest centroid are removed. The metric space is then updated with the set containing both source and target data until no more target data can be added. This work showed that learning a metric space can improve domain adaptation over learning a classifier in the original space, outperforming SA [56] when the metric space is adapted with target data.

2.3 Deep Domain Adaptation

This section summarises the approaches which combine unsupervised domain adaptation with the learning of representations with deep learning architectures. Initial work simply extracted features from deep learning architectures, which shallow domain methods approaches can be applied to (Sec. 2.3.1). Deep learning representations greatly improved domain adaptation, therefore the work in this thesis focuses on deep learning.

Improved target domain performance is achieved by adapting deep learning models for domain adaptation. Works have aligned feature activation from source and target domains by minimising discrepancy metrics (Sec. 2.3.2), adversarial training (Sec. 2.3.3) and re-purposed batch normalisation layers (Sec. 2.3.4). Throughout this thesis, our methods utilise ideas in these sections to align source and target domain distribution. In addition, we directly apply many of these methods directly to video architectures as baselines.

Other approaches used additional self-supervised objectives on target data (Sec. 2.3.5), or pseudo-labelling approaches to obtain labels for target task (Sec. 2.3.6). A self-supervised method tailored to video is proposed for domain adaptation in Chapter 4, inspired by works in Sec. 2.3.5. In addition, a pseudo-labelling is proposed in Chapter 6 with many works in Sec. 2.3.6 used as baselines.

Most approaches assumed closed-set domain adaptation where the label space is the same between domains, although some approaches relax this assumption (Sec. 2.3.7). The benchmark described in Chapter 5 provides an open-set domain adaptation benchmark for video. Not all actions classes that occur in the source domain, occur in the target domain, and vice-versa. Chapter 6 proposes a domain adaptation method on this benchmark, using a sampling strategy align actions of common classes across domains.

2.3.1 Deep Feature Extraction

Features extracted from deep learning methods have been shown to create more transferable representations for the target domain [67, 68], greatly improving domain adaptation performance on standard domain adaptation benchmarks [60]. Shallow domain adaptation methods can be applied directly to these deep features. Therefore, Chapters 4, 5 and 6 focus on deep learning approaches for action understanding.

Initial work using deep features for domain adaptation was for sentiment classification of amazon reviews [67]. Stacked Denoising Autoencoders (SDA) were trained in a greedy layer wise fashion, extracting the intermediate layers as inputs feature. They showed that training an SVM on the extracted features from the SDA could reduce the transfer error (the difference in accuracy between the domain adaptation method and an SVM trained on raw pixel values from target) compared to feature alignment of raw inputs [59].

Donahue *et al.* showed that features extracted from Convolution Neural Networks (CNN) can generalise to unseen datasets and greatly improve domain adaptation performance [68]. They proposed DeCaf which extracts the feature activations from AlexNet trained on the ImageNet dataset [69]. These features showed large improvements on the Office dataset [60] over using SURF [70] which was previously used by most domain adaptation methods to provide a robust feature representation.

Despite the improved generalisation capability of features extracted from CNNs [68], the performance of models trained on these features drops when evaluating across datasets [33]. Similarly, our work shows that fine-grained action understanding models show a drop in performance when evaluated on a different domain, where the domain shift is due to videos filmed in different locations (Chapter 4), or videos collected two years later apart (Chapters 5 and 6). Instead, more recent approaches jointly optimise domain adaptation with the learning of features using neural networks. Next we discuss key works in the literature that align source and target domains of deep learning models to improve domain adaptation.

2.3.2 Deep Discrepancy Minimisation

A common approach to adapt neural networks for domain adaptation is to minimize some discrepancy, such as the Maximum Mean Discrepancy metric (MMD) [45], jointly with the task specific losses, such that the source and target distributions remain similar. Unlike shallow domain adaptation methods, these losses are back-propagated through the Neural Network to learn the feature representation.

Domain Adaptation Neural Networks (DANN) [71] was one of the first methods to apply domain adaptation to deep learning architectures. MMD was applied to the hidden layers of a multilayer perception or a de-noising autoencoder to reduce the domain gap. Training was conducted in two steps, first the network was pretrained using labelled source examples to achieve good classification performance, followed by fine-tuning with the MMD objective to align the distribution of source and target. The addition of MMD provided improved target domain accuracy on the office dataset [60], outperforming older shallow transfer-learning methods [3, 56] . However, using SURF features [70] as input to their method still outperformed raw images as input as both networks were shallow and not pre-trained on large datasets.

Adapting Convolution Neural Network (CNN) architectures, such as AlexNet [16], for domain adaptation greatly improved domain adapation performance. Deep Domain Confusion (DDC) [72] applies MMD to a bottleneck layer, which was placed directly before the final fully connected layer. The low-dimensional bottleneck layer limited the ability of the network to over-fit to the source, and MMD aligned the source and the target distributions. The placement and size of the bottleneck layer was also chosen such that MMD between source and target was minimised, from the network pre-trained on source data. MMD and the classification losses were jointly optimised and back-propagated through the network. This work showed substantial improvements on DANN, outperforming handcrafted [70] and DeCAF [68] features.

Discrepancy measures can be used to align multiple layers of a neural network [73, 74]. Domain Adaptation Networks (DAN) [73] applies MMD to multiple, feed-forward, task specific layers of AlexNet. Although the lower-layers of a CNN are often general and less dataset specific, the feed-forward layers higher in the network suffer greatly from a domain shift. Note that separate MMD losses were trained jointly to align different layers of the network. Joint Adaptation Network (JAN) [74] uses a single MMD measure to align the joint distribution over multiple layers in RKHS. The authors argued that capturing the joint distributions over multiple layers, MMD can better reason about the conditional shift between domains.

One difficulty of the MMD metric is choosing the correct kernel. Most often a Gaussian kernel is used, however changing the variance parameter will result in a different weighted combination of moments used by the discrepancy measure. Long *et al.* [73] proposed a multiple-kernel variant of MMD, which used a variety of Gaussian kernels with different variance to avoid the issue of choosing an appropriate kernel for the test statistic. Zellinger *et al.* [75] argued that it is most important to align the central moments of the first k orders, and defines Central Moment Discrepancy (CMD). This avoids the kernel calculations in MMD and the authors argued that classification accuracy is less sensitive to changes in k than the choice of kernel in MMD.

Many discrepancy measures for deep domain adaptation, including those using MMD, were inspired by shallow domain adaptation approaches [57, 63]. Deep CORAL [76] extends CORAL [57] to align the activations of the last fully connected layer of AlexNet. The Euclidean distance between source and target covarience matrices is minimised with respect to the weights of the neural network. The covariance matricies are computed using the current mini-batch statistics of each domain. This showed comparable target classification accuracy on the Office dataset [60] compared to DAN [73]. DeepJDOT [77] applies the shallow optimal transport alignment approach, JDOT [63], to hidden layers of a neural network. A iterative approach is taken for optimisation. First JDOT is optimised to find the optimal mapping from source to target, before minimising the cost of the optimal mapping with respect to the weights of the nerual network. This provided large improvements on target classification accuracy compared to DAN [73] on the Office [60] and Office-Home datasets [78]. Note that DeepJDOT uses label information during alignment, where pseudo-labels, or class predictions, are obtained for the unlabelled target data. Pseudo-labelling approaches are discussed in more detail in Sec. 2.3.6.

This section discussed works that align source and target distributions for domain adaptation. Chapters 4 and 6 propose domain adaptation methods which align source and target distributions. The key work in the literature DAN [73], which uses the multiple kernel variant of MMD to align domains, is applied to video architectures and used as a baseline in both chapters.

2.3.3 Adversarial Alignment

An alternative to discrepancy minimization is to use adversarial training [4]. This uses the minimax game of Generative Adversarial Networks [79] to align the source and target domains. We propose adversarial training to align multi-modal video architectures in Chapter 4, and apply discriminate adversarial domain alignment as baselines in Chapter 5 and Chapter 6.

Adversarial methods can be generative, which translate source images into the style of the target [80, 81, 82], or discriminative, which align the domains in feature space [4, 83, 84, 85, 86]. Due to the success of adversarial alignment, recent domain adaptation approaches for action recognition incorporate discriminative adversarial training [1, 2, 37, 43]. In Chapter 4, we show that adversarial training is an effective method for domain adaptation for fine-grained action recognition, and how to employ adversarial training for multi-modal architectures.

Discriminative approaches consist of a feature extractor, G, used to produce the representation of source and target inputs, $x^{\mathcal{S}}$ and $x^{\mathcal{T}}$, and a discriminator, D. The aim of D is to predict if a given feature representation is sampled from the source, $\mathcal{D}^{\mathcal{S}}$, or the target domain, $\mathcal{D}^{\mathcal{T}}$. By learning to distinguish between examples in each domain, D can be used as a learning signal for G to align

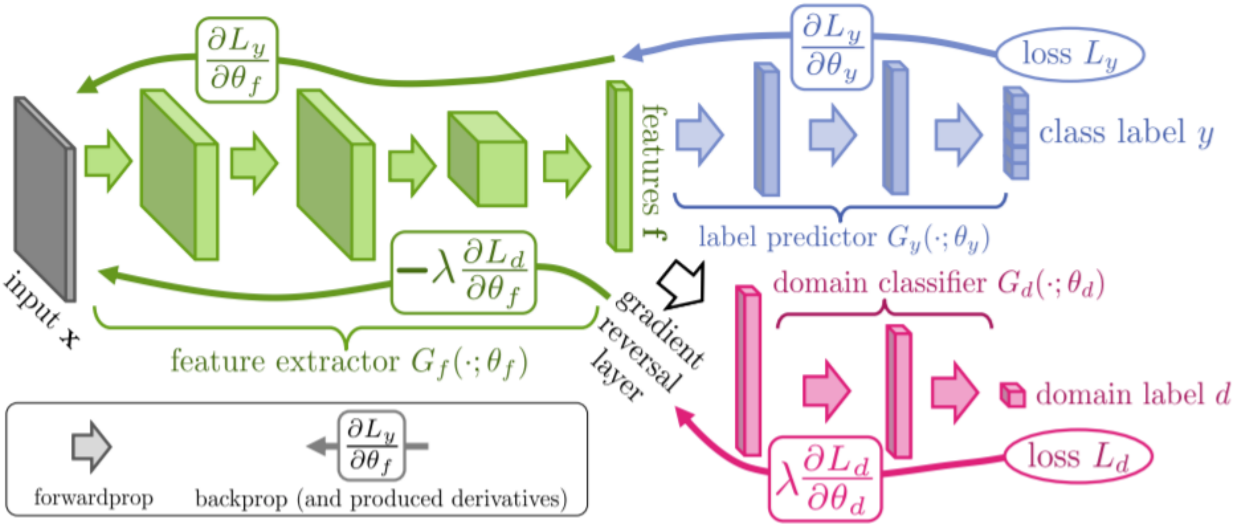

$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
features f
class label y
label predictor $G_y(\cdot;\theta_y)$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
input x
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
$\lambda\frac{\partial L_d}{\partial \theta_d}$
loss L_d
forwardprop
$\frac{\partial L_y}{\partial \theta_f}$
backprop (and produced derivatives)

Figure 2.2: Domain Adversarial training with a gradient reversal layer (GRL). Figure taken from ReverseGrad [4]

.

the domains. D is optimised to minimise the following loss function:

$$\mathcal{L}_d(\theta_D,\theta_G) = \mathbb{E}_{x^{\mathcal{T}}\sim\mathcal{D}^{\mathcal{T}}}[\log D(G(x^{\mathcal{T}};\theta_G);\theta_D)] + \mathbb{E}_{x^{\mathcal{S}}\sim\mathcal{D}^{\mathcal{S}}}[\log(1-D(G(x^{\mathcal{S}};\theta_G);\theta_D)] \tag{2.9}$$

A minimax optimisation is used to align the domains, $\min_{\theta_D}\max_{\theta_G}\mathcal{L}_d(\theta_D,\theta_G)$, where θ_D and θ_G are the weights of D and G, respectively. Therefore, the feature extractor will bring the domains closer in order to fool the domain discriminator, while the domain discriminator becomes better at distinguishing between the domains. Ganin *et al.* [4] showed that D can be interpreted as learning the hypothesis function that distinguishes between source and target distributions in Theorem 1 (Sec. 2.1.2). Therefore, similar to the previous section, adversarial optimisation will align the marginal distributions between domains.

The seminal work, ReverseGrad [4] (Fig. 2.2), uses a Gradient Reversal Layer (GRL) to solve the minimax objective in a single optimisation step. During the forward pass, GRL acts as the identity function, whereas in the backward pass the gradient is negated before back-propgating into the feature extractor. Adversarial alignment is jointly optimised with the source domain classification task. The task classifier, C, acts on the features generated from G with a classification loss $\mathcal{L}_y$. A hyper-parameter λ is used to trade-off the alignment and classification losses. The gradients of the domain discriminator will not be useful at the start of training, therefore a warmup strategy is used to initialise the domain discriminator by increasing λ from zero to one as a function of the current training step.

Alternatively GAN based losses [83] can be used to adversarially align domains, this is advantageous if separate weights are used to map the source and the target domains into feature space. The target specific weights can learn domain specific low-level features, such as textures and basic shapes, to align the more class specific layers with the source domain. The initialisation of the target weights is important to avoid degenerate solutions. The authors proposed to first initialise the target weights with source weights pretrained on the classification task. For alignment, a domain discriminator is used to map source and target features to a domain label, while the target weights are optimised to minimise the domain loss with inverted domain labels.

Improving Discriminative Target Representations

Chen *et al*. [87] showed that adversarial alignment approaches are less discriminative than models trained only on source. They explain this through singular value decomposition, where ReverseGrad [4] produces one singular value much larger than the others. They proposed Batch Spectral Penalization (BSP) to penalise the k largest singular values, which improves upon ReverseGrad on the Office-31 [60], Office-Home [78] and VisDA-2017 [88] datasets. Other works improve discriminability by considering the conditional distributions [84, 86, 89], using multiple domain discriminators[85, 86], or an adversarial minimax game between multiple classifiers [5, 90]

To align the conditional distributions, class label information can be included when training the domain discriminator [84, 89, 91]. Conditional Domain Adaptation Network (CDAN) [84] conditions the domain discriminator on both the activations of the feature extractor and the logits of the task classifier. The logits enable the domain discriminator to utilise class information during alignment. In CDAN the outer-product between the logits and the features are used to explicitly capture the interaction between the two spaces. They showed that the concatenation of the spaces leads to worse performance, as it doesn't explicitly model the interaction between features and class probabilities. The downside of the outer-product operation is that the input to the domain discriminator can be very large. Therefore, an approximation is provided which multiplies the individual spaces by separate random vectors, that are fixed during training, followed by element-wise multiplication. This allows a fixed sized vector as input to the domain discriminator. Their method outperforms MMD based alignment [73, 74] and adversarial approaches [4, 83] on the Office-31 [60], Office-Home [78] and ImageCLEF-DA [92] datasets, and generative approaches [81, 93] on the VisDA2017 challenge [88]. Generative approaches are discussed later in this section.

Multiple domain discriminators have been used to align the conditional distributions [86]. Each domain discriminator takes in source features from a single class. However, as the target domain in unlabelled, the classifier probability for the kth class is used to weight the target features used as input for the kth domain discriminator. This approach provides small improvements over ReverseGrad on the Office-31 [60] and ImageCLEF-DA [92] datasets, but larger gains were achieved in the partial domain adaptation setting, where only 25 out of the 31 classes were used in the target domain. Partial domain adapation will be revisited in Sec. 2.3.7.

Some adversarial approaches aim to incorporate class knowledge into a single domain discriminator [89, 91]. The domain discriminator in Discriminative Adversarial Domain Adaptation (DADA) [91], predicts $K + 1$ labels. $(1 \cdots K)$ represent the classes for the source classification task, and $K + 1$ is the domain label. The domain discriminator is optimised to predict ground-truth class labels for source examples, and the domain label for target examples. The addition of the class information into the domain discriminator encourages the feature extractor to align

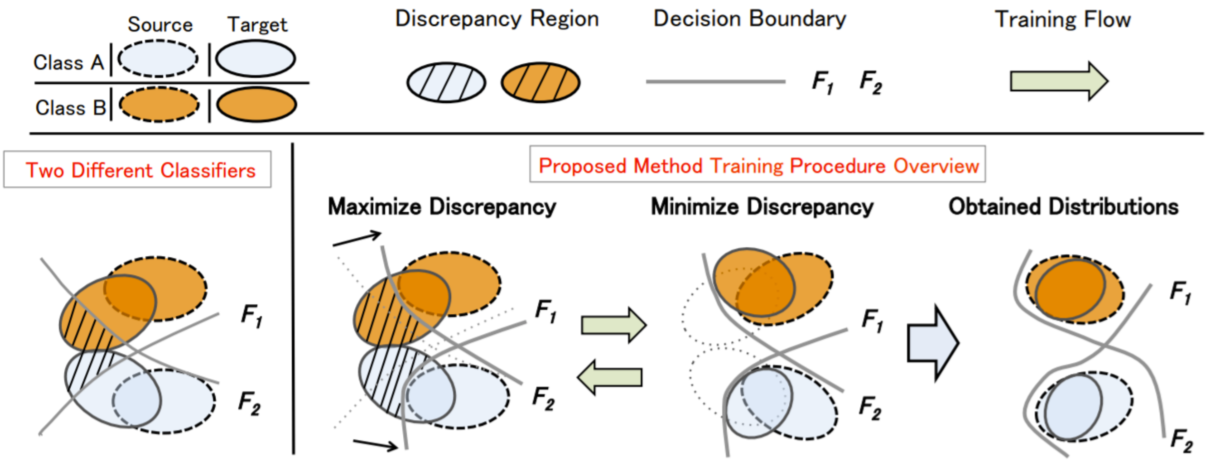


Figure 2.3: Adversarial training with Maximum Classifier Discrepancy (MCD). Figure taken from [5]

target instances to one of the K source classes, avoiding target examples lying near of decision boundaries.

Zhang *et al*. [85] applied multiple domain discriminators to encourage domain specific features in the lower layers of a CNN and domain invariant features in the higher layers. While the domain invariant layers are learnt using a GRL, the domain specific features are obtained by back-propagating the discriminator gradient without negation. Therefore the lower layers are encouraged to learn domain specific low-level attributes, such as corners and edges, that may aid class discrimination in the target domain.

Adversarial Training using Task-Classifiers

Adversarial training strategies have been proposed that avoid the use of a domain discriminator and use a mini-max optimisation directly on the task classification scores [5, 90, 94]. Maximum Classifier Discrepancy (MCD) (Fig. 2.3) [5] exploits the disagreement of classification scores from two classifiers sharing the same feature extractor. The L1 distance between the class probabilities is referred to as the classifier discrepancy, which measures the disagreement between classifiers. This discrepancy is maximised on target data, whilst each classifiers is optimised to correctly predict the class label on source data. Therefore, the classifiers will disagree on the class label predictions for target examples that are far from source examples. The domains are aligned by minimizing the classifier discrepancy with respect to the feature extractor weights, such that classifiers have similar predictions. This minimax game can be optimised iteratively, or in a single step with a gradient reversal layer placed between the features and the two classifiers, applied only to target data. Adversarial Dropout Regularization [90] uses the same approach as MCD but uses dropout to obtain the two separate classifiers. Dropout randomly sets a percentage of feature activations used as input to the classifier to zero. Applying dropout twice obtains two classifiers that can be optimised using MCD.

Generative solutions

Generative solutions for domain adaptation translate source images into the style of the target [80, 81, 82, 93, 95]. Initial work [80, 95] used a GAN architecture, where a generator is trained to reconstruct the original source image while the discriminator is trained to distinguish the reconstructed source image from target images. By maximising the discriminators loss with respect to the generator, the GAN produces images with a similar style to the target with task information from the source preserved. Gen-to-adapt [93] uses the same feature representation for the generator and the task classifier. The feature extractor is jointly trained to reconstruct source images in the style of the target and to predict the task labels. Cycle consistency constraints were added to avoid features collapsing into uni-modal solutions [81, 82]. This translates the generated source images in the style of the target back to the style of the source. CyCADA [81] uses the stylised source images to train a separate network for the classification task, whereas, Image-to-Image Translation [82] maps source and target domains to an intermediate representation. The intermediate representation is jointly trained to reconstruct source and target images, as well as to predict labels for a classification task.

These generative approaches are not best suited to classification in video, where there is a large amount of redundant and task-irrelevant information,. Therefore, our work in Chapter 4 and other video domain adaptation works [1, 2, 37, 43] adopt a discriminative approach to domain adaptation. The big benefits of generative methods are with the tasks of semantic segmentation [81, 82] and person Re-Id [96, 97, 98, 99]. For semantic segmentation, class labels are needed for individual pixels. Generative methods align domains on a pixel level, therefore, they are more suited for this task. For person Re-Id, different identities (labels) are used in both source and target domains. Discriminative methods do poorly if the label space differs between source and target, as aligning the distributions of each domain results in sub-optimal solutions.

2.3.4 Batch Normalisation Statistics

Batch Normalisation layers (BN) [100] were originally proposed to overcome the internal covariate shift, where the distribution of activations changes over training. The layer standardises the distribution of activations, such that the mini-batch statistics have a mean of zero and a standard deviation of one. BN layers are placed in many hidden layers of the network in order to overcome the covariate shift. For testing, population statistics of the training dataset set are used. BN improved the gradient flow to lower layers in the network, allowing the use of higher learning rates. This section explores works that exploit BN for domain adaptation.

Li et.al. [101] showed that the BN statistics from Caltech-256 [102] and Bing datasets [103] differ on an Inception-BN model pretrained on Imagenet [100]. This shift in mean and variance exists in both shallow and deep layers of the CNN. Their method, AdaBN, updates the population statistics of all BN layers with the statistics of the target domain as a post-processing step after training. This aligns the target domain to the standardised distribution seen during training. As

BN is applied across many layers of the network, AdaBN is able to learn a non-linear domain transformation due to the non-linear activation functions between layers. This method also has no hyper-parameters to optimise, which is beneficial for unsupervised domain adaptation as no target labels can be used for hyper-paramter tuning. Furthermore, they showed that AdaBN can be used with other domain adaptation methods such as CORAL [57].

Separate BN layers have been used for source and target distributions to align domains during training [6, 104, 105, 106]. These use a different batch normalisation layer for source and target, whilst training with source and target losses. Carlucci *et al.* [105] argued deeper layers learn more abstract, and ideally domain invariant, concepts, so should share BN layers. However, the low-level layers, which are domain specific, should still have separate BN layers. Their method, AutoDIAL uses a mixing parameter to control the extent to which distribution statistics of each BN layer should be shared across domains, which is learnt as training progresses. If distribution statistics are completely shared, AutoDIAL acts as a single BN layer, and if statistics are not shared, AutoDIAL acts as two separate BN layers.

2.3.5 Target Self-supervision

These methods jointly optimise an auxiliary task on the target domain and a task specific loss on the source domain. The auxiliary task only requires unlabelled target data and disentangles the representation of the target domain while learning the target task. Such tasks include reconstruction [107, 108], predicting transformations of images [109, 110, 111, 112] and encouraging features of individual examples to have a large norm [113].

Deep Reconstruction-Classification Network (DRCN) [107] uses target reconstruction as the auxilary task. A convolutional decoder maps the feature space to a reconstructed input space, and the mean squared error between the input and reconstructed images is back-propagated into the same feature extractor used for the source classification task. This provided competitive results with ReverseGrad [4]. Domain Seperation Network [108] uses a similar reconstruction task to DRCN, but use three encoders to disentangle the domain specific features from the shared features. Two encoders are domain specific and one encoder is trained with data from both domains. MMD is used to ensure the shared encoder has a similar distribution feature activations for both domains, and a difference loss encourages the domain specific features to be orthogonal to the shared encoder. A classification task is learned on the source domain with a combined representation from the source specific encoder and the shared encoder. A decoder is trained to reconstruct the target images with a combined representation of the target specific and shared encoders as input.

Predicting the transformation of input images has been used as a self-supervised task for domain generalisation [112, 114] and domain adaptation [109, 110]. Carlucci *et al.* [114] proposed a jigsaw puzzle task, which applied to multiple source domains, improved generalisation to unseen target domains. Input images were split into patches and the task was to predict the ordering of the patches that would reconstruct the image. This method was extended to include the rotation prediction task [112], where the image is rotated and the task is to predict the angle of rotation.

Self-supervised tasks are used for domain adaptation by applying the auxiliary task to unlabelled target data [109, 110]. Sun *et al.* [110] applied multiple tasks to unlabelled target data, predicting: rotation orientation, image patch location and whether the image was horizontally flipped. These self-supervised tasks were also shown to benefit adversarial training when jointly trained for semantic segmentation. Predicting rotation orientation improved partial domain adaptation performance [109], which combines the self-supervision objective with the adversarial training.

Xu *et al.* [113] found that target domains have substantially smaller norms than that of source domains. They propose HAFN as self-supervised task which penalises the norms of the source and the target examples being too different from a common scalar R. A large R is beneficial for domain adaptation but can cause exploding gradients at beginning of training, therefore they propose SAFN which incrementally increases R throughout training.

Predicting input transformations and enforcing a larger norm have been tested for partial domain adaptation [109, 113]. This is the setting where the source domain has classes not present in the target. Both works show, in this setting, that their methods outperform standard closed-set domain adaptation approaches [4, 73]. Bucci *et al.* [109] showed their approach can be combined with Partial Adversarial Domain Adaptation (PADA) [115] for further improvements. Partial domain adaptation is revisited in Sec. 2.3.7.

Our work in Chapter 4 proposes a self-supervised task for video that exploits the fact videos have multiple modalities. This improves domain adaptation performance for fine-grained action recognition and is evaluated on both the closed-set and open-set domain adaptation scenarios.

2.3.6 Self-training and Pseudo-labelling

Many Unsupervised Domain Adaptation (UDA) methods [6, 105, 106, 116, 117, 118] take inspiration from the semi-supervised learning literature [119, 120, 121] to adapt classifiers with unlabelled data from the target domain. A common approach is to enforce the cluster assumption on unlabelled data [119], with UDA methods ensuring that such that target examples should not lie near decision boundaries [105, 106, 116, 118]. This is achieved through entropy minimization on each unlabelled target example, x^t:

$$\mathcal{L}_e = \sigma(x^t)log(\sigma(F(G(x^t))) \tag{2.10}$$

F and G are the classifier and features extractor of a CNN, respectively, and σ is the softmax function, determining a probability distribution over the class labels.

Lipschitz Constraints

To enforce the cluster assumption further, Virtual Adversarial Domain Adapation (VADA) [116] provides a locally-Lipschitz constraint on the classifier. They use Virtual Adversarial Training [120] to ensure that the classifier response, $F(x)$, to input x, does not change with a small additive perturbation, $x + r$, by minimizing the KL-divergence between $F(x)$ and $F(x + r)$. The adversarial perturbation, r, is found by maximising the KL-divergence, such that the classifier response with the perturbed input is closer to a decision boundary. Drop to Adapt [117] uses a similar approach to provide a locally-Lipschitz constraint but uses an adversarial dropout mask instead of a perturbation.

Self-ensembling

Entropy minimization can also be considerd as a pseudo-labelling approach, where the network predictions of the current training step are used as labels to train the network. However, these predictions, or pseudo-labels, will be very noisy and error prone. Ensemble based approaches in semi-supervised learning [121, 122], reduce the noise of these labels by averaging predictions [122], or the models weights [121], over training steps. French *et al.* [6] (Fig. 2.4) used an ensemble approach for UDA. A student network is trained jointly on the source classification task and unlabelled data from the target domain. A teacher network, with a rolling average of the student's weights, generates pseudo-labels for the unlabelled target data. An unlabelled target image is passed through both the student and teacher networks, with separate augmentations, and the mean squared error between the predictions is used as the loss to train the network. At the time, this approach achieved state-of-the-art results on VISDA-2017 [88]. However, a class-balanced loss was needed to ensure that label predictions were uniformly distributed over the classes during training. In addition, the target loss was only applied to examples which the teacher classifier's prediction is above a threshold.

Co-Training [123] trains multiple classifiers and uses the predictions of each on unlabelled data to train the other. Mutual Mean Teacher [118] uses this approach with ensemble based learning for domain adaptation. Two sets of teacher-student networks are used, which similar to French

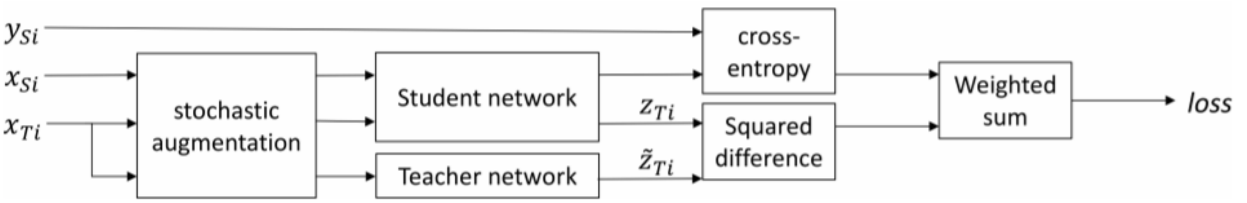


Figure 2.4: Self-ensembling for adaptation. A teacher network containing a rolling average of the students weights is used to provide pseudo-labels to train the student. Figure taken from [6]

et al. [6], the teacher's weights are a rolling average of the student. However, this model takes advantage of co-training where the teacher model is used to train the student model of the different teacher-student network.

Pseudo-label Assignment

Alternatively to entropy minimisation and ensemble methods, many methods use an iterative approach to domain adaptation [85, 104, 124, 125, 126, 127, 128]. First the target inputs are pseudo-labelled, this is to provide an estimate of the ground-truth label for a target instance. Next the model is optimised on target data to refine target label predictions with pseudo-labels [85, 104, 126, 129] and/or the class conditional structures are aligned across domains [124, 126, 128, 130].

Classifier predictions can be used to generate pseudo-labels. These soft-labels which are defined as the classifier score or posterior probabilities of classes [6, 118], or more commonly, a hard-label associated with the argmax of the classifier score [85, 104, 126, 127, 128, 129, 130].

Sener *et al.* [124] created a labelling function in feature space, where the target instances inherit labels from individual source instances. They adopt the largest margin nearest neighbour, which enforces a margin such that the similarity between a source point and the nearest target instance of the same class is larger than the similarity to the target instance of a different class.

Alternatively, the centroids of source features of a single class have been used for pseudo-labelling [125]. A target instance is assigned the pseudo-label as the class of the nearest centroid.

Kang *et al*. [130] used spherical k-means to adapt the source centroids to better fit the target distribution before generating pseudo-labels. K-means is first initialised with means of the source centroids, with k equal to the number of source classes. Target examples are pseudo-labelled by the class associated with the nearest centroid of the k-means classifier.

Deep clustering approaches have been used to produce pseudo-labels [127]. This allows the generation of labels that consider both the classifier predictions and a prior label distribution. Auxiliary labels are found by minimising the KL divergence between an auxiliary label distribution and a predictive distribution (classification scores). A prior distribution is enforced by an additional entropy term in the loss function that encourages label assignments to be uniform across class labels.

Pseudo-label Selection

To reduce the impact of incorrectly labelled pseudo-labels, a subset of reliable pseudo-labels can be selected for training. Approaches use the consensus of multiple classifiers [104], a threshold on the classification score [85], or distance from prototypical examples in feature space [125].

Asymetric Tri-training [104] exploits the consensus of two independently trained classifiers, F_1 and F_2 to select pseudo-labels. Pseudo-labelled target samples are selected if the predictions of F_1 and F_2 agree. In addition, the classification score of the prediction must exceed a threshold from a least one of the classifiers in order for the target sample to be selected. Training is conducted in an iterative procedure of labelling target data and training with those labels. This allows different target data to be selected in each iteration, with more accurate labels as the classifiers are adapted to target data. Progressive Modality Cooperation [129] uses the consensus of two classifiers trained on different modalities, *e.g*. RGB and Depth, if the classifier predictions from both modalities both agree on a pseudo-label with high classification scores, the target instance is selected for training.

The iCAN model [85] selects target pseudo-labels if they are both domain invariant and informative to the classification task. A pseudo-labelled target example is considered informative if the classifier prediction is above a threshold. Their threshold is adaptive requiring more informative examples throughout training by considering the average accuracy on the source domain. A Domain discriminator is used to determine if the target example is domain invariant. Target examples are selected if the domain discriminators logits are close to 0.5, *i.e.* if the network cannot predict the domain the examples is sampled from, signifying the example is aligned with the source.

Distanced based metrics are used in feature space to select pseudo-labels [125]. Progressive Feature Alignment (PFA) [125] pseudo-labels target instances based on the distance to the nearest source prototype. Prototypes are calculated as the centroid of source feature activations in each class. To reduce the impact from wrongly labelled pseudo-labels they propose and an Easy-to-Hard sampling strategy which trains with target videos close to the source prototypes before training with those further away. Gu *et al.* [128] modeled the distribution of distances to the class centroids as a gaussian-uniform distribution, using the posterior probability to determine a confidence associated with labelling target instances. The target loss used to train a classifier with the pseudo-label is weighted by the confidence of the labelling.

Target Refinement

The optimal classifier for the source domain may not be optimal for the target domain. Some works do not share the classifier weights between the source and the target domain such that the classifier is refined for the target [104, 116, 130, 131, 132]. Asymmetric Tri-training [104] trains a classifier only on pseudo-labelled target data, with separate weights from the classifier trained with source data. The source classifier is used to generate pseudo-labels. The target classifier is initialised from the classifier trained on the source.

Domain adaptation assumes that the source and the target domains are still highly related, so the optimal target classifier, $f_{\mathcal{T}}(x)$, should only differ from the source classifier, $f_{\mathcal{S}}(x)$, by a small perturbation, $\nabla f(x)$ [116, 131]. Residual Transfer Network (RTN) [131] models this as a residual function: $f_{\mathcal{S}}(x) = f_{\mathcal{T}}(x) + \nabla f(x)$. The residual is learnt by the source classification loss on $f_{\mathcal{S}}(x)$ and an entropy loss with target data on $f_{\mathcal{T}}(x)$.

Dirt-t [116] refined a classifier that has been pre-trained on the source domain. Entropy minimisation on target data is used to refine the classifier to the target domain over a series of training steps. In order to avoid decision boundaries moving too far, they propose a regulariser that limits the ability of the classifier response to change significantly between training iterations. This is achieved by the KL divergence between the current classifier response and the classifier response before the gradient descent step.

Class Conditional Alignment

As opposed to self-training, pseudo-labelling is used to align the class conditional distributions across both domains. UDA assumes there is no access to target labels during training, therefore pseudo-labels are used as an estimate for class labels. Once the target domain is pseudo-labelled, the distribution of each class across the source and the target domains can be aligned.

Given that the target domain is pseudo-labelled, triplet losses can be used between source and target examples to align domains [124]. For each source instance, the distance to the nearest target instance of the same class in minimized, whilst the distance to the nearest target instance of a differing class is maximised. Alternatively, the centroids from each class can be used to align domains [125, 126, 133]. Using the centroids reduces the impact of incorrectly labelled pseudo-labels. To overcome the impact of small mini-batch sizes on the estimate of the centroids for each domain, MSTN [133] maintains a rolling average of the centroid feature representation that is updated each training step.

Recent approaches have aligned centroids in RKHS using the Maximum Mean Discrepancy (MMD) measure per class [130, 134]. Transferable Prototypical Networks (TPN) [134] uses MMD between three kernel mean embeddings calculated from the source instances, target instances and the mean of both source and target instances combined. The Contrastive Domain Discepancy (CDD) measure [130] aims to minimise the intra-class domain discrepancy and maximise the inter-class domain discrepancy. This metric minimises MMD between the source and target distributions of the same class and maximises MMD between all source and target distributions of differing classes.

Weighted Maximum Mean Discrepancy [132] uses an instance re-weighting approach to correct for the different label proportions in source and target domain. First the target domain is pseudo-labelled and the proportions of source and target instances per class are calculated. The source instances are weighted such that the density of the re-weighted source and target distributions match. Along with the other conditional alignment approaches using pseudo-labels [125, 125, 126, 126, 133, 133], optimisation iterates between pseudo-labelling and alignment.

Conditional alignment methods using adversarial training strategies [84, 86] were discussed in Sec. 2.3.3. These used the classification scores directly as soft-labels, to approximate the joint distribution [84] or as a weighting for multiple domain discriminators [86]. Joint distribution alignment approaches using MMD [74] or optimal transport [77] were discussed in Sec. 2.3.2.

Target Clustering

Other works aim to maintain discriminative structures in target domain during alignment of classes by clustering the target domain. Instances assigned to the same cluster are encouraged to be closer in feature space [91, 126, 130]. Approaches cluster the target domain to produce pseudo-labels that better fit target data [130], and encourage target pseudo-labels to be closer in feature space [91, 126]. Deng *et al.* [126] used a margin loss to encourage target examples assigned to the same pseudo-label to be close in feature space whilst maximising the distance between target examples with differing labels. This ensured the class conditional structures are clearer before aligning domains by minimising the Euclidean distance between source and target centroids of the same class. Tang *et al*. [127] encouraged pseudo-labelled target instances to move closer to their class centroid. A softmax distribution over the distances to all class centroids is obtained. The entropy of this distribution was minimised to encourage each class to have tight clusters. A similar loss is used on the source domain to produce tight clusters for source features.

In Chapter 6, we consider aligning the class conditional structures for domain adaptation on a highly imbalanced dataset, using a pseudo-labelling approach. The approaches outlined in this section do not test on highly imbalanced datasets that occur during uncurated data collection. Some works in this section have also exploited that those datasets have near uniform class distribution to avoid degenerative solutions [6, 91]. Instead, our work proposes a prototype based sampling strategy to increase the diversity of class assignments, as well as a robust confidence measure to select reliable pseudo-labels.

2.3.7 Beyond Closed-set Domain Adaptation

The works in the previous section have assumed that the same set of labels are used in the source and target, *i.e*. $\mathcal{Y}^s \neq \mathcal{Y}^t$. However, in more realistic scenarios this will not be the case. There could be labels present in the source that are not in the target (partial domain adaptation), or labels in the target that are not in the source (open-set domain adaptation). Other tasks, such as cross-modal retrieval, often evaluate on labels not seen during training. This section focuses on methods for classification and the following section will address domain adaptation for retrieval.

Adversarial training has been adapted to the partial domain adaptation setting by weighting the source losses differently for each source class [115, 135]. A weight vector, γ, approximates the likelihood that each source class is present in the target domain, and is determined by the label predictions, $\hat{y}_i^t$ from all n target examples. Both the domain discriminator and task classification losses are on source examples from the kth class are weighted by γ_k. Partial Adversarial Domain

Adaptation (PADA) [115] calculates the weighting for the kth source class as the mean of the target classification scores: $\gamma_k = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_{ik}^{t}$, and is updated every 500 training steps. To evaluate their method they form partial domain adaption benchmarks by adapting closed-set domain adaptation benchmarks. Less than half of the classes from target domains are used, but keeping all classes from the source domains. They showed large improvements on Office-31 [60] and Office-Home [78] datasets, and smaller improvements on ImageNet-to-Caltech [115] and VISDA-2017 [88]. Their method benefits from the fact these datasets are unnaturally balanced. With a large class imbalance, minority classes would be unfairly down-weighted despite their presence in the target domain.

In open set adaptation the target domain can contain labels unseen in the source [136, 137]. In these works a new class is defined for all unknown classes in the target. The difficulty is to learn a decision boundary for this unknown class without labelled data. The seminal work [136] proposed to map the source to the target by assigning each target example to a class, including the unknown class. A cost is defined for assigning a class as the distance of a target example to the centroid of source examples for that class. The unknown class has a fixed cost which is a hyper-parameter to be tuned. The assignments can be found as a linear optimisation to find an assignment to minimise total cost, subject at least one target example must be assigned to each class and all target examples must be assigned a class or unknown class. This work was extended by Saito *et al.* [137] for adversarial alignment. The classifier was to trained to predict class labels on source inputs, but the unknown class for target instances with a probability t, where $0 < t < 1$. Similar to ReverseGrad [4] a Gradient Reversal Layer (GRL) was placed between the feature extractor and the classifier, so the feature extractor attempts to maximise the loss on target instances. Note that the GRL was only activated for target instances. If the classifier predicts a target instance as unknown with probability less than t, the feature extractor will align the target instance with a source class. If the classifier predicts the target instance as unknown with a probability greater than t, the feature extractor is optimised to maximise the unknown class probability. To evaluate their methods the

Office-31 dataset [60] was adapted for open-set classification, all classes were present in the target but only 10 were present in the source. Both methods [136, 137] outperformed solutions tailored for closed-set recognition [4] with the adversarial approach to open set recognition showing the largest improvements.

More recent work have aimed a creating a solution that will work in both the partial and open-set domain adaptation setting, where there is no prior information on the label space of the target domain [138, 139]. Universal Domain Adaptation finds separate weights for target and source instances. Each weight represents the probability that a given example is from a shared class. A domain classifier is trained to distinguish between source and target examples. If the domain of the source or target instance is predicted incorrectly, the instance is more likely to belong to a shared class and is given a higher weighting. At test time, if the classifier response does not exceed a threshold, then the instance is predicted as the unknown class. DANCE [139] proposed to cluster target examples with Neighbourhood Clustering (NC), which brings visually similar target examples closer together without specifying the number of classes in the target domain. NC minimises the entropy over a distribution of similarities of one target example to all other target examples and source class means. This encourages target examples to increase the similarity to the few target examples likely from the same class, by move further away other examples. To compute the similarity between target examples, all target examples cannot be used in one forward pass through their network, therefore they store features in a memory bank.

Self-supervised methods discussed in Sec. 2.3.5 were shown to improve partial domain adaptation, as well as the conditional alignment approach MADA [86] in Sec. 2.3.3. The multi-modal self-supervised objective in Chapter 4 is evaluated in an open-set domain adaptation setting where the source and target can contain nouns and verbs that are not shared between the domains. In Chapter 6 a conditional alignment approach is used for open-set adaptation in the context of cross-modal retrieval.

2.4 Conclusion

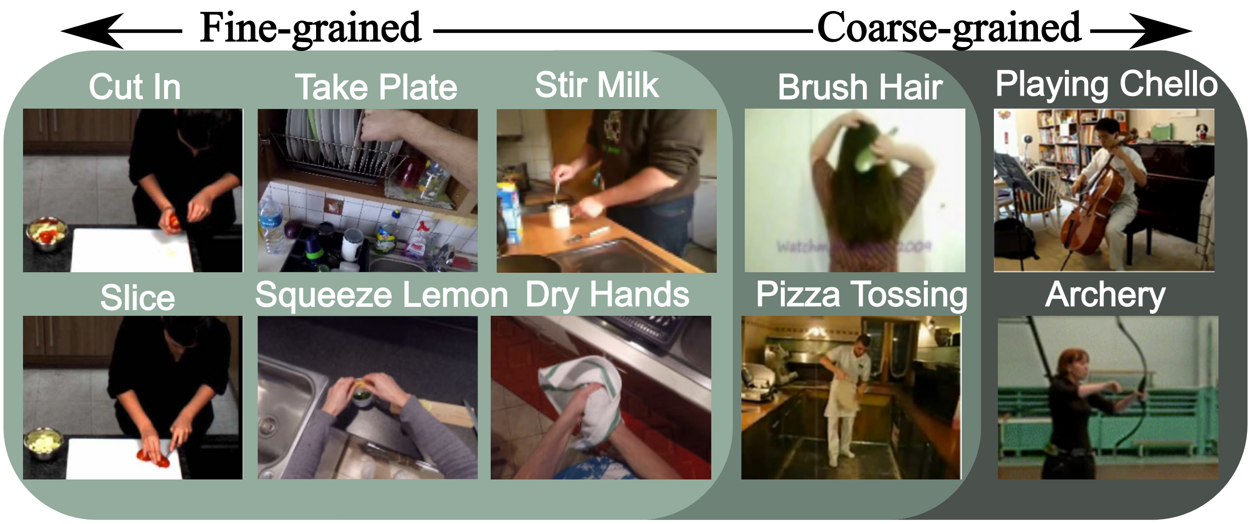
This chapter provided the definition of a domain shift, introduced the problem of unsupervised domain adaptation, and offers related work to all technical chapters in this thesis. In particular, Sec 2.3 provided an extensive review of domain adaptation works applied to deep learning architectures, which outperform the shallow-based approaches described in Sec 2.2. Our work in Chapter 4 extends adversarial training, described in Sec 2.3.3, to video based architectures, and incorporates a self-supervised objective for domain adaptation, with inspiration from works described in Sec 2.3.5. Our method is compared to baselines provided in Sec 2.3.2 and Sec 2.3.4. Chapter 6 explores class conditional alignment for text-to-video retreival, with inspiration from works in Sec. 2.3.6. Similar to the works in Sec 2.3.7, all technical chapters evaluate in an open-set setting. The next chapter provides a review of video understanding datasets and models, before outlining the related work for domain adaptation in video.

Chapter 3

Literature on Action Understanding in Video

An action is what someone does as they interact with their environment, in order to achieve a desired goal. These describe movement through space, *e.g. walking, running and gestures*, as well as interactions with object and humans, *e.g. kick football, wash frying pan and shake hands*. Course-grained actions or activities describe movement over long periods of time, such as *playing football, braiding hair* and *making a salad*, whereas fine-grained actions describe movement or object interactions that last a matter of seconds, *e.g. kick football, pick up hair brush, cut tomato*. Fig. 3.1 visualises several actions in video datasets, which lie on the spectrum from fine-grained to course-grained.

In addition to the coarseness of the visual representation of actions, descriptions can vary in their level of detail. Fig. 3.2 shows different possible descriptions for a single fine-grained action. A less detailed description, *add salt*, could describe a large number of different fine-grained actions, *e.g. grinding salt into pan*, *pouring salt into pan*, whereas the space of possible videos reduces with more detail. The different descriptions also arise from the ambiguity of verbs, as



e.g. MPII Cooking, EPIC-KITCHENS, Charades, Breakfast e.g. HMDB, UCF-101, ActivityNet, Kinetics

Figure 3.1: Visualisation of fine-grained and coarse-grained actions

many verbs can be used to describe the same action.

A variety of datasets exist for action understanding with different levels of granularity and descriptions of actions (Sec. 3.1). Key action understanding datasets are highlighted in Sec. 3.1, with a focus on describing the biases within the datasets. This is followed by a more in-depth discussion of coarse fine-grained action understanding datasets, explaining that the unscripted nature of dataset collection leads to a large location bias. This will provide better context of fine-grained action understanding, and highlight the need for domain adaptation research for this problem.

Action recognition models predict the class belonging to a video (Sec. 3.2). To improve action recognition performance on videos from a different domain to those used during training, domain adaptation methods have been tailored for video (Sec. 3.3). Sections 3.2 and 3.3 provide related work for Chapters 4 and 5, where we explore domain adaptation for action recognition. Sec. 3.2 is used to provide an overview of the different architectures available for video. Sec. 3.2.3 and 3.2.5 are most relevant to Chapter 4 as these outline multi-modal architectures and self-supervision for video. The remaining sub sections in Sec. 3.2 are generic for video, which provide context for

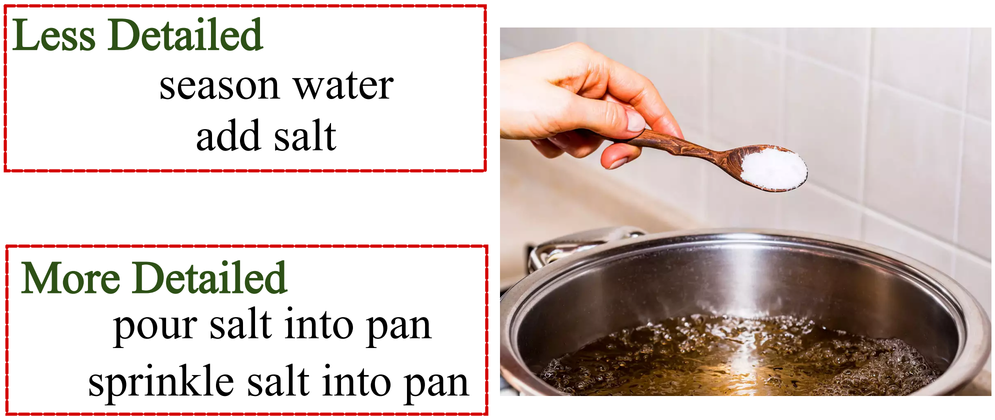


Figure 3.2: Fine-grained actions can be described using different level of granularity

all technical chapters. A comprehensive comparison of domain adaptation literature for action recognition is provided in Sec. 3.3.

Other tasks utilise free-form descriptions of actions. Cross-modal retrieval methods aim to retrieve video depicting actions that are semantically related to a free-from textual query, or vice-versa (Sec. 3.4). Few domain adaptation works have addressed the problem of cross-modal retrieval, where the captioned videos during training are not from the same domain as the videos used for retrieval (Sec. 3.5). Sections 3.4 and. 3.5 provide related work for Chapter 6 where we explore domain adaptation for cross-modal retrieval.

3.1 Datasets for Action Understanding

This section summarises the different datasets for modelling actions in video. First, several coarse-grained action recognition datasets that provided key breakthroughs for video understanding are discussed (Sec. 3.1.1). This is followed by a comparison between fine-grained action recognition datasets, with a emphasis on the different dataset collection strategies (Sec. 3.1.2). Lastly, several key datasets that include textual descriptions of actions instead of action labels are discussed

(Sec. 3.1.3).

3.1.1 Coarse-grained Action Recognition

The earliest datasets for action recognition were captured in controlled conditions, with very little camera motion and background clutter, from a fixed viewpoint [140, 141, 142]. Actors performed basic activities [140] (*e.g.* walking, running, jogging) with little variation of how they were performed. To create datasets more indicative of real world scenarios, videos were collected from sports broadcasts [143], movies [144] and YouTube [145]. This increased the diversity of recording conditions and intra-class variability of the activities, however, the datasets were still small with only 10-12 action classes. The following datasets (UCF-101 [25], HMDB51 [146], ActivityNet [147] and Kinetics [26]) allowed significant breakthroughs for action recognition by scaling course-grained datasets to a large number of videos and classes:

UCF-101 [25] contains unconstrained videos collected from YouTube. Videos have cluttered backgrounds, occlusions, camera motion and a variety of different lighting conditions and camera viewpoints. Unlike previous datasets [144] that used movie clips to increase the diversity of actions, most videos are not performed by actors and contain a mix of videos that are shot by amateurs and professionals. The dataset contains 101 classes, adding an additional 51 classes ontop of previously collected datasets collected by the authors [145, 148]. Along with HMDB [146], these datasets were the primary benchmark for action recognition for many years.

The authors grouped classes into 5 main categories: *Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments and Sports*, however, many actions can be recognised based on their appearance without any motion information. For example, the detection of a piano in an image is sufficient to recognise the action: *Playing Piano*. Without any motion information, a 2D-CNN trained on RGB images can achieve over 70% on the dataset [18]. There is a strong environmental bias in the dataset. The presence of a river or snow would be sufficient to classify *Rowing* or *Skiing* activities. Masking out humans in the video with a black box can achieve 47% accuracy on the dataset [34], despite only using background features from the scene.

HMDB51 [146] contains videos from 51 classes collected from movies and online sources. Actions are less environment specific than UCF-101 and focus on more generic activities describing human movement. Classes are grouped into five main categories: *General Facial Actions, General Body Movements, Body Movements with Object Interaction and Body Movements for Human Interaction*. In addition, actions were not pre-determined before data collection. For annotation, students were asked to watch videos and annotate any segment where there was a single non-ambiguous human action exceeding a minimum quality standard.

ActivityNet [147] collected videos from YouTube, searching for a pre-determined taxonomy of activity classes. The aim was to ensure that the dataset contains a diverse set of action that are more representative of every day life. Activities are described in 4 layer hierarchy, with 203 activity classes in the lowest level. The top level activities are: *Personal Care*, *Eating and Drinking*, *Household*, *Caring and Helping*, *Working*, *Socializing*. and *Leisure and Sports and Exercises*. Amazon Mechanical Turk was used to ensure that videos selected from YouTube contained an activity, and were later asked to annotate temporal bounds. On average dataset collection produced 1.4 activities per video, therefore they proposed benchmarks that go beyond trimmed activity recognition. They define untrimmed activity recognition as the task of predicting the activity without the start and end times available, and action detection as the task of predicting the start and end-times of all activities in an untrimmed video.

Kinetics [26] also collected videos from YouTube, but on a larger scale. The initial dataset contained 400 action classes and over 15 times as many videos as the previous largest action recognition dataset with manual annotation, ActivityNet. This dataset was intended for training large-scale deep learning architectures, to provide similar performance gains for action recognition as ImageNet did for image recognition. To date, Kinetics-400 and the extension of the dataset, Kinetics-700 [149], are the go-to datasets for evaluating video architectures.

Despite the large dataset collection, there is a strong bias towards visual features. Li *et al.* [35] showed that Kinetics has a large bias towards the appearance of objects, scenes and people. This bias was greater than both HMDB51 and UCF101, and slightly more than ActivityNet. The score determined visual bias by considering the performance of a linear classifier trained on images features from off-the-shelf, object, scene and person attribute detectors.

3.1.2 Fine-grained Action Recognition

Exploiting internet databases, such as YouTube, is not suitable for collecting fine-grained actions, as the meta-data attached to videos that describe the contents of the video are often too course-grained. The camera setup from online databases is also inappropriate for many applications, *e.g.* the majority of online content is taken from a third person perspective, that will be of little benefit to egocentric applications.

Compared to course-grained datasets, collecting and annotating fine-grained actions requires a large annotation effort. Figure 3.3 shows the recent surge in large-scale fine-grained action datasets. Two approaches have attempted to achieve scalability: crowd-sourcing scripted actions [7, 8, 9], and long-term collections of natural interactions in homes [11, 15, 41]. While the latter offers more realistic videos, many actions are collected in only a few environments.

For datasets with natural interactions, EPIC-KITCHENS [10] is the largest, with the most

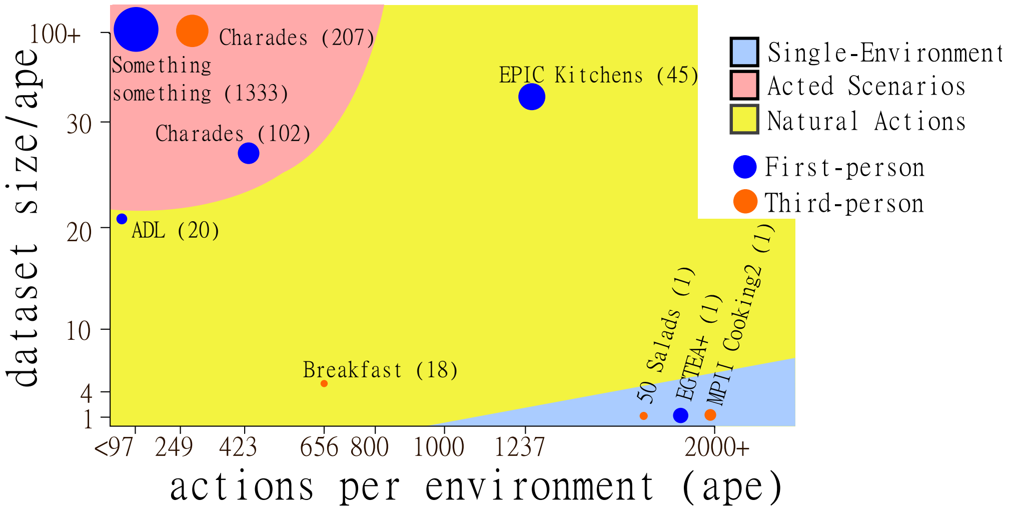


Figure 3.3: Fine-grained action datasets [7, 8, 9, 10, 11, 12, 13, 14, 15], *x-axis:* number of action segments per environment (ape), *y-axis:* dataset size divided by ape. EPIC-Kitchens [10] offers the largest ape relative to its size.

diverse number of environments. Despite this attempt to scale to large number of diverse actions, there is still a strong environmental bias as the dataset only contains 37 different participants and 45 locations.

Crowd-sourcing

Crowd-sourcing platforms, such as Amazon Mechanical Turk (AMT), can be used to generate videos of scripted actions. A set of actions are pre-determined and AMT workers are asked to record themselves act out the actions. Note that actions must be able to be performed by the general public, therefore crowd sourcing may be insufficient for a number tasks, such as operating scientific or military machinery.

Charades [8] was the first dataset to use crowd-sourcing to generate both labels and videos. AMT workers acted out house-hold activities from scripts written by a different set of AMT workers. Scripts were generated from a set of verbs and nouns, selected by analysing movie scripts from house-hold scenes. The most common actions (verb-preposition-noun triples) in the scripts were grouped into 157 action classes, and workers annotated the temporal bounds for actions present in the videos. In total, 267 workers generated 10,000 videos from their own homes, with an average duration 30 seconds per video. On average, there are 6.8 actions per video, each lasting 12.8 seconds.

Charades-Ego [9] uses the scripts from Charades [8] to crowd-source 4000 videos of paired first and third person footage. Each video is on average 31.2 seconds long, with an alignment error between first person and third person of 1.2 seconds. Participants were asked to hold a camera to their forehead, or improvise their own head mount, therefore many actions are unnatural as the actors use one hand. The dataset contains 68,536 actions from the same 157 classes as Charades.

Something-Something uses a similar approach to Charades [8] for dataset collection. Their dataset is much larger and more fine-grained, but less natural than Charades, with workers acting a single object interaction, *e.g. spilling smth onto smth* and *opening smth*. The dataset focuses on learning verbs from videos, with all nouns replaced with *smth* in the annotation. In total, 1133 workers generated 108,500 videos. The average duration of a videos were 4 seconds.

Towards Unscripted Actions

In order to collect footage of natural actions, participants are filmed over longer periods of time. All natural interactions between the participants and their environment are observed over a recording period. This comes at a cost, as the lack of scripting increases the annotation effort. It is expensive to exhaustively annotate a large number of short actions, as well as their temporal bounds. Some datasets collected video from a fixed camera [12, 13], while others use an egocentric viewpoint to provide a clear viewpoint of object interactions. Often a single lab controlled environment [13, 14, 15, 31] is used to collect videos, although some datasets have increased the diversity of scenes by collecting data in participants homes [10, 11, 12, 30].

Many datasets collect videos in a single lab controlled environment [13, 14, 15, 31]:

GTEA Gaze [31] recorded unscripted activities during meal preparation from 14 different participants. However, the environment consisted of a single kitchen table and the food and objects were constrained to 30 different kinds. They collected 94 unique actions, however due to the long-tailed distribution of unscripted activities, they only considered the 25 actions that occur multiple times in the training sequences and at least once in test.

EGTEA+ [14] includes unscripted actions from 32 participants from 7 meal preparation tasks: American Breakfast, Turkey Sandwich, Cheese Burger, Greek Salad, Pizza, Pasta Salad, and Afternoon Snack. Filming took place in a single kitchen, and natural actions were annotated with approximately 200 different verbs, e.g. putting, pouring, flipping, dividing, *etc*. Videos were taken from a egocentric viewpoint with SMI eye-tracking glasses, producing around 29 hours of video footage.

50-Salads [13] recorded 27 participants preparing two mixed salads from a wall mounted camera above the same counter-top. Only the participant's hands and arms are visible in the videos. Annotations consists of three activities (*cut and mix ingredients, prepare dressing, serve salad*), broken down into 17 fine-grained actions (*e.g. peel cucumber, cut cheese, add salt*).

MPII-Cooking 2 [15] recorded 30 participants preparing a single meal, starting from a clean and empty kitchen. Eight different third person viewpoints are available for most videos. Annotations consists of 59 composite activities (*e.g. preparing onion*) broken down into 67 fine-grained actions (*e.g. move, cut, stir*) paired with manipulated objects (*e.g. pan, onion, knife*). Scripts from recipe books were used to obtain the taxonomy of composite activities, but this collected independently from the collection of videos.

Some datasets have attempted to scale datasets to multiple environments by recording in participants homes [10, 11, 12, 30]:

Activities of Daily Living (ADL) [11] collected house-hold actions from a chest-mounted camera, across 20 participants in their own homes. Participants were asked to record 30 minutes of uninterrupted morning activity, which contain some of the 18 pre-determined action classes. Examples include: *making coffee, brushing teeth, watching tv, laundry, using computer*. The actions are more visually distinct and less fine-grained than other datasets [12, 13, 14, 15, 31, 41], and lasted on average 100 seconds.

Breakfast [12] recorded 52 participants in 18 different kitchens. Videos contain footage from 10 cooking activities (*e.g. preparing coffee, fruit salad or scrambled eggs*), annotated with fine-grained actions (*e.g. pour milk, take plate, crack egg*). Footage is taken from a third person viewpoint from different positions in each kitchen. Compared to other datasets with a fixed viewpoint [13, 15], the actions are less visible with more occlusions.

EPIC-KITCHENS [10, 30] is the largest dataset of unscripted, natural actions. The initial dataset, EPIC-KITCHENS-55 [30], collected videos from 32 participants in their own homes. Unlike other datasets of natural actions [11, 12, 31], activities were not specified in advance. All household activities were recorded that occurred in a three-day recording period, including laundry, washing-up and cooking. 45 hours of videos were collected from a head-mounted GoPro at full-HD.

Annotations used a *'live commentary'* strategy, where participants re-watched their videos and described fine-grained actions as they happened, in an audio file. Live commentary encouraged annotators to densely annotate each video, while providing a rough time-stamp for each action. In order to produce accurate annotations, participants described actions in their native language. Amazon Mechanical Turk was used to transcribe audio files into textual narrations and to label start and end-times of each action. Verbs and nouns in the transcriptions were then parsed and grouped into verb and noun classes.

The extension, EPIC-KITCHENS-100 [10], added 55 hours of footage, from 16 of participants in the original dataset collection and 5 new participants. This was collected two-years later, with a different GoPro version, and half of the original participants had changed home. This produces a domain gap between the original footage and the extension. Chapter 5 outlines the Unsupervised Domain Adaptation Challenge for the dataset.

In total EPIC-KITCHENS-100, contains 90K labelled actions, across 97 verb and 300 noun classes.

3.1.3 Description Datasets

Instead of a single action class, some datasets contain free-form textual description of actions. These can be used for a variety of tasks, including video captioning and text-to-video retrieval.

Most datasets obtain videos by querying the YouTube on a specific topic. The videos are segmented into short clips depicting an action or activity, and a textual caption is paired with the video clip. For example, YouCook2 [150] contains cooking videos collected from YouTube. Videos are split into clips depicting the procedures (a series of fine-grained actions) used to create a meal, and each clip is annotated with sentence (*e.g. add a bit of Worcestershire sauce to mayonnaise and spread it over the bread.*). Other popular datasets used to evaluate cross-modal retrieval tasks include MSR-VTT [151], which contains videos from more general categories, *e.g. vehicles, sports and TV shows*, and HowTo100M [152] which contains instructional videos. HowTo100M transcribes speech in instructional videos to generate textual captions, allowing the dataset to be significantly larger than other description datasets (170x more videos than MSR-VTT). However, the annotations are weakly aligned with the video, and the action described may not be present in the video. Despite this, models trained on this datasets can generalise well to other datasets. Learning a cross-modal retrieval model on HowTo100M and evaluating on YouCook2 outperforms the same model trained on YouCook2. Note that there is little domain shift between YouCook2 and HowTo100M, as both contain cooking vidoes collected from YouTube, and some videos are common to both datasets.

Other description datasets utilise videos from action recognition datasets. TACoS [153] and Charades-STA [154] attach sentence level descriptions to a subset of videos from MPII-Cooking2 [153] and Charades [8], respectively. EPIC-KITCHENS-100 [10] collected annotations by asking participants to narrate their videos. These narrations have been used as a companion text for cross-modal retrieval tasks. Chapter 6 uses the domain shift present in EPIC-KITCHENS-100 and the text narrations to evaluate our proposed domain adaptation strategy for text-to-video retrieval.

3.2 Action Recognition Models

Action recognition is the task of assigning a class label to what someone is doing in a video. Historically, action recognition relied on hand-crafted features to represent salient regions of the video useful for detecting actions (Sec. 3.2.1). Due to the success of deep learning for image classification [69], handcrafted features were replaced with Convolutional Neural Networks (CNN) which learn features jointly with the action classification task. Two approaches were used to include motion information into CNNs: extending convolutions over the temporal dimension (Sec. 3.2.2) or with a consensus of networks trained on RGB and optical flow displacement vectors (Sec. 3.2.3). These approaches were combined with consensus of multiple frames throughout the video to model longer temporal structures (Sec. 3.2.4). In the absence of labelled data, temporal and multi-modal properties of video are exploited in self-supervised objectives representations suited for action recognition (Sec. 3.2.5).

Section 3.2.1 provides an historical background for action recognition before deep learning. Sec. 3.2.2, Sec. 3.2.3, Sec. 3.2.4 outline models

3.2.1 Handcrafted Features for Action Recognition

Historically, action recognition consisted of a pipeline of distinct stages. First, local features describing salient regions of a video are extracted [155, 156, 157, 158, 159], and encoded to create a fixed length descriptor of the video [160, 161]. A learning algorithm, such as an SVM, is trained on these descriptors to find decision boundaries separating different actions. This section provides an overview of feature extraction and encoding methods for action recognition. Later sections will consider deep learning approaches that model all stages of the pipeline using a neural network.

Feature Extraction

Initial works extended interest point detectors to the temporal dimension to detect salient regions in a video [155, 156, 157]. The seminal work, Space Time Interest Points (STIP) [155] adapts the Harris corner detector method to include temporal information, detecting regions with significant changes in pixel intensity across time, as well as spatially. This captures salient regions that are invariant to translation, rotation and illumination changes, which also have a non-constant motion across the video. Other detectors have been proposed that considered the second order derivatives of pixel intensities [157], and the use of separate filters for spatial and temporal dimensions to increase the number of detected interest points [156].

Descriptors are used to specify the motion and appearance information at each interest point. Optical flow is commonly used to capture motion information, which describes the apparent motion between frames. Laptev *et al.* [162] detected interest points using STIP and compute descriptors over multiple spatial and temporal scales. For each scale, a histogram of orientated gradients (HoG) and optical flow (HoF) are computed over a 3D volume, around the detected point to describe appearance and motion information, respectively. Klaser *et al.* [163] generalised HoG to 3D, computing a histogram of the gradients with respect to both the spatial and temporal dimensions. This captured motion information without the expensive computational cost of optical flow.

Later action recognition methods used dense representations to describe a video [158, 159, 164]. Wang *et al.* [164] extracted features at regular spatial-temporal intervals instead of interest points. This improved action recognition performance on UCF-sports [143] and Hollywood2 [144] datasets, over extracting features only at key-point detection [155, 156, 157]. They also show that HoGHoF [162] features outperform 3D-HoG [163] on these datasets.

Dense Trajectories (DT) [158] use optical flow to track densely sampled points through 15 frames of a video. Multiple descriptors, HoGHoF [162] and MBH [165], were computed along the trajectories, instead of a fixed 3D volume to account for the change in spatial locations as an object moves. They re-purpose the MBH descriptor for action recognition, which captures the changes in motion of the optical flow field. This is computed as the histogram of gradients of the optical flow field, and reduces the effect of background motion. MBH was the best performing descriptor on all evaluated datasets (KTH, YouTube, HollyWood2 and UCF-sports), showing improved performance with all descriptors combined with the DT.

Improved Dense Trajectories (iDT) [159] corrects for camera motion when computing dense trajectories. The vectors found by matching SURF [70] features across frames, as well as vectors of the optical flow field are used to provide correspondence between frames. RANSAC is then used to warp the optical flow field, suppressing the camera motion. This improved the quality of the descriptors, and trajectories could be removed that were similar to the camera motion. The performance of the HoF [162] descriptor was improved most by their method, as the warped optical flow field is less affected by camera motion.

Feature Encoding

Commonly, an SVM is used as the learning algorithm to find decision boundaries between actions in feature space, but this requires a fixed length input. Feature encoding methods extract a fixed length descriptor by clustering the videos descriptors into groups. The collection of the groups is

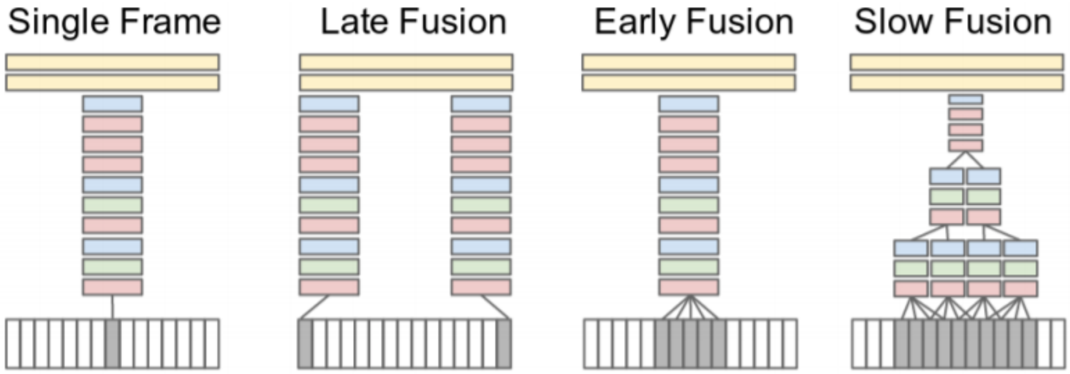


Figure 3.4: Different options to combine temporal information into a CNN. Figure taken from [16]

called a *visual codebook* and each group is refered to as a *visual word*. The majority of action recognition works [158, 162, 163] used the Visual Bag of Features [160] (BoF), which stores the number of occurrences of visual words in local regions of the image. The codebook is found by clustering the data into k visual words using k-means. Each local discriptor is assigned the closed visual word, and a histogram of the word occurrences creates the encoded feature descriptor. Improved Dense Trajectories [159] achieved increased performance using Fisher vectors [161], which fits a Gaussian Mixture Model (GMM) to find visual words. The gradient of the log-likelihood of the GMM is used to represent an image. This captures the 1st and 2nd order statistics of the distribution of local features around visual words. Fisher vectors [166] require a much smaller codebook than BoF [162], increasing computational efficiency, and the embedded descriptor can be trained with a linear SVM, whereas BoF benefited from a non-linear SVM to separate actions. The downside is that visual words produced by Fisher vectors are much larger in size than BoF.

3.2.2 Spatio-temporal Neural Networks

Karpathy *et al*. [16] explored how to include temporal information into the AlexNet [69] architecture used for image recognition (Fig 3.4). They proposed three strategies: early, late and slow fusion. Early fusion modifies the filters of the first convolutional layer to span the entire temporal dimension as well channel dimension. Late fusion combines the activations from two single-frame networks spaced 15 frames apart, which are fed into a fully connected layer. Slow-fusion extends all convolutional layers to convolve across-space and time. Early fusion resulted in a worse performance than the single frame architecture and late fusion provided similar performance to the single frame architecture. Slow-fusion showed a small improvement over the single-frame architecture, highlighting the benefit of fusing temporal information gradually. However, all methods still did not utilise the temporal dimension effectively, and the slow-fusion model was not competitive with traditional methods using iDTs [167].

C3D [168] shows that deeper 3D convolutional architectures can be competitive with the best performing methods using iDTs [167]. This uses eight convolutional layers with zero padding to preserve the size of the temporal dimension in higher layers of the network. In comparison, the slow fusion model proposed by Karpathy *et al.* loses all temporal information by the third convolutional layer. The work also experiments with different kernel sizes for the temporal dimension. They show that using a kernel size of three for both the spatial size and temporal depth of the kernel achieves the best performance. C3D improves on the slow fusion model of Karpathy *et al.* by 20%. Their qualitative results show that the architecture first focuses on the appearance information for the first few frames and then focuses on the salient motion.

Long-term Temporal Convolution (LTC) [169] increases the number of frames used as input, beyond the 16 frames used by C3D. In order to keep a similar network complexity, the spatial resolution of the input image was reduced. Increasing the input temporal dimension from 16 to 60 increased action recognition performance on UCF-101 by 8%. Increasing spatial resolution showed improvements on UCF-101, however the improvement gain was only significant with networks using few frames as input.

To exploit advances in image-based architectures, Carreira *et al.* [26] inflated all convolutional and pooling layer of a 2D-CNNs to three dimensions. This has the added benefit of using ImageNet pretrained models for video architectures by repeatedly copying the ImageNet weights across the temporal dimension. They inflate the Inception-v1 architecture, which uses less than half the parameters as C3D and substantially improved performance on UCF-101 [25], HMDB [146] and Kinetics [26].

A large improvement for deep learning action recognition models was the development of large-scale video datasets. Karpathy *et al*. [16] created the Sport-1M dataset, annotating YouTube videos with labels automatically generated from the textual metadata. Action recognition performance of their slow-fusion method on UCF-101 was improved by over 20% by pretraining on Sports-1M, than training from scratch. Carreira *et al*. [26] proposed the Kinetics dataset for the purpose of providing a large-scale dataset for pretraining video architectures. Although smaller in size than Sports-1M, the manual annotation used in Kinetics provided cleaner labels, resulting in improved performance when fine-tuning on UCF-101 [25] and HMDB [146].

Several works have reduced the computational complexity of 3D-CNNs by factorising convolutions. 3D convolutions can be separated into a 2D convolution to learn spatial information followed by a 1D convolution over time [170, 171]. The performance of factorised convolutions [171] is comparable to I3D [26], but greatly reduces the number of parameters. Channel Separated Network [172] factorises out the spacio-temporal interactions from the channel interactions. A 3D convolution is applied to each channel dimension separately, followed by a point-wise, 1x1x1 convolution to learn interactions over the channels. This reduces computational complexity and increases the performance compared to 3D convolutions. To increase performance further, design choices of 3D-CNNs can be optimised through an architecture search [28]. Parameters of the input video (temporal duration, frame rate and spatial resolution), and the number of convolutions layers and filters are optimised for maximum performance on Kinetics [26], for a given computation cost budget.

Convolutional layers have a small receptive field, so are unable to capture long-rage dependencies of pixels across both spatial and temporal dimensions. The Non-local Block [17] (Fig. 3.5)

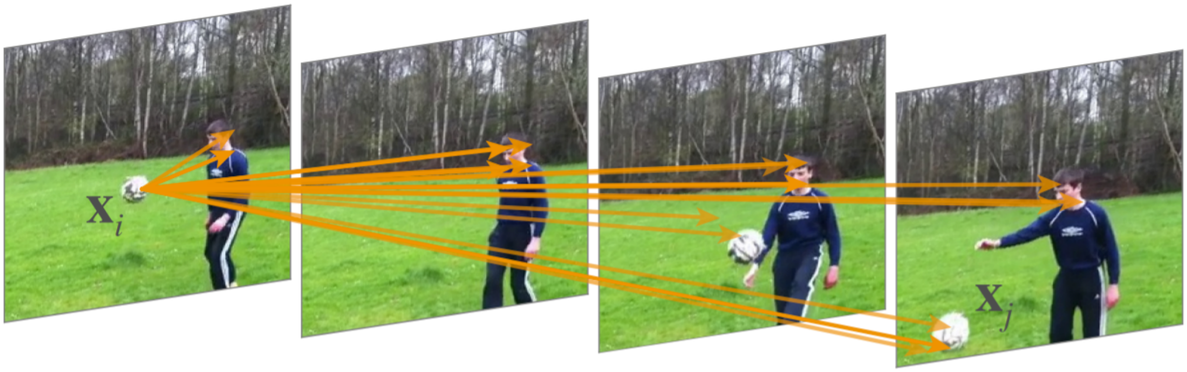


Figure 3.5: The Non-local Block [17] uses self-attention to model the interactions between all pixels in a video. Figure taken from [17].

uses self-attention [173] to model the interactions between all input features. The self-attention of video x at input position, x_i, is a weighted average over a representation learnt at all positions in the video, x_j. The similarity score between x_i and x_j determines which regions are given a higher weighting in the output. Multiple instances of the Non-local block are inserted between 3D convolutional layers, increasing the performance of I3D on Charades [8] and Kinetics [26]. Recent architectures [29] replaced all convolutional layers with self-attention, showing competitive results with state-of-the-art on Kinetics [26], EPIC-KITCHENS [10] and Something-Something [7] datasets.

3.2.3 Two-stream Architectures

An alternative approach to learning spatio-temporal features is to use a two-stream approach [18, 19, 20, 27, 174, 175, 176]. One network is trained on RGB images to learn appearence information, and another is trained on optical flow to learn motion information. Each stream is trained separately and the class scores are combined during testing. This overcomes the high computational cost of

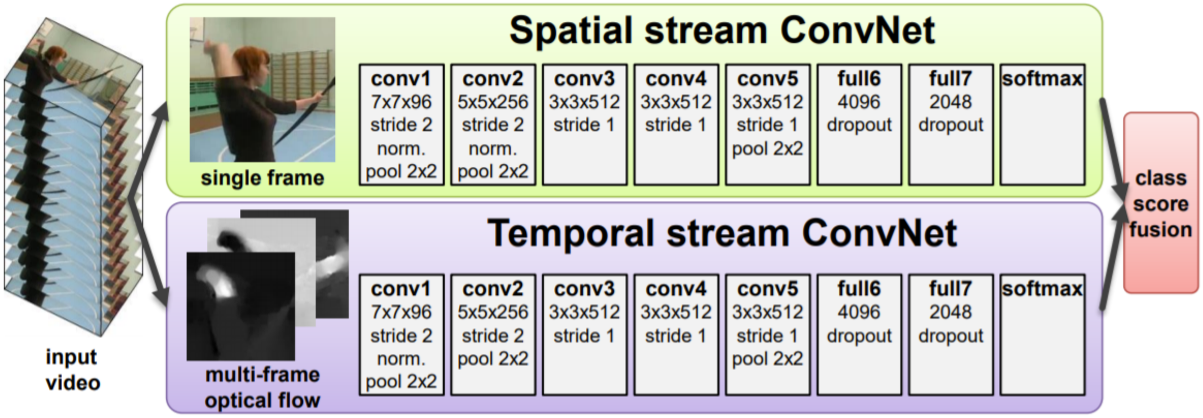


Figure 3.6: The two-stream network fuses predictions from RGB and optical flow streams. Figure taken from [18].

3D convolutions and the inability of early convolutional architectures to fully exploit the temporal dimension [16]. Modern 3D convolutional architectures, such as I3D [26], benefit from using a two-stream approach, fusing scores from a separate 3D CNN learned on RGB and optical flow.

The first two-stream approach was proposed by Simonyan *et al*. [18] (Fig. 3.6). The horizontal and vertical displacement vectors from optical flow produced by 10 consecutive frames are stacked and used as input. This provided a 7% improvement over using a single optical flow frame. The authors also experiment with stacking dense trajectories [158] but this provided worse action recognition accuracy on UCF-101 [25]. One benefit of using 2D CNNs is that ImageNet pretraining can be used for the spatial stream, however, the optical flow stream needs to be trained on video data. To increase the amount of video training data, they train the optical flow stream on both HMDB and UCF-101, using a separate classification head for each dataset. This improved the classification performance on HMDB by 9%. Finally, they show that the optical flow stream outperformed the RGB stream on both UCF-101 and HMDB, but the fusion of both streams provided the best performance. Their method outperformed Improved Dense Trajectories (iDT) [159], and C3D [168] required additional iDT features in order to outperform the two-stream approach.

Wang *et al.* [19] showed that using ImageNet pretraining on the optical-flow stream can improve performance. The weights of the first layer were replaced by the average of the weights across the RGB channels. Additional regularisation strategies were used including dropout, batch-normalisation and augmentation strategies, such as corner cropping and scale jittering. The ImageNet pretraining of flow and additional regularisation strategies improved two-stream fusion by 2%.

Rather than combining classification scores from optical flow and RGB streams at test-time, Feichtenhofer *et al.* explored how to fuse the two streams during training [174]. Two fusion approaches improved performance on UCF-101: concatenating convolutional feature maps from each stream followed by a projection to reduce dimentionality, and simply summing the softmax predictions during training. They showed that only concatenating, summing, or taking the maximum activation across feature maps leads to worse results than combining predictions at test time. Late fusion of the last convolutional layer achieved best performance, and the earlier fusing takes place, the lower the action recognition accuracy. The final architecture performs a 3D convolution and pooling operation in the fusion layer to include additional temporal information during training.

The performance of the RGB stream can be improved by including motion information learnt in optical flow stream at different layers. Feichtenhofer *et al.* [175] exploited a two-stream ResNet architecture [177] for this purpose. The ResNet architecture is a stack of blocks, which uses convolutions to learn a residual, $f(x)$, between the input, x, and output, $x + f(x)$. For each layer, the residual of the RGB stream is multiplied with the residual of the optical flow stream. Using residual information was necessary as combining feature maps led to poor performance. For fusing information, multiplication outperformed summation, and only the RGB stream benefited from optical flow and not vice-versa. This method improved action recognition accuracy over the standard two-stream architecture [18] by 6% on UCF-101 and 9% on HMDB.

Two-stream approaches using different resolutions of input videos improve action recognition performance. Karpathy [16] trained 2D-CNNs on low and high resolution images, fusing the classification scores during training. The high resolution CNN takes in the center crop of the images, mirroring the fovea in the human eye, where a small central portion of visual perception is observed more clearly. Slow-Fast [27] uses a two-stream approach with different temporal resolutions. A slow-stream acts on video sequence sampled at a low-frame rate, to capture appearance information, and the fast-stream takes high-frame rate sequences to capture motion information. The fast-stream replaces the optical flow stream in previous works [18, 175], which contains smaller number of feature maps in the convolutional layers to reduce the number of paramters and computational complexity. Similar to Feichtenhofer *et al.* [175], lateral connections propagate residuals of the fast-stream to the slow-stream. Fusing stream information achieved a 2% improvement of Kinetics and the use of the fast-stream, instead of optical flow, improved performance by 1.7%.

Audio can be used as a third modality alongside RGB and optical flow. In the Temporal Binding Network [23], each modality is trained with a separate 2D-CNN backbone, and fused by concatenating convolutional features from each stream followed by a multi-layer perceptron to learn interactions between the streams. The audio of an action may not be temporally aligned with the motion, *e.g.* the motion of *'putting down a cup'* occurs before the sound of the cup hitting the table. Therefore modalities are sampled from a video with different temporal offsets. The addition of audio improved action recognition performance on EPIC-KITCHENS-55 [41] by 4% and 2% for the seen and unseen participants, respectively.

Our work in Chapter 4 adapts a multi-modal architecture for unsupervised domain adaptation, proposing a self-supervision task to exploit the differing robustness of optical flow and RGB modalities to environmental changes. This can be applied to any action recognition architecture that late fuses feature representations from multiple modalities.

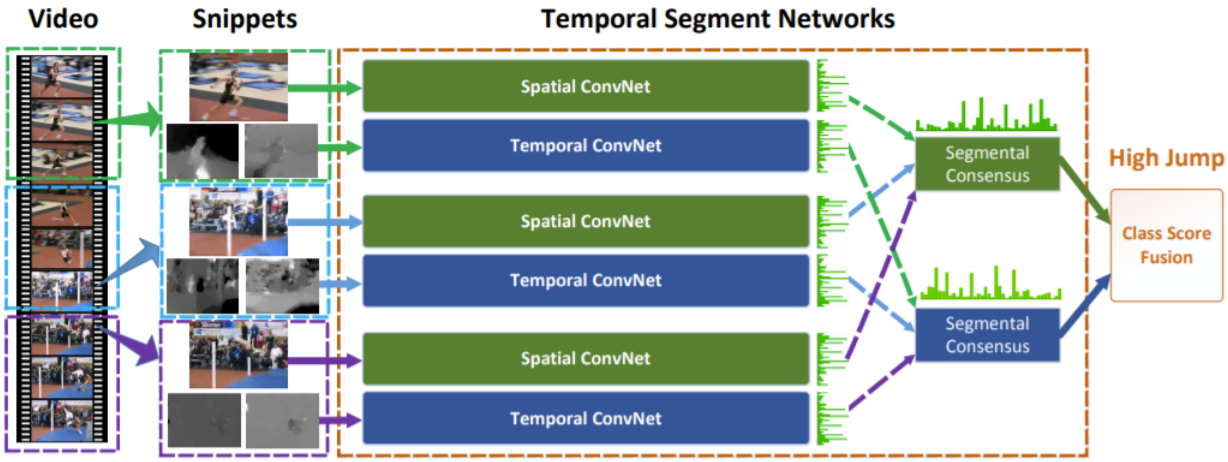


Figure 3.7: Temporal Segment Network classify actions using a consensus of frame-level predictions from the start, middle and end of a video. Figure taken from [19]

3.2.4 Temporal Consensus of Frames

To better model longer temporal structures, the consensus of multiple frames can be used during training. Temporal Segment Network (TSN) [19] (Fig. 3.7) splits the video into three segments of equal size, representing the beginning, middle and end of the video. A frame is sampled randomly from each segment, and passed through the CNN to produce classification scores for individual frames. The frame-level scores are averaged to produce a video-level prediction, which is used to train the network end-to-end with a softmax cross-entropy loss. At test time, logits from 25 frames, spaced equally throughout the video, are averaged. The method uses a two-stream approach, training optical flow and RGB streams separately, fusing video-level predictions at test time. The consensus strategy provided a 1.5% improvement over original two-stream network [18].

Several works use recurrent networks to model interactions between frames in long temporal structures [178, 179]. A 2D-CNN is connected to the input of an LSTM, and trained end-to-end with a sequence of frames. Yue *et al.* [179] show that max-pooling CNN features across multiple frames achieves competitive performance to that of an LSTM. TSN, which has no information on

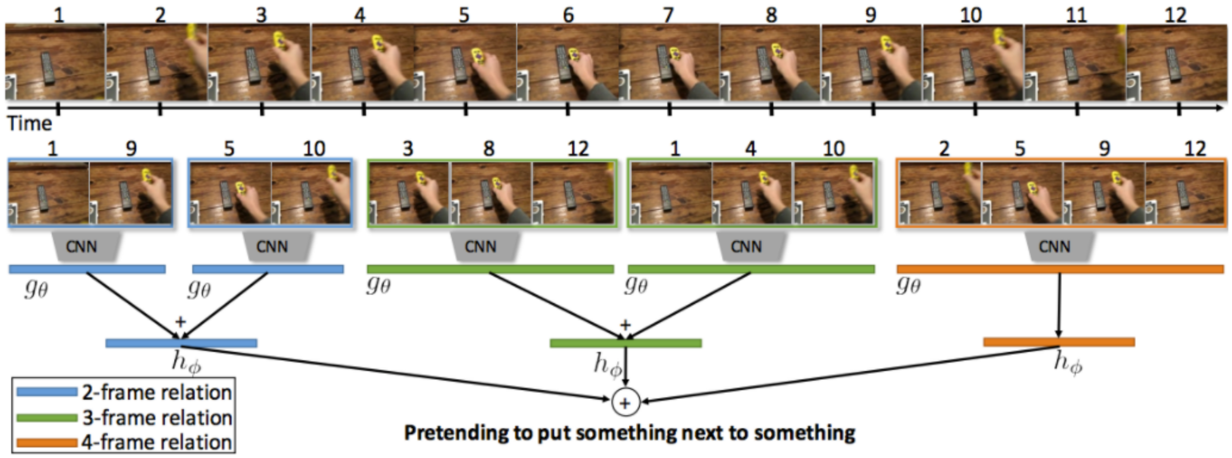


Figure 3.8: The Temporal Relation Network [20] models the interaction between different combinations of frames. Figure taken from [20].

the ordering of frames, outperformed both approaches.

Temporal Relation Network (TRN) [20] (Fig. 3.8) extends TSN [19] with a learnable pooling strategy that considers the temporal ordering of frames. CNN features from a subset of m frames are sorted in temporal order and fed into an MLP to produce a multi-frame representation. This considers the interaction between pairs of frames ($m = 2$), up-to the interaction between all N sampled frames ($m = N$). An m-frame relation, T_m, aggregates the representations from all possible combinations of m frames, and the final classification score is the sum of all frame relations, $\sum_{m=2}^{N} T_m$. TRN showed comparable performance to TSN on activity recognition datasets, UCF-101 [25] and Kinetics [26], and a substantial improvement on fine-grained action recognition datasets, Something-Something [7] and Charades [26]. This shows that temporal reasoning is more important for recognising fine-grained actions than activities which have a larger bias toward visual appearance.

The Temporal Shift Module (TSM) [176] uses the sampling strategy of TSN [19] to approximate 3D convolution with the same computational cost as 2D convolution. Before each convolution operation, part of the channel dimension is shifted forward and backwards in time. Therefore, feature activations for a given frame, X_t, at time t have some channel information replaced with activation at times $t-1$ and $t+1$. This mirrors a convolution of size 3 across time, where the output at time t, $Y_t = w_1X_{t-1} + w_2X_t + w_3X_{t+1}$, is a weighted summation of the inputs over different time-steps, where w_1, w_2, w_3 are weights. To reduce data movement between time-steps, only one eighth of the channels are shifted, increasing latency compared to the standard 2D-CNN by only 3%. The authors noticed improved performance shifting the channels inside the residual of a ResNet based architecture, this improved learning of spatial features as the original channel information is preserved in the skip connections. TSM improved upon TSN on Kinetics [26] by 3.5%, and a larger 31.3% improvement on the fine-grained action recognition dataset, Something-Something [7]. TSM required half the number of floating point operations compare to I3D [26], and outperformed both TRN [20] and I3D on Something-Something.

3.2.5 Self-supervision

Self-supervision in video is used as a pretext task to learn visual representations of videos without label information. This learnt representation should be easily adaptable for fine-tuning on downstream action recognition tasks, as the representation learnt through self-supervision disentangles many explanatory factors of the videos. Self-supervised approaches for video exploit the temporal and multi-modal properties of videos to generate ‘free’ labels to train representations.

Temporal Self-supervision

Shuffling the order of frames has been used to generate self-supervised tasks [180, 181, 182, 183]. Misra *et al.* [180] sampled three frames from a video and trained a network to predict when they are in the correct temporal order. This pre-text task improves performance on UCF-101, when fine-tuning with labelled data, over random initialisation. Odd-One-Out Network [181] trains a network to detect which of three video sequences contains shuffled frames. To encode temporal information of each input video, they use the sum of the differences of RGB frames as input. This improves down-stream action performance on UCF-101 compared to Misra *et al.* [180], due to the more difficult classification task and the encoding of motion information.

Other works predict the order of shuffled frames [182, 183]. Ahsan *et al.* [183] splited video frames across multiple time-steps into patches and trained a network to predict the order which recreates the video. This requires the network to learn both spatial and temporal information. All patches are passed through a 2D-CNN and fed into an MLP to solve the self-supervised task. All of the above approaches use the AlexNet [69] as their 2D-CNN backbone, but fine-tuning performance on UCF-101 does not outperform ImageNet pretraining.

Xu *et al.* [182] trained a 3D-CNN back-bone to predict the order of shuffled sequences of frames. Fine-tuning a 3D-CNN on UCF-101 from scratch already outperforms the previous approaches [180, 181, 183]. Predicting the order of frames as a pre-text task increases performance over random initialisation by 16% on UCF-101.

Reversing videos in time has been used to produce self-supervised tasks [184]. While some time-reversed videos produce videos of valid actions (*swipe lift* reversed is *swipe right*), many actions in forward-time are irreversible and their reversed videos defy the laws of physics (*e.g. crack egg*) [185] . Wei *et al.* [184] exploited this asymmetry of time by training a network to distinguish forward-time videos from reversed-time videos. Their network predicts if a video is reversed from a stack of optical flow frames as a pre-text task and is fine-tuned on action recognition datasets [25]. Artifacts from video encoding were removed prior to training, so the network doesn't exploit the cues to determine if a video is reversed. Their method improves action recognition accuracy on UCF-101 [25] by 2% over ImageNet pretraining.

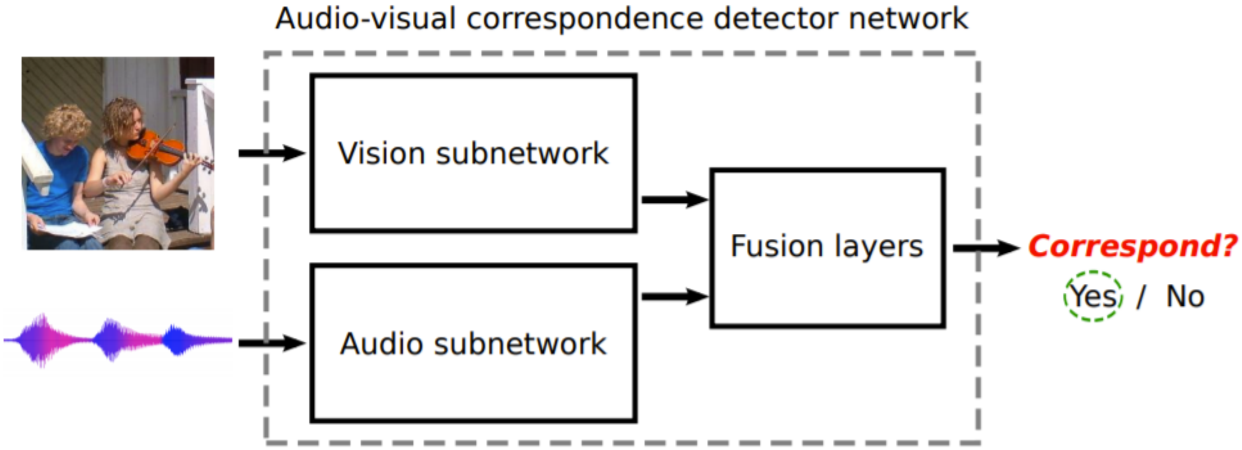


Figure 3.9: The temporal alignment between audio and visual inputs is used to create self-supervised representations. Figure taken from [21].

Multi-modal Self-supervision

Videos contain a variety of different modalities synchronised in time: gyroscopes and accelerator sensors attached to robots [186], optical flow displacement vectors extracted from videos [187, 188], and audio [21, 189, 190]. Self-supervised objectives exploit the temporal correspondence between modalities to learn video representations beneficial for action recognition.

The majority of multi-modal self-supervised approaches utilise the temporal correspondence between audio and RGB [21, 189, 190]. The seminal work [21] (Fig. 3.9), proposed a binary classification task which predicts if a paired image and audio input is sampled from the same, or a different video. The visual and audio modalities are processed by separate CNNs and the features from each modality are concatenated and fed into an MLP to predict temporal correspondence. They show that the visual CNN produces spatial activations at pixel location of objects that produce sound, despite complex backgrounds and scene clutter.

Several works use misaligned audio and visual modalities in a single video for a self-supervised objective [189, 190]. Owens *et al.* [189] shifted the audio in half of the videos from AudioSet [191] by 2-5.8 seconds, and a 3D-CNN was trained to predict if the visual and audio modalities are synchronised. A separate stream of five convolutional layers processes the audio waveform before fusing into the video architecture after the second convolution. Their model is fine-tuned on UCF-101, improving action recognition performance over I3D [26] trained on RGB from scratch by 14%, and randomly sampling of audio from different videos [21] by 3%. In addition, their method outperforms self-supervised tasks that consider the temporal order of frames [180, 181, 182, 183].

Korbar *et al.* [190] used a curriculum learning strategy to learn the temporal correspondence of modalities. For the first 50 epochs, a CNN is trained to align *easy negatives*, where audio is taken from different videos to that of the visual modality. Later in training, 25% of mis-aligned videos were *hard negatives*, where audio is sampled from the same video with a temporal offset greater than 0.5 seconds. Hard-negatives improved the performance of their network fine-tuned on UCF-101 by 9%. However, the performance decreased if hard-negatives were included from the start of training. They use contrastive learning to embed audio and visual information processed by a separate CNNs. When modalities are synchronised, the distance between the output of each CNN is minimised, otherwise the distance is maximised. The binary classifier used in previous work [21, 189] would not converge in the authors setup. Their method improves upon Owens *et al.* [189] by 5% on UCF-101.

Optical flow has been used as a self-supervised task for pretraining video models [187, 188]. Wang *et al.* [187] trained a 3D CNN on a sequence of frames to predict a series of labels describing motion and appearance. The derivative of optical flow was used to generate motion labels, and the change in color at spatial positions across the frames was used to generate appearance labels. Pretraining to predict motion information produced greater action recognition performance than predicting appearance labels when fine-tuning on UCF-101. Predicting the optical flow field has been applied as self-supervised task for first person action recognition [188]. A 2D CNN was first trained to regress to the optical flow displacement vectors, before fine-tuning for action recognition on the same dataset. The self-supervised task showed a 3.5% improvement on action recognition on EGTEA+ [14] over training from scratch.

Inspired by the audio-visual correspondence objective [21, 189], Chapter 4 proposes a temporal correspondence between multiple modalities (optical flow, RGB and audio) as a self-supervised task for unsupervised domain adaptation. Ours is the first work to show the benefit of self-supervision for domain adaptation in video.

3.3 Domain Adaptation for Action Recognition

Of the several domain shifts of using videos for action understanding, cross-viewpoint (or viewpoint-invariant) action recognition has seen the most research attention [9, 192, 193, 194, 195, 196, 197]. These works focus on adapting to the geometric transformations of a camera but do little to combat other shifts, like changes in environment. Works utilise supervisory signals such as skeleton or pose [194, 196] and corresponding frames from multiple viewpoints [9, 192, 193, 197]. For domain shifts other than viewpoint (*e.g.* changes in environment), corresponding frames are not available in different domains.

Before 2018, few works had conducted UDA for action recognition without the use of skeleton data or corresponding images across the domains. All used shallow models to align source and target distributions of handcrafted features [36, 198, 199, 200]. It has only been recently that UDA has seen significant research attention [1, 2, 37, 38, 42, 43, 44, 129]. Research has looked into how to align the temporal dynamics of video [1, 2] and how to avoid uninformative background frames [37, 43]. Other works have used self-supervised objectives for domain adaptation [43, 44] and considered the multiple modalities of video for domain alignment [42, 44, 129]

Domain Adaptation approaches used either 3D convolutional architectures [2], or the interaction between frames [1], to model the temporal dynamics. Jamal *et al.* [2] compared deep domain adaptation to shallow-based UDA methods to align features from C3D [168]. For shallow based approaches, they use Subspace Alignment (SA) [56] and Geodesic Flow Kernel (GFK) [3], on a stack of features extracted from C3D. Their deep domain alignment approach (DAAA), uses adversarial training with gradient reversal layers [4] to align the activations of the final convolutional layer of C3D. They tested their methods on cross-dataset adaptation on 5 coarse-grained action recognition datasets: UCF50 [148], Olympics [201], KTH [140], MSR [202] and SonyCam [2]. Adversarial training outperforms SA and GFK on all benchmarks. The majority of UDA methods for action recognition have since used adversarial training to align domains [1, 37, 43, 129].

Temporal Attentive Alignment (TA3N) [1] aligns temporal dynamics by placing multiple domain discriminators within the Temporal Relation Network (TRN) [20]. A domain discriminator acts on each frame-relation, learning to distinguish between domains at different temporal scales Additional domain discriminators act on the pooled video-level features and the input spatial features. Video-level features are a weighted summation of frame-relations. Higher weighting is given to frame-relations with low-entropy domain predictions, thus focusing video-level alignment on frame-relations that have not been aligned. They show that both the alignment of the frame-relations and the domain attention mechanism improve domain adaptation results on 5 benchmarks of course grained action recognition. This includes three proposed in their work, using the datasets: UCF-101 [25], HMDB [146], Kinetics [26] and GamePlay [1].

Several works used an attention mechanism to focus on informative frames for action recognition [37, 43]. Choi *et al.* [43] learnt a weighted summation of individual frame predictions, thus giving a higher weight to informative frames for action recognition and less to background frames. Temporal Co-attention Network (TCoN) [37] proposes to attend to informative frames which have features in common to both source and target domains. To select informative frames, they use self-attention to give higher weightings to frames with high similarity to other frames in the same video. To select common features between domains, frame representations with high similarity to all frames in the other domain are selected. The resulting co-attention is the product of both the cross-domain similarity and self-attention scores. TCoN improves on TA3N [1] on various domain adaptation benchmarks for coarse-grained action recognition, and proposes a new benchmark for gesture recognition using the Jester dataset [203].

Self-supervised approaches have been effective to align source and target domains [43]. These learn a self-supervised task jointly on both source and target domains, while jointly optimising the source classification task. Choi *et al.* [43] randomly shuffled the order of video clips in video sequences, and predicted the correct temporal order of frames. This method improves on TA3N [1] and TCoN [37] on the domain adaptation benchmark using coarse grained action recognition datasets, UCF-101 [25] and HMDB [146]. Chapter 4 is the first work to explore self-supervision for domain adaptation in video, which uses the temporal correspondence of multiple modalities.

Multiple modalities (RGB, Audio and Flow) have been used for domain adaptation [42, 44, 129]. Early approaches use RGB only [1, 2], or reported results on both RGB and Flow [37], however, the modalities are aligned independently and only fused during inference. The first approach using multiple modalities for domain adaptation is presented in Chapter 4. This learns the temporal correspondence between modalities as a self-supervised objective, jointly trained on both domains.

Spacio-temporal Contrastive Domain Adaptation (STCDA) [42] uses a contrastive loss to ensure that the distance between optical flow and RGB representations is minimised in feature space. They apply the loss to representations of both modalities on individual clips (I3D or BNInception features from 16 frames), and the pooled video representations aggregating 3 clips using a GRU. In addition, they align the class means of the video representations, in source and target domains, in Reproducing Kernel Hilbert Space. This is similar to the Contrastive Domain Discrepancy metric (CDD) [130] in Sec. 2.3.6. Similarly to CDD, they obtain class labels for target videos by clustering target videos with K-means, using the cluster assignment as pseudo-labels. Both target pseudo-labelling and alignment are applied separately for each modality. Their method improves on ours by 0.9% on our EPIC-KITCHENS-55 domain adaptation benchmark. Zhang *et al.* [129] used a pseudo-labelling approach to train an action classification task with optical flow and RGB target data. Pseudo-labels are generated for target data from the classification scores. Target data is only selected for training if the classification scores from both RGB and flow exceed a threshold and predict the same class. Their domain adaptation strategy is evaluated on coarse grained action recognition dataset, UCF-101 [25] HMDB [146] and Olympics [201], outperforming TA3N [1] by a large margin.

RNA-Net [44] matches the feature norm of the audio and RGB to generalise to new domains. This is similar to Xu *et al.* [113], which aims to match the feature norm between domains. The RGB stream is an I3D network and the Audio stream is BNInception. They evaluate on our EPIC-KITCHENS-55 domain adaptation benchmark (Chapter 4), for both domain generalisation and adaptation. Matching feature norms improves single source domain generalisation by 4%, where there is no access to target data during training. They extend their approach to domain adaptation, matching the feature norms of both source and unlabelled target data. Their method provides a 5.8% improvement over training only on the source domain classification task. They compare to our self-supervised loss in Chapter 4, tested on RGB and Audio, and outperform it by 1.3%. Our method outperforms theirs when the optical flow modality is included.

Our work in Chapters 4 and 5 provide the only domain adaptation benchmarks for first-person action recognition. Chapter 4 considers the domain gap due to changes in environment, and Chapter 5 considers the domain gap that occurs during long-term collection of footage. Table 3.1 compares the size of different domain adaptation benchmarks for action recognition datasets. With the exception of Kinetics-Gameplay, our benchmarks provide the largest domain adaptation benchmarks for action recognition.

3.4 Cross-modal Retrieval Models

In retrieval problems, a query (*e.g.* a single text, image or video instance) is used to retrieve instances from a gallery (*e.g.* a database of texts, images or videos) which are semantically relevant to the query. Cross-modal retrieval is the case were the queries and the gallery set contain different modalities. For example, Chapter 6 focuses on text-to-video retrieval, which uses textual queries

Paper	Domain Pairs			No. Instances		No. Classes
	Domain 1		Domain 2	Domain 1	Domain 2	
Sultani *et al*. [36]	UCF50	⟷	HMDB	671	500	5
	UCF50	⟷	Olympic	600	315	6
Jamal *et al*. [2]	MSR	⟶	SonyCam	202	90	3
	KTH	⟷	MSR	765	202	3
	KTH	⟶	SonyCam	1530	180	6
Chen *et al*. [1]	UCF101	⟷	HMDB	2009	1200	12
	Kinetics	⟶	Gameplay	46624	3374	30
Pan *et al*. [37]	Jester (S)	⟶	Jester (T)	N/A	N/A	N/A
Choi *et al*. [38]	Kinetics	⟶	NECDrone	2942	1500	7
MM-SADA (Chapter 4)	D2	⟷	D1	3245	1978	8
	D3	⟷	D1	4871	1978	8
	D3	⟷	D2	4871	3245	8
EPIC-KITCHENS-100 (Chapter 5)	2018	⟶	2020	16115	26115	3369

Table 3.1: Action recognition domain adaptation datasets. Key: ⟶ Domain Adaptation is evaluated in one direction, ⟷ Domains are used as both source and target.

on a gallery of videos. To achieve this, the similarity between different texts and videos is learnt on a large corpus of captioned videos.

Learning a joint embedding space is a common approach to perform text-to-video retrieval [22, 152, 204, 205, 206, 207, 208, 209, 210, 211]. This projects both video and text into a shared space to obtain compatible representations. Videos related to a given text query can be retrieved efficiently via a similarity metric in the embedding space, and descriptors from the gallery set can be pre-computed before any textual queries. Some recent approaches predict the semantically between text and video using a transformer to model complex cross-modal interactions [212, 213], however these are computationally expensive and inefficient when scaling to large gallery sets.

3.4.1 Learning an Embedding

To learn an embedding space, ranking losses are used to enforce alignment of text and videos. These ensure that all videos that are semantically related to a textual query (*positives*) are closer in space than those that are not relevant (*negatives*). The triplet loss, Eq. 3.1, is most commonly used ranking loss in the literature [204, 205, 206, 207, 209], which acts on an anchor, a (a single text or video instance), and sampled positive and negative instances from the other modality, b^+ and b^-. The aim is to encourage the distance, d, between positive pairs (a and b^+), to be greater than the distance between negatives (a and b^-), by at least a margin, γ.

$$H(a, b^+, b^-) = max(d\big(h(a), h(b^+)\big) - d\big(h(a), h(b^-)\big) + \gamma), 0) \tag{3.1}$$

Most works [204, 205, 206, 207, 209] minimise a bi-directional triplet loss which uses both text and video as anchors: $H(t, v^+, v^-) + H(v, t^+, t^-)$. This considers the distance of positive and negative videos, v^+ and v^-, to a text anchor, t, as well as positive and negative texts, t^+ and t^- to an video anchor, v.

Rather than using all possible triplets in a mini-batch, training with a single negative that is closest to the query can improve retrieval performance [214]. This is referred to as *hard negative mining* and focuses the optimisation on triplets with the most incorrect semantic structure in the embedding space. Training with hard negatives improves text-to-video performance when training and evaluating on the same [209], or different datasets [204].

In order to obtain positive and negative sets, the relevancy between texts and videos in datasets needs to be determined. Many datasets use human annotators to create textual descriptions for videos [150, 151], or generate automatic captions from audio transcriptions of the video [152]. The captioned text is used as a single positive instance, and all other textual descriptions in the dataset are considered negatives. Wray *et al.* [215] showed that defining a single text as relevant is problematic as there will exist other text that would describe the video equally well or better. Instead, they propose a set of non-binary relevancy metrics considering the semantic distance between the captioned text and other texts in the dataset. One such metric (Eq. 3.2) uses the intersection over the union of words, w, for different parts-of-speech (*i.e.* verbs, nouns and adverbs), P, to measure the relevancy between texts, y_i and y_j. A weight, α^p is used to trade-off different parts of speech in the relevancy metric:

$$S_{PoS}(y_i, y_j) = \sum_{p \in P} \alpha^P \frac{|w_i^p \cap w_j^p|}{|w_i^p \cup w_j^p|} \tag{3.2}$$

Large-scale pretraining from weakly captioned videos from online source has been shown to have good zero-shot and performance on downstream datasets [152], however many videos are labelled with text that is weakly related, incorrect or mis-aligned in time. This is a particular issue for HowTo100M [152], which automatically generates text captions from speech in instructional videos. MIL-NCE *et al.* [208] utilises a loss with Noise Contrastive Estimation (NCE) [216] and multiple instance learning to handle the noisy data. The key idea is to use a bag of positive texts instead of a single positive text for every video. Texts in the video closest in time to the ground-truth (temporally aligned) text are chosen as additional positive examples.

3.4.2 Representing Text

A common approach to represent words in text is to learn a word embedding, which maps words onto a continuous space such that semantically related words are closer than those that are not related. Word2Vec is an efficient strategy to learn word embeddings [217], which trains a feed-forward neural network to solve a word association task on a large text corpus. Many text-to-video retrieval methods [22, 206, 207, 208, 209, 218] use the skip-gram model [217], where the task is to predict the surrounding words based on current word. Directions in the embedding space produced by Word2Vec describe semantic properties of words. Therefore, learning an embedding between Word2Vec and visual features allows retrieval of videos from unseen words not seen in the training text corpus (zero-shot retrieval). Action2Vec [206] exploits Word2Vec to perform zero-shot classification of verbs. They learn an embedding between features extracted from C3D [168] and verbs encoded by Word2Vec.

To represent textual captions, word embeddings are pooled to form a text descriptors. Several works have used Recurrent Neural Networks (RNN) [210, 218], or transformers (BERT [219]) [205] to capture the interactions between word embddings to form textual descriptors. Burns *et al.* [220] compared different pooling operations of text embeddings for cross-modal image retrieval tasks. They showed that averaging Word2Vec embeddings performs comparably to modelling sentences with LSTMs or using a pretrained BERT embedding. Miech *et al.* [208] showed max-pooling word embeddings to represent text outperformed more complex sentence representations using RNNs or transformers [219] for cross-modal retrieval tasks on YouCook2 [150] and MSR-VTT [151].

Various works create representations from a subset of words that provide meaningful concepts to describe actions [22, 204, 218]. A seminal work for cross-modal video retrieval parses subject-verb-object triplets from each text [218], with each word represented by a Word2Vec embedding. A Recursive Neural Network combines and projects the Word2Vec embedddings into the visual-text embedding space.

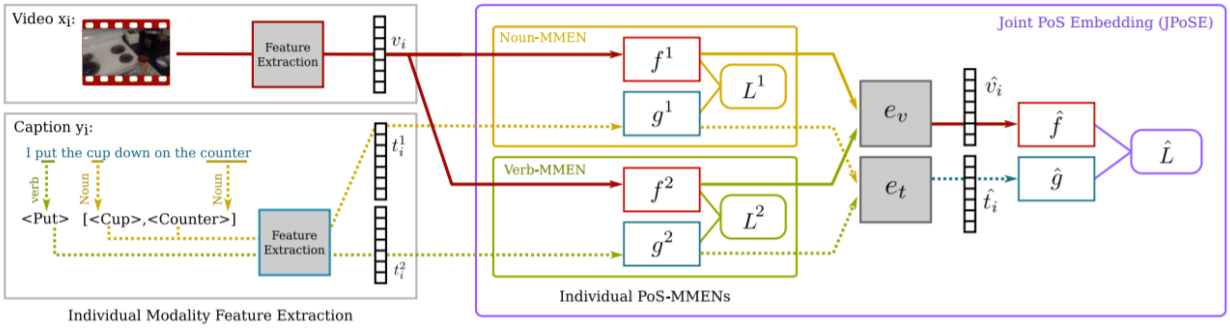


Figure 3.10: Joint Part of Speech Embedding (JPOSE) [22] learns a separate embedding for verbs and nouns. A final embedding is learnt from the concatenation of each part-of-speech embedding. Figure taken from [22]

Later works learnt text representations for different parts-of-speech in the text (i.e. verbs and nouns) [22, 204]. Representations for part-of-speech have been modeled by separate functions [22] or as part of a hierarchical semantic graph [204], with nodes representing verbs, nouns and captions at different layers of the graph. Unlike Xu *et al*. [218], these works learn a visual embedding to align the video with different syntactic properties.

Joint Part of Speech Embedding (JPOSE) [22] (Fig. 3.10) average the Word2Vec embeddings of individual parts-of-speech (verbs and nouns) as the input text representation. A visual and text embedding is learnt for each part-of-speech, with triplet losses to ensure that videos and text depicting the same verb, or noun, are close in space. The embedding spaces are concatenated and fed into a final embedding which ensures modalities depicting the same action (verb-noun combinations) are close. JPOSE is evaluated on EPIC-KITCHENS [41] and MSR-VTT [151], showing their method improves cross-modal retrieval performance over a single embedding trained on actions. Chapter 6 proposes a domain adaptation strategy to improve text-video retrieval performance of JPOSE [22] on target videos from a different domain to videos used during training. This difficulty of this task is that the text modality in the target domain is not available during training.

Chen *et al*. [204] modeled the text as a hierarchical semantic graph with different levels for nodes describing different syntactic properties: verbs, nouns phrases or the entire sentence. Edges represent the semantic role between nodes at different levels. A Relational Graph Convolutional Network (R-GCN) [221] projects the textual representations for each node into a text-visual embedding space. This models the interactions between adjacent nodes, with some weights specific to the semantic role. Similar to Wray *et al.* [22], a separate projection maps the visual features into embedding space of nouns, verbs and captions, which is enforced using a triplet loss. The average similarity score obtained from each embedding space is used rank textual queries to videos. They showed improved text-video performance on MSR-VTT, compared to learning a single embedding space for a captions.

3.4.3 Learning from Multiple Visual Experts

Many text-video retrieval works learn a separate embedding for different visual 'experts' [205, 207, 209, 210]. Experts can be representations of modalities (*e.g.* RGB, optical flow and audio) or feature extractors recognising different visual cues (*e.g.* networks pretrained to detect faces, objects or actions). Mithun *et al.* [210] learnt a separate embedding for embedding object features and action features with text. The final similarity score is the sum of the scores obtained from each embedding. Object features were extracted from a model trained on ImageNet. Action features are a concatenation of spatio-temporal features from I3D, pretrained on Kinetics, and audio features. Learning a separate embedding for object and action features improved text-video and video-text retrieval performance on MSR-VTT and MSVD datasets, compared to learning a joint embedding across all experts, or a single embedding between text and a concatenation of all visual features.

Increasing the number of visual experts improves cross-modal retrieval performance [207], but not all visual experts/modalities are available in all videos. The Mixture of Experts (MEE) [209] learns a similarity score between text and a varying number of visual experts (appearance, motion, audio and face descriptors). This learns a weight for each expert, based on the content of the caption, in addition to an embedding between text and each expert present in a training video. To allow videos to contain a different number of modalities, the softmax is taken over the weights of each experts present in a video. The final similarity score is then a weighted sum of the similarity scores using the expert weights. They show that learning from captioned images from COCO [222], which do not contain motion of audio information, improves text-video retrieval performance of MSR-VTT[151]. Further improvements were achieved using face descriptors from training videos that contain face detections. Instead of learning an embedding per expert, Collaborative Experts [207] learns an embedding for all pair-wise interactions of visual experts. Their work is evaluated with eight visual experts, including OCR, scene and object features. They show that both the learning of pair-wise interactions and the additional experts improve text-video retrieval performance on MSR-VTT. Gabeur *et al.* [205] extended the approach, using a transformer to learn the interactions between all available visual experts.

3.5 Domain Adaptation for Retrieval

It is common to evaluate text-video retrieval on a gallery set of the same domain to that used during training. Recent work has shown that learning an embedding on large-scale datasets of collecting weakly paired videos and text provides good performance on cross-modal retrieval when evaluated on a different target dataset [152], however, fine-tuning on the target dataset improves performance considerably. Domain adaptation aims to improve target cross-modal retrieval performance by leveraging the gallery set of target videos during training, without the paired textual queries. This reduces that annotation effort of pairing textual captions with each target video.

Most works related to domain adaptation for retrieval were proposed in the context of person re-identification where the aim is to retrieve instances of an individual across different cameras. The gallery set used in training does not contain the same set of individuals used during testing, therefore this is an extreme setting of the open-set domain adaptation problem (Sec. 2.3.7) where no labels are shared between domains. To overcome the domain shift between cameras, generative models learnt with similarity preserving constraints learn to transfer the style between domains [96, 97, 98, 99], similar to the generative models discussed in Sec. 2.3.3, or exploit pseudo-labels obtained by joint source and target clustering [118, 223, 224]. Different from the pseudo-labelling methods discussed in Sec. 2.3.6, unsupervised clustering on the target domain is conducted as the label space in the source in distinct from the target. Closer to our retrieval work in Chapter 6, some methods use a curriculum learning paradigm to progressively adapt the retrieval space [225, 226].

In contrast to the above literature, where the retrieval task is performed within a single modality (namely images) only few domain adaptation works have considered cross-modal retrieval [227, 228, 229]. This is where the query (*e.g.* text) is a different modality to that of the gallery (*e.g.* images). Similarly to Person-ReID, this is an open-set domain adaptation problem as there will be many captions in the target domain that are not present in the source, and vice-versa. Initial work [227] learnt correlations between captions and images using a scene graph and Maximum Mean Discrepancy (MMD) for domain alignment. However this doesn't address the differing class distributions in each domain as MMD does not utilise class information.

Recently, two concurrent works to ours, conduct UDA for text-to-video retrieval [228, 229]. Similar to our approach in Chapter 6, Chen *et al.* [228] used a pseudo-labelling strategy for domain alignment. Their method assigns source captions to target videos with an iterative, mutual-exclusion selection mechanism. A contrastive margin loss is trained on these captioned target videos to learn a cross-modal embedding on target data. In addition they preserve the knowledge of a set of predefined concepts in the embedding space. Labels for these concepts are obtained by the classifier response of an ImageNet pretrained model. A classifier is trained to predict these concepts labels from the embedding space used for retrieval. This work doesn't address the fact that some target instances may be incorrectly labelled and should not be used for training. Our work in Chapter 6, addresses this problem with a robust confidence measure for sampling pseudo-labelled target videos. Liu *et al.* [229] avoided aligning target videos to source textual captions, and instead attempts to align concepts. They train two prototypical classifiers in the cross-modal embedding space, one for the source and one for the target, where labels are found by respectively clustering the source text and target video representations. To align domains, they maximise the mutual information between the prototype assignments of both modalities.

3.6 Conclusion

This chapter introduced the datasets and models used in video understanding. Section 3.1 showed that fine-grained action recognition datasets have a large bias towards few environments, highlighting the need for domain adaptation methods for fine-grained action understanding. The literature on action recognition models (Sec. 3.2 and 3.3) provides related work for Chapters 4 and 5. In particular, Sec 3.2.5 describes multi-modal self-supervision and is re-purposed for domain adaptation in Chapter 4. Sec 3.3 reviewed domain adaptation works tailored for action recognition, which some are evaluated for fine-grained action recognition in Chapters 4 and 5, while others took inspiration from our multi-modal domain adaptation strategy, proposed in Chapter 4. The literature on cross-modal retrieval (Sec 3.4 and 3.5) provides related work for Chapter 6. In particular, Sec 3.4.2 describes the Joint Part-of-Speech Embedding model [22], which we adapt for domain adaptation in Chapter 6.

Chapter 4

Adapting Action Recognition Models to New Environments

Supervised approaches rely on collecting a large number of labelled examples to train discriminative models. However, due to the difficulty in collecting and annotating fine-grained actions, many datasets collect long untrimmed sequences. Often collection occurs in a single [13, 15] or few [8, 30] environments, with a limited number of participants and tools. This lack of diverse examples can lead to poor generalisation in new locations.

Although significant attention has been given to deep Unsupervised Domain Adaptation (UDA) in other vision tasks [4, 71, 73, 74, 76, 83], few works have attempted UDA for video data [1, 2], prior to this work. Surprisingly, none have tested on videos of fine-grained actions. Additionally all these approaches only consider video as images (*i.e.* RGB modality), and do not exploit the **multi-modal** nature of video.

Videos are highly multi-modal providing multiple different views of the same data (*e.g.* appearance, audio and motion). Each modality will have differing level of robustness to domain changes, which has not been exploited for domain adaptation. This is in contrast to self-supervised

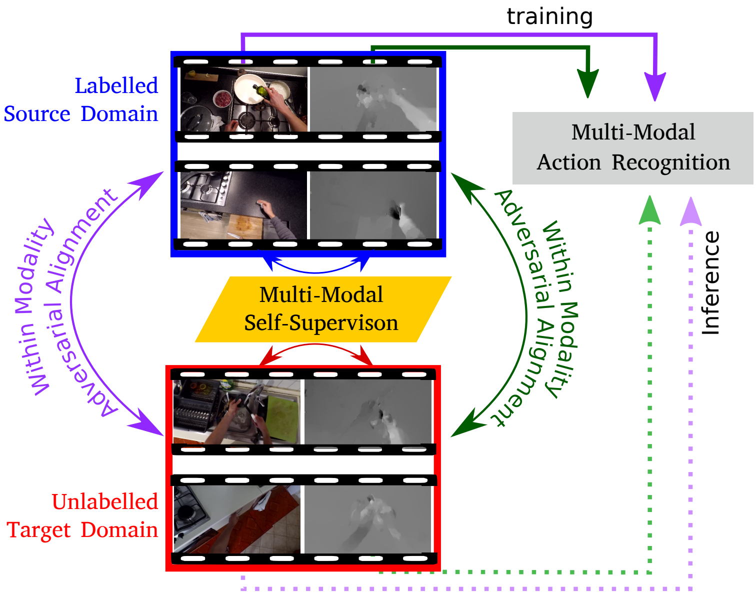


Figure 4.1: The proposed UDA approach for multi-modal action recognition. Improved target domain performance is achieved via multi-modal self-supervision on source and target domains simultaneously, jointly optimised with multiple domain discriminators, one per-modality.

approaches that have successfully utilised multiple modalities within video when labels are not available during training [21, 189, 190]. Additionally, self-supervised tasks have been shown to benefit domain adaptation for image classification and semantic segmentation [110, 114].

This chapter proposes the multi-modal UDA strategy, MM-SADA (Fig. 4.1), which combines domain adversarial alignment [4], with multi-modal self-supervision, to adapt fine-grained action recognition models to unlabelled target environments. The self-supervised objective exploits the differing levels of robust of each modality to environmental changes, to create a more robust representation for the domain shift. While this can generalise to a number of modalities, this work focuses on RGB, optical flow and audio.

A detailed description of the MM-SADA is provided in Sec. 4.1, which is tested on the first domain adaptation benchmark for fine-grained action recognition (Sec. 4.2). MM-SADA, instantiated with RGB and optical flow modalities (Sec. 4.3), outperforms source-only generalisation, previous UDA methods for action recognition [1, 2] and alternative domain adaptation strategies such as batch-based normalisation [101], distribution discrepancy minimisation [73] and classifier discrepancy [5]. In addition, the self-supervised task can be applied to audio, which improves both the recognition of actions and objects in the target environment (Sec. 4.4).

This chapter expands on our publication [40], which only considered RGB and Optical Flow, with additional experiments with Audio (Sec. 4.4). In addition the publication only considered closed-set domain adaptation of 8 classes, whereas this chapter extends to all verb and noun classes in an open-set setting. We also provide more experimental results of the performance of the self-supervised task and individual modality performance (Sec. 4.3).

4.1 Multi-modal Domain Adaptation

This section outlines the proposed action recognition domain adaptation approach: *Multi-Modal Self-Supervised Adversarial Domain Adaptation (MM-SADA)*. An overview of MM-SADA is provided in Fig. 4.2, visualised for action recognition using two modalities: RGB and Optical Flow. We incorporate a self-supervision alignment classifier, C, that determines whether modalities are sampled from the same or different actions to learn modality correspondence. This takes in the concatenated features from both modalities, without any labels. Learning the correspondence on source and target encourages features that generalise to both domains. Alignment of the domain statistics is achieved by adversarial training, with a domain discriminator per modality that predicts the domain. A Gradient Reversal layer (GRL) reverses and backpropagates the gradient to the features. Both alignment techniques are trained on source and unlabelled target data, whereas the action classifier is only trained with labelled source data.

The next section details MM-SADA, generalised to any two or more modalities. First the problem of *domain adaptation* is revisited and multi-stream late fusion is outlined, before describing the adaptation approach.

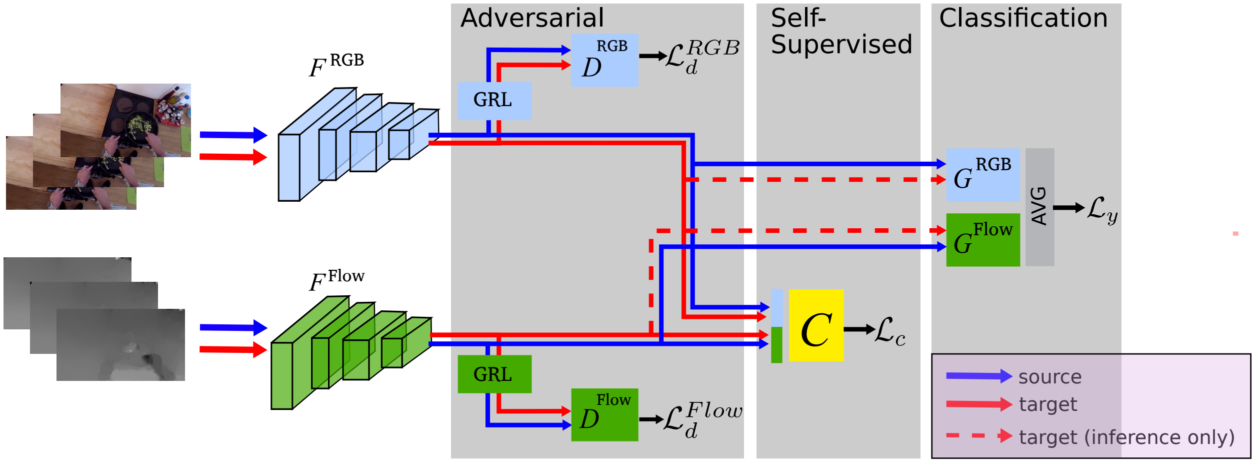


Figure 4.2: Proposed architecture: feature extractors F^{RGB} and F^{Flow} are shared for both target and source domains. Domain Discriminators, D^{RGB} and D^{Flow}, are applied to each modality. Self-supervised correspondence of modalities, C, is trained from both source and unlabelled target data. Classifiers, G^{RGB} and G^{Flow} are trained using source domain examples only from the average pooled classification scores of each modality. During inference, multimodal target data is classified.

4.1.1 Unsupervised Domain Adaptation (UDA) for Action Recognition

A domain is the distribution over the input population $\mathbf{X}$ and the corresponding labels $\mathbf{Y}$, which is described in more detail in Chapter 2. In this chapter, the source domain, $\mathbf{S} = \{X^s, Y^s \mathcal{D}^s\}$[1], is collected from a single recording environment, which is used to learn a representation, $G(\cdot)$, over some learnt features, $F(\cdot)$, that minimises the empirical risk, $E_{\mathbf{S}}[\mathcal{L}_y(G(F(x)), y)]$, given input videos with action labels $\{(x, y)\}$. The goal is to minimize the empirical risk on a target domain, $\mathbf{T} = \{X^t, Y^t, \mathcal{D}^t\}$, which is collected in a different recording environment, such that the distribution of videos in the source and target domains differ, $\mathcal{D}^s \neq \mathcal{D}^t$. In UDA, labels from the target domain are not available during training, thus methods minimise both the source risk and the distribution discrepancy between the source and target domains [46].

[1] A domain is defined in Chapter 2. The definition in this chapter excludes features space for clarity, as we homogeneous setting where feature spaces is are same for the source and target domains

4.1.2 Multi-modal Action Recognition

When the input is multi-modal, *i.e.* $X = (X^1, \cdots, X^M)$ where X^m is the m^{th} modality of the input, fusion of modalities can be employed. Most commonly, late fusion is implemented, where we sum prediction scores from modalities and backpropagate the error to all modalities, *i.e.*:

$$\mathcal{L}_y = \sum_{x \in \{\mathbf{S}\}} -y \log P(x) \qquad \text{where: } P(x) = \sigma\big(\sum_{m=1}^{M} G^m(F^m(x^m))\big) \tag{4.1}$$

where G^m is the modality's task classifier, and F^m is the modality's learnt feature extractor. The consensus of modality classifiers is trained by a cross entropy loss, $\mathcal{L}_y$, between the task label, y, and the prediction, $P(x)$. σ is defined as the softmax function. Training for classification expects the presence of labels and thus can only be applied to the labelled source input.

4.1.3 Multi-modal Adversarial Alignment

Both generative and discriminative adversarial approaches have been proposed for bridging the distribution discrepancy between source and target domains. Discriminative approaches are most appropriate with high-dimensional input data present in video. Generative adversarial approaches require a huge amount of training data and temporal dynamics are often difficult to reconstruct. Discriminative methods train a discriminator, $D(\cdot)$, to predict the domain of an input (*i.e.* source or target), from the learnt features, $F(\cdot)$. By maximising the discriminator loss, the network learns a feature representation that is invariant to both domains.

To extend adversarial training to a multi-modal architecture, we propose to use a domain discriminator per modality that penalises domain specific features from each modality's stream. Each separate domain discriminator, D^m, is thus used to train the modality's feature representation F^m. Given a binary domain label, d, indicating if an example $x \in \mathbf{S}$ or $x \in \mathbf{T}$, the domain discriminator, for modality m, is defined as,

$$\mathcal{L}_d^m = -\sum_{x \in \{\mathbf{S},\mathbf{T}\}} d \log(D^m(F^m(x))) + (1-d)\log(1 - D^m(F^m(x))) \tag{4.2}$$

This loss is maximised with respect to the feature extractor weights, $max_{\theta_{Fm}} L_d^m$ and minimized with respect to the domain discriminators weights, $min_{\theta_{Dm}} L_d^m$. θ_{F^m} and θ_{D^m} are the weights of F^m and D^m, respectively.

An alternative approach would be to train a single domain discriminator on a combined representation from both modalities. The combined representation could be the concatenation of all modalities, or the outer product between two modalities. However, initial experiments showed that learning a separate domain discriminator per modality was more effective for improving target action recognition performance. To benefit from multiple modalities, this chapter proposes a multi-modal self-supervised objective, which is outlined next.

4.1.4 Multi-modal Self-supervised Alignment

Prior approaches to domain adaptation have mostly focused on images and thus have not explored the multi-modal nature of the input data, where each of the modalities (RGB, optical flow and audio) are robust to different environmental changes. Optical flow is robust to appearance changes and background textures as it represents the apparent motion between frames. Some actions produce distinctive sounds that are robust to visual appearance, such as the crack in *cracking an egg* or the audible click when *flicking a switch* or *turning on a stove*. For actions with a large variation in sound and motion across participants, RGB will be more robust. For example, in the action *open a parcel*, the parcel could be cut open, ripped open or neatly unfolded. The visual representation of hands interacting with the object may be more robust than pure motion information.

Therefore, this chapter proposes a multi-modal self-supervised task to align domains by learn-

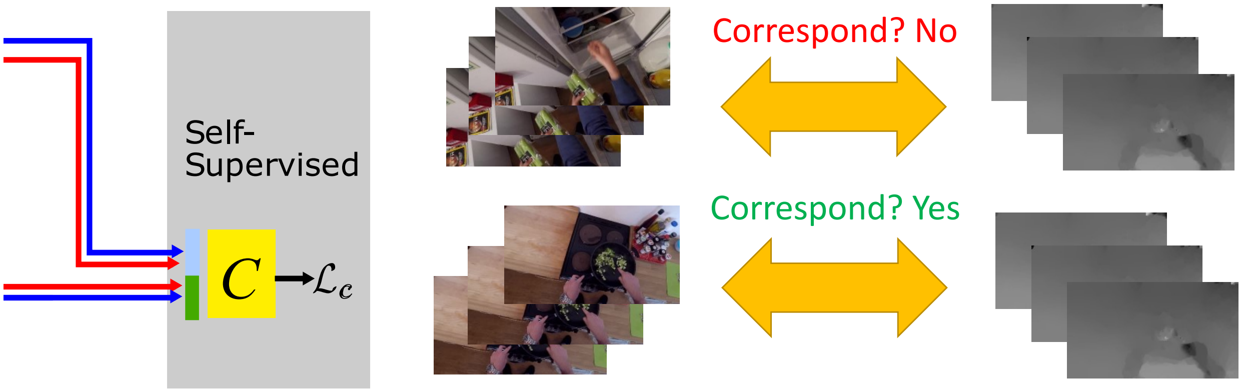


Figure 4.3: The self-supervised classifier predict the temporal correspondence of modalities, optimised over both source and target inputs.

ing from different modalities. This exploits that modalities are synchronised in time and present in both source and target domain. Multi-modal self-supervision has been successfully exploited as a pretraining strategy [21, 230] (Chapter 3, Sec 3.2.5). However, we show that self-supervision for both source and target domains can also align domains. Note that the self-supervised objective has no knowledge of which modality is more robust during training. This is advantageous as the robustness of each modality will vary depending on the action performed in the input video, and allows flexibility for different applications with different modalities.

The proposed approach learns the temporal correspondence between modalities as a self-supervised binary classification task (Fig. 4.3). For positive examples, indicating that modalities correspond, we sample modalities from the same action. These could be from the same time, or different times within the same action. For negative examples, each modality is sampled from a different action. The network is thus trained to determine if the modalities correspond. This is optimised over both domains. A self-supervised correspondence classifier head, C, is used to predict if modalities correspond. This shares the same modality feature extractors, F^m, as the action classifier. Given a binary label defining if modalities correspond, c, for each input, x, and concatenated features of the multiple modalities, we calculate the multi-modal self-supervision loss as follows:

$$\mathcal{L}_c = -\sum_{x\in\{\mathbf{S},\mathbf{T}\}} c\log C(F^0(x),...,F^M(x)) + (1-c)\log(1-C(F^0(x),...,F^M(x))) \tag{4.3}$$

4.1.5 Proposed MM-SADA

The Mutli-Modal Self-Supervised Adversarial Domain Adaptation (MM-SADA) approach is defined as follows. The classification loss, $\mathcal{L}_y$, is jointly optimised with the adversarial and self-supervised alignment losses. The within-modal adversarial alignment is weighted by λ_d, and the multi-modal self-supervised alignment is weighted by λ_c. Optimising both alignment strategies achieves benefits in matching source and target statistics and learning cross-modal relationships transferable to the target domain.

$$\mathcal{L} = \mathcal{L}_y + \lambda_d \sum_m \mathcal{L}_d^m + \lambda_c \mathcal{L}_c \tag{4.4}$$

Note that the first loss $\mathcal{L}_y$ is only optimised for labelled source data, while the alignment losses $\forall m : \mathcal{L}_d^m$ and $\mathcal{L}_c$ are optimised for both unlabelled source and target data.

4.2 EPIC-KITCHENS for Action Recogntion Domain Adaptation

The EPIC Kitchens dataset [30] offers a unique opportunity to test domain adaptation for fine-grained action recognition, as it is recorded in 32 environments[2]. Similar to previous works for action recognition [2, 4], MM-SADA is evaluated on pairs of domains. The three largest kitchens, in number of training action instances, are selected to form the domains. These are P01, P22, P08,

[2] This refers to the initial dataset collection, EPIC-KITCHENS-55 [30], and not the extension [10].

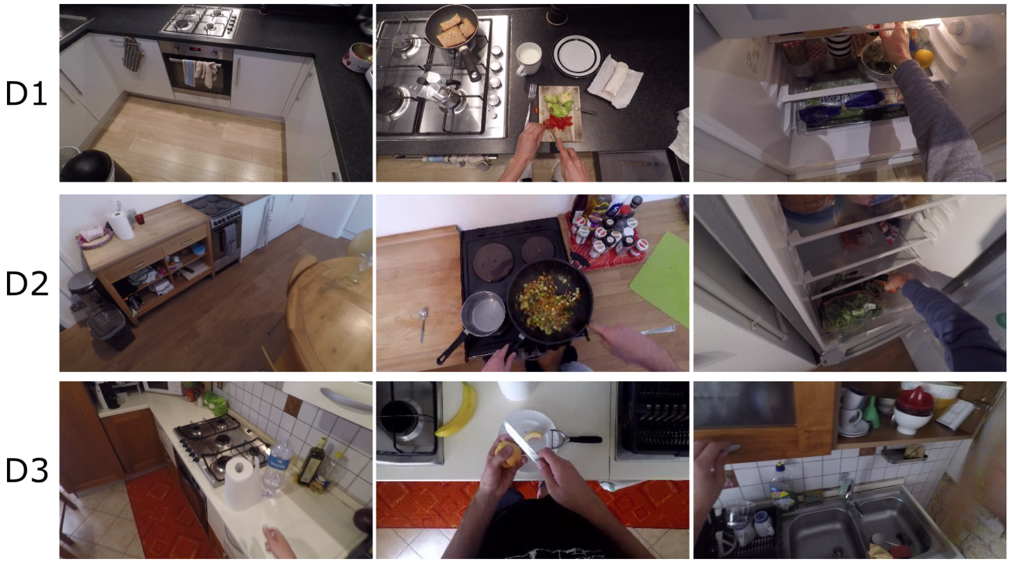


Figure 4.4: Three kitchens from EPIC-Kitchens selected as domains to evaluate our method.

which we refer to as D1, D2 and D3, respectively (Fig. 4.4).

This chapter proposes two benchmarks for unsupervised domain adaptation, closed set domain adaptation and open set domain adaptation.

4.2.1 Closed Set Domain Adaptation

Closed set domain adaptation analyses the performance for the 8 largest action classes: (‘put’, ‘take’, ‘open’, ‘close’, ‘wash’, ‘cut’, ‘mix’, and ‘pour’), which form 80% of the training action segments for these domains. This ensures sufficient examples per domain and class, without balancing the training set. The label imbalance of these 8 classes is depicted in Fig. 4.5 which also shows the differing distribution of classes between the domains. Most domain adaptation works evaluate on balanced datasets [3, 4, 60] with few using imbalanced datasets [78]. EPIC-Kitchens has a large class imbalance offering additional challenges for domain adaptation. The number of action segments in each domain are specified in Table 4.1, where a segment is a labeled start/end

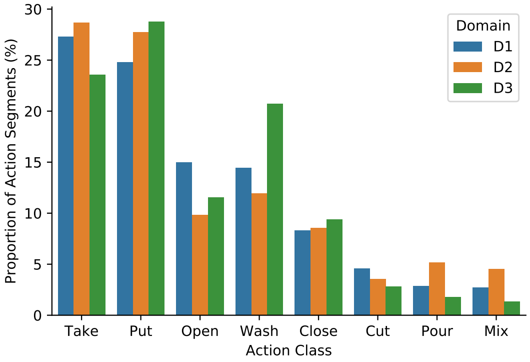


Figure 4.5: Class distribution per domain, for the 8 classes in legend.

Domain	D1	D2	D3
Ref. EPIC Kitchen	P08	P01	P22
Training Instances	1543	2495	3897
Test Instances	435	750	974

Table 4.1: Number of action segments per domain in the closed-set benchmark.

time, with an action label.

4.2.2 Open Set Domain Adaptation

A more challenging scenario exists when the label spaces between source and target differ, this is referred to as Open Set Domain Adaptation. In this setting we evaluate on all verbs and nouns collected in each domain. Note that some verbs and nouns may present in source and not in target, and visa versa. Whilst the majority of our analysis focuses on the closed-set domain adaptation (Sec. 4.3), we showcase that our self-supervised alignment objective can improve target performance in the open-set scenario (Sec. 4.4).

Tables 4.2 and 4.3 show quantitative statistics for various training sets of the domains in the open set benchmark. Approximately half of the classes are not shared between the domains, as shown in Table 4.2 which states the Intersection over Union (IoU) between the classes in the source and target domains. Table 4.3 states the percentage of target video clips with a class label present in the source. While the vast majority of target videos (greater than 95%) belong to a shared verb class, less than 85% of target videos belong to a shared noun class. Therefore, open-set domain adaptation will be more challenging for classification of nouns than verbs.

		Target		
		D1	D2	D3
Source	D1	-	0.47	0.49
	D2	0.47	-	0.51
	D3	0.49	0.51	-

(a) Verbs

		Target		
		D1	D2	D3
Source	D1	-	0.38	0.47
	D2	0.38	-	0.40
	D3	0.47	0.40	-

(b) Nouns

Table 4.2: IoU of verb and noun classes between domains in the open-set benchmark

		Target		
		D1	D2	D3
Source	D1	-	0.95	0.96
	D2	0.95	-	0.98
	D3	0.94	0.96	-

(a) Verbs

		Target		
		D1	D2	D3
Source	D1	-	0.74	0.85
	D2	0.74	-	0.75
	D3	0.79	0.77	-

(b) Nouns

Table 4.3: Percentage of target videos from a class in the source domain, in the open-set domain adaptation benchmark

4.3 Closed Set Adaptation Experiments with RGB and Optical Flow[3]

First, the architecture and implementation details are outlined in Sec. 4.3.1, followed by a comparison against baseline methods noted in Sec. 4.3.2. Results are presented in Sec. 4.3.3, with an

[3] Results from the CVPR paper [40]

ablation study of the method's components in Sec. 4.3.4 and qualitative results including feature space visualisations are provided in Sec. 4.3.5. An analysis of the proposed domain adaptation strategy on individual modalities, and a comparison to previous work using only RGB [1, 2], is provided in Sec. 4.3.6.

4.3.1 Implementation Details

Architecture. The inflated 3D convolutional architecture (I3D) [26] is used as the backbone for feature extraction, for each modality (F^m), and is trained end-to-end with action classifier, G. In this work, F convolves over a temporal window of 16 frames from videos sampled at 30 frames per second. For short action clips that contain less than 16 frames, temporally adjacent frames to the action clip are used. This improves action recognition performance over the original I3D sampled strategy, which repeats frames from the beginning of the clip. In training, a single temporal window is randomly sampled from within the action segment each iteration. In testing, as in [19], we use an average over 5 temporal windows, equidistant within the segment. The weights of F were initialised from I3D pretrained on Kinetics [26]. The RGB and optical flow frames were provided publicly [30]. The output of F is the result of the final average pooling layer of I3D, with 1024 dimensions. G is a single fully connected layer with a softmax activation to predict class labels. Each domain discriminator D^m is composed of 2 fully connected layers with a hidden layer of 100 dimensions and a ReLU activation function. A dropout rate of 0.5 was used on the output of F and $1e - 7$ weight decay for all parameters. Batch normalisation layers are used in F^m and are updated with target statistics for testing, as in AdaBN [101]. For data augmentation, random crops, scale jitters and horizontal flips are applied to videos as in [19]. Image widths and heights are scaled by $\{1, 0.875, 0.75\}$ during scale jittering and are resized to 224. During testing, only center crops are used. The self-supervised correspondence function C (Eq. 4.3) is implemented as 2 fully connected layers of 100 dimensions and a ReLU activation function. The features from both modalities are concatenated along the channel dimension as input to C.

Training and Hyper-parameter Choice. Models are trained using the SGD optimiser and momentum in two stages. First the network is trained with only the classification and self supervision losses, $\mathcal{L}_y + \lambda_c \mathcal{L}_c$, at a learning rate of $1e-2$ for 3K iterations. Then, the overall loss function (Eq. 4.4) is optimised, applying the domain adversarial losses $\mathcal{L}_d^m$, and reducing the learning rate to $2e-4$ for a further 6K steps. The self-supervision hyper-parameter, $\lambda_c = 5$ was chosen by observing the performance on the labelled **source domain** only, *i.e.* this has not been optimised for the target domain. Note that while training with self-supervision, half the batch contains corresponding modalities and the other non-corresponding modalities. Only source examples with corresponding modalities are used to train for action classification. The domain adversarial hyper-parameter, $\lambda_d = 1$, was chosen arbitrarily; we show that the results are robust to some variations in this hyper-parameter in an ablation study. Batch size was set to 128, split equally for source and target samples. On average, training takes 9 hours on an NVIDIA DGX-1 with 8 V100 GPUs.

4.3.2 Baselines

For all results, the top-1 target accuracy is reported, and averaged over the last 9 epochs of training for robustness. The first baseline evaluates the impact of domain shift between source and target by testing using a multi-modal source-only model (MM source-only), trained with no access to unlabelled target data. Additionally, 3 baselines are proposed for unsupervised domain adaptation as follows:

- *AdaBN* [101]: Batch Normalisation layers are updated with target domain statistics (see Sec 2.3.4 for further details).

- *Maximum Mean Discrepancy (MMD)*: The multiple kernel implementation of the commonly used domain discrepancy measure MMD is used as a baseline [73] (see Sec 2.3.2 for further details). This directly replaces the adversarial alignment with separate discrepancy measures

	D2→ D1	$D3 \rightarrow D1$	$D1 \rightarrow D2$	$D3 \rightarrow D2$	$D1 \rightarrow D3$	$D2 \rightarrow D3$	Mean
MM Source-only	42.5	44.3	42.0	**56.3**	41.2	46.5	45.5
AdaBN [101]	44.6	47.8	47.0	54.7	40.3	48.8	47.2
MMD [73]	43.1	48.3	46.6	55.2	39.2	48.5	46.8
MCD [5]	42.1	47.9	46.5	52.7	43.5	51.0	47.3
MM-SADA	**48.2** ▲+5.7	**50.9**▲+6.6	**49.5**▲+7.5	56.1▼-0.2	**44.1**▲+2.9	**52.7** ▲+6.3	**50.3** ▲+4.8
Supervised target	62.8	62.8	71.7	71.7	74.0	74.0	69.5

Table 4.4: Top-1 accuracy on the target domain, for MM-SADA, compared to different alignment approaches. On average, MM-SADA outperforms the source-only performance by 4.8%.

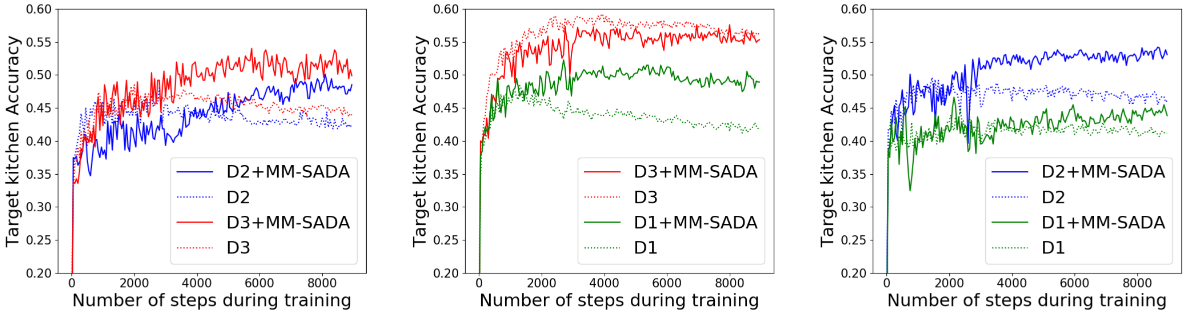
applied to individual modalities.

– *Maximum Classifier Discrepancy (MCD)* [5]: Alignment through classifier disagreement is used. This uses two multi-modal classification heads as separate classifiers. The classifiers are trained to maximise prediction disagreement on the target domain, implemented as L1 loss, finding examples out of support from the source domain (see Sec 2.3.3 for further details). We use a GRL to optimise the feature extractors.

Additionally, as an upper limit, we also report the supervised target domain results. This is a model trained on labelled target data and only offers an understanding of the upper limit for these domains. Supervised target performance is highlighted in the table to avoid confusion.

4.3.3 Comparative Results

First, the proposed method, MM-SADA, is compared to the various domain alignment techniques in Table 4.4. This shows that MM-SADA outperforms batch-based [101] (by 3.1%), classifier discrepancy [5] (by 3%) and discrepancy minimisation alignment [73] (by 3.5%) methods. The improvement is consistent for all pairs of domains. Additionally, it significantly improves on the



(a) Target D1

(b) Target D2

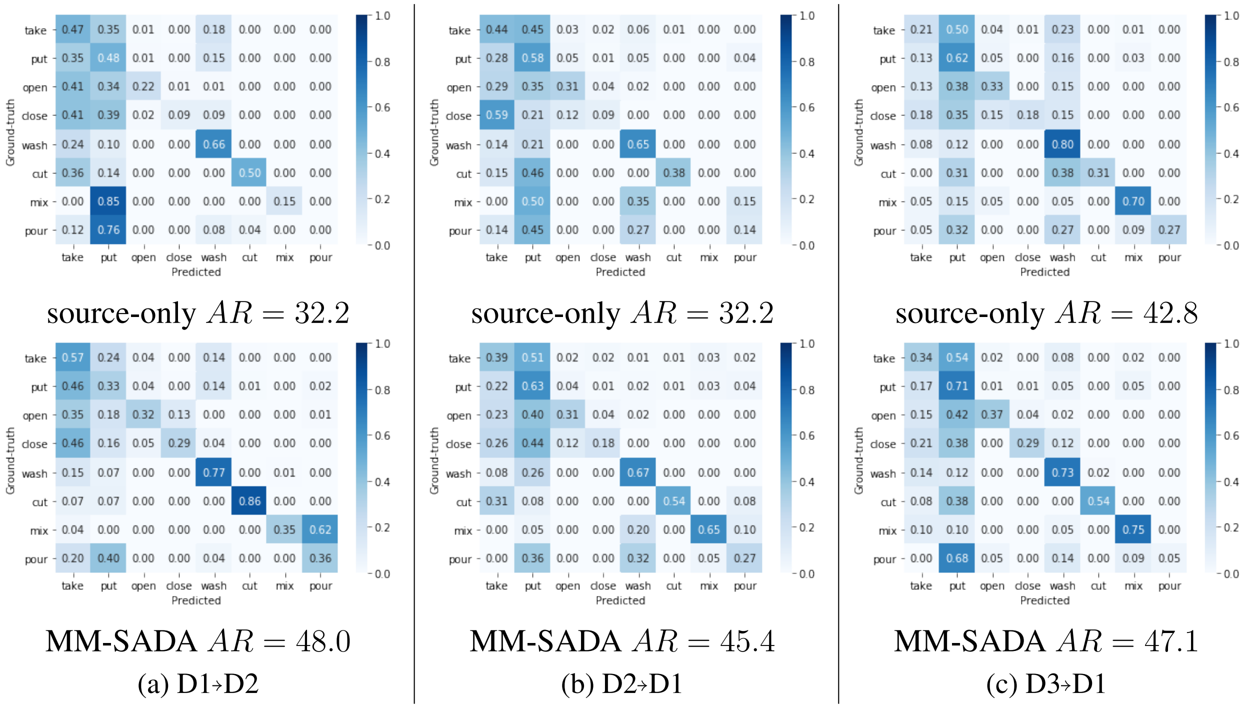
(c) Target D3

Figure 4.6: Accuracy on target during training epochs. Solid line is MM-SADA and dotted line is source-only performance.

source-only baseline by up to 7.5% in 5 out of 6 cases. For a single case, $D3 \rightarrow D2$, all baselines under-perform compared to source-only. MM-SADA has a slight drop (-0.2%) but outperforms other alignment approaches. The small performance drop is revisited in the ablation study.

Figure 4.6 shows the top-1 accuracy on the target during training (solid lines) vs source-only training without domain adaptation (dotted lines). Training without adaptation has consistently lower accuracy, except for our failure case $D3 \rightarrow D2$, showing the stability and robustness of MM-SADA during training, with minimal fluctuations due to stochastic optimisation on batches. This is essential for UDA as no target labels can be used for early stopping.

Figure 4.7 compares the performance of individual classes for MM-SADA and source-only. Confusion matrices are normalised with respect to the number of instances in the respective ground-truth class. For clarity, we show three pairs of domains: $D1 \rightarrow D2$, $D2 \rightarrow D1$ and $D3 \rightarrow D1$. The additional pairs of domains can be found in Appendix A.2. MM-SADA increases the recall of the majority of actions classes, over source-only. This is shown by the more pronounced diagonal of the confusion matrix produced by MM-SADA, and the increase in Macro-averaged Recall (AR). Overall, the classes with the fewest examples (*cut*, *mix* and *pour*) show the greatest improvement in recall, as shown in the confusion plots of $D1 \rightarrow D2$ and $D2 \rightarrow D1$. For $D3 \rightarrow D1$, the recall for 6



source-only $AR = 32.2$ source-only $AR = 32.2$ source-only $AR = 42.8$

MM-SADA $AR = 48.0$ MM-SADA $AR = 45.4$ MM-SADA $AR = 47.1$

(a) D1→D2 (b) D2→D1 (c) D3→D1

Figure 4.7: Confusion matrices and Macro-averaged Recall (AR) of MM-SADA (bottom) and source-only (top). Our method improves performance for a number of different classes.

out of 8 classes increases, with only *pour* and *wash* decreasing in performance.

4.3.4 Ablation Studies

The first ablation study assesses the impact of the different alignment losses. The second analyses several design decisions of the self-supervised task, comparing the proposed method to: the self-supervised objective trained only on source, and to an alternative sampling strategy to predict temporal correspondence.

	λ_d	λ_c	$D2 \rightarrow D1$	$D3 \rightarrow D1$	$D1 \rightarrow D2$	$D3 \rightarrow D2$	$D1 \rightarrow D3$	$D2 \rightarrow D3$	Mean
source-only	0	0	42.5	44.3	42.0	56.3	41.2	46.5	45.5
MM-SADA (Self-Supervised only)	0	5	41.8	49.7	47.7	**57.4**	40.3	50.6	47.9▲+2.4
MM-SADA (Adversarial only)	1	0	46.5	51.0	50.0	53.7	43.5	51.5	49.4▲+3.9
MM-SADA (Adversarial only)	0.5	0	46.9	50.2	50.2	53.6	**44.7**	50.8	49.4▲+3.9
MM-SADA	0.5	5	45.8	**52.1**	**50.4**	56.9	43.5	51.9	50.1▲+4.6
MM-SADA	1	5	**48.2**	50.9	49.5	56.1	44.2	**52.7**	**50.3**▲+4.8

Table 4.5: Ablation of MM-SADA, showing the contribution of the various loss functions (Eq 4.4). When $\lambda_d = 0$, modality adversarial is not utilised. When $\lambda_c = 0$, self-supervision is not utilised.

Ablation on alignment losses

Table 4.5 compares the individual contributions of different components of MM-SADA. The self-supervised component on its own gives a 2.4% improvement over no adaption. This shows that self-supervision can learn features common to both source and target domains, adapting the model to better fit the target domain. Importantly, this on average outperforms the three baselines in Table 4.4. Adversarial alignment per modality gives a further 2.4% improvement as this encourages the source and target distributions to overlap, removing domain specific features from each modality. Compared to adversarial alignment only, MM-SADA improves in 5 of the 6 domains and by up to 3.2%.

For the single pair noted in Sec. 4.3.3, $D3 \rightarrow D2$, self-supervision alone outperforms source-only and all other methods reported in Table 4.4 by 1.1%. However when combined with domain adaptation using $\lambda_d = 1$, the overall performance of MM-SADA reported in Table 4.4 cannot beat the baseline. In Table 4.5, we show that when halving the contribution of adversarial component to $\lambda_d = 0.5$, MM-SADA can achieve 56.9% outperforming the source-only baseline. Therefore, self-supervision can improve performance where marginal alignment domain adaptation techniques fail.

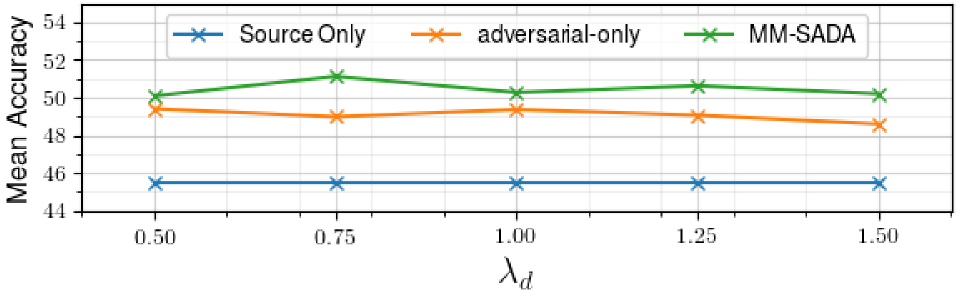


Figure 4.8: Robustness of the average top-1 accuracy over all pairs of domains for various λ_d on the target domain.

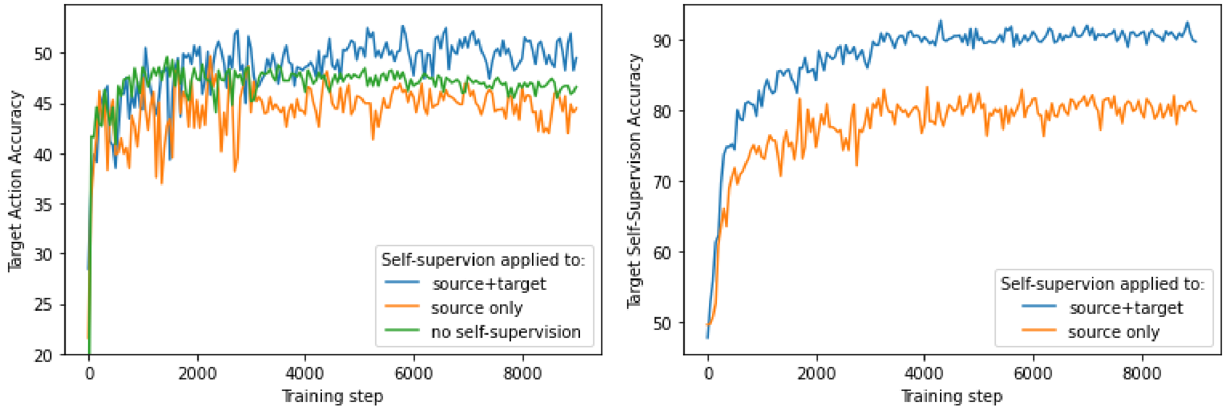
Self-Supervision	$D2 \rightarrow D1$	$D3 \rightarrow D1$	$D1 \rightarrow D2$	$D3 \rightarrow D2$	$D1 \rightarrow D3$	$D2 \rightarrow D3$	**Mean**
Sync.	44.2	50.2	48.0	54.6	41.0	49.4	**47.9**
Seg. Corr.	41.8	49.7	47.7	57.4	40.3	50.6	**47.9**

Table 4.6: Comparision of two self-supervision tasks for modality correspondence: determining modality synchrony vs. determining whether modality samples come from the same segment. The two approaches perform comparably on average.

Figure 4.8 plots the performance of MM-SADA as λ_d changes. Note that λ_c can be chosen by observing the performance of self-supervision on source-domain labels, while λ_d requires access to target data. We show that our approach is robust to various values of λ_d, with even higher accuracy at $\lambda_d = 0.75$ than those reported in Table 4.5.

The choice of self-supervised task

Figure 4.9 compares training the self-supervised task jointly on both source and target domains, to training only on the source domain. For the self-supervised task applied only to source, the batch-normalisation layers are updated with target data to conduct a fair comparison. Figure 4.9a shows that applying the self-supervised task jointly to both domains (BLUE), is a necessity to im-



(a) Action accuracy on Target

(b) Self-supervised objective accuracy on Target

Figure 4.9: The impact of training our self-supervised objective on both Source and Target or only Source. Training the self-supervised objective on both domains increases the accuracy of action recognition (a) and the self-supervised objective on Target (b).

prove target domain performance. If self-supervision is applied only to source (ORANGE), target performance will decrease compared to only training with the action classification loss (GREEN). Training on target data learns target specific features needed to solve the correspondence task, leading to the increase in action recognition performance. Figure 4.9b compares the accuracy of the self-supervised task on target test data. Applying the self-supervised loss to target data improves temporal correspondence prediction accuracy by over 10%.

Two approaches for multi-modal self-supervision are compared in Table 4.6. The first, which has been used to report all results above, learns the correspondence of RGB and Flow within the same action segment. This is referred as *'Seg. Corr.'*. The second learns the correspondence only from time-synchronised RGB and Flow data, which is referred to as *'Sync'*. The two approaches are comparable in performance overall, with no difference on average over the domain pairs. This shows the potential to use a number of multi-modal self-supervision tasks for alignment.

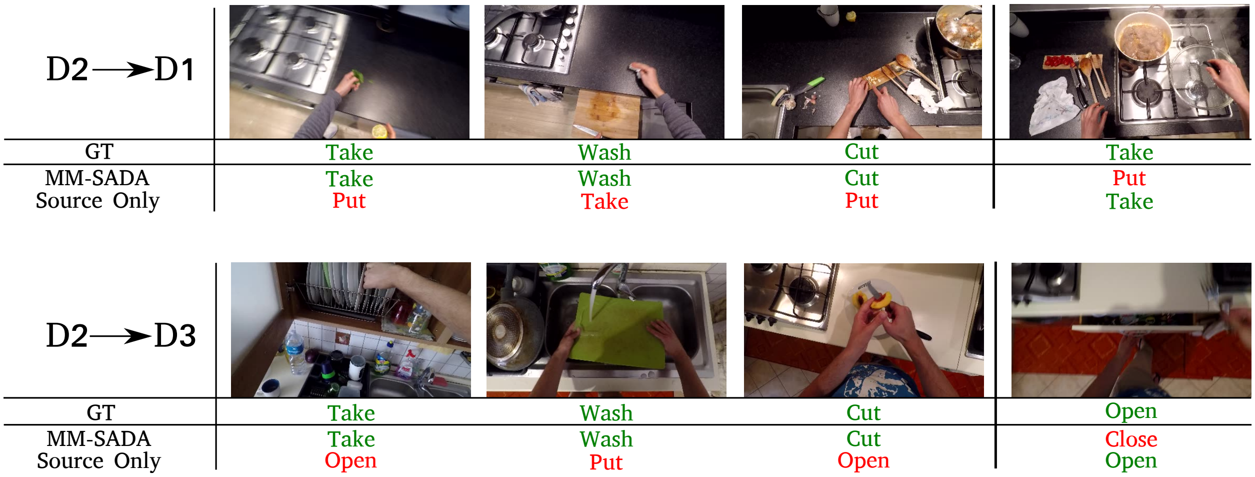


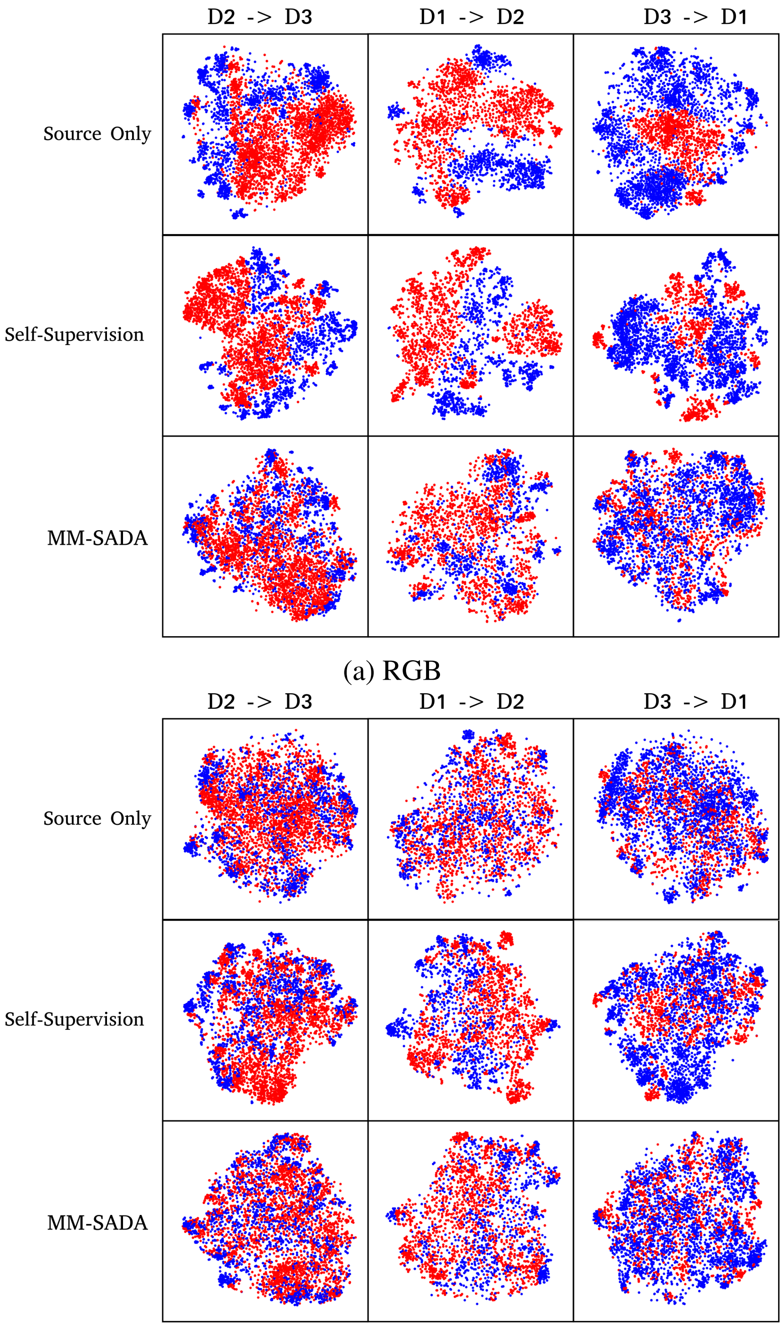
Figure 4.10: Qualitative results for MM-SADA and source-only, showing success and failure cases.

4.3.5 Qualitative Results

Figure 4.10 shows qualitative results of MM-SADA relative to source-only performance, with three success cases and one failure case for two pairs of domains. Without adaptation, models cannot utilise appropriate visual cues in the target environment, *i.e.* appearance of a chopping board, knife, sink or tap, therefore the model fails to predict cut and wash. Both adapted and non-adapted models struggle with ambiguous examples, where different actions are occurring using both hands.

Figure 4.11 shows the t-SNE [231] visualisation of the RGB (Fig. 4.11a) and Flow (Fig. 4.11b) feature spaces F^m. Several observations are worth noting from this figure. First, Flow shows higher overlap between source and target features pre-alignment (first row). This shows that Flow is more robust to environmental changes. Second, self-supervision alone (second row) changes the feature space by separating the features into clusters, that are potentially class-relevant. This is most evident for $D3 \rightarrow D1$ on the RGB modality (second row, third column). However, alone this feature space still shows domain gaps, particularly for RGB features. Third, MM-SADA (third row) aligns the marginal distributions of source and target domains.

Figure 4.12 shows a video from the source domain (D3), and a video from the target domain



(a) RGB

(b) Flow

Figure 4.11: t-SNE plots of RGB and Flow feature spaces produced by source-only, self-supervised alignment and the proposed model, MM-SADA. Target is shown in red and source in blue. Our method better aligns both modalities.

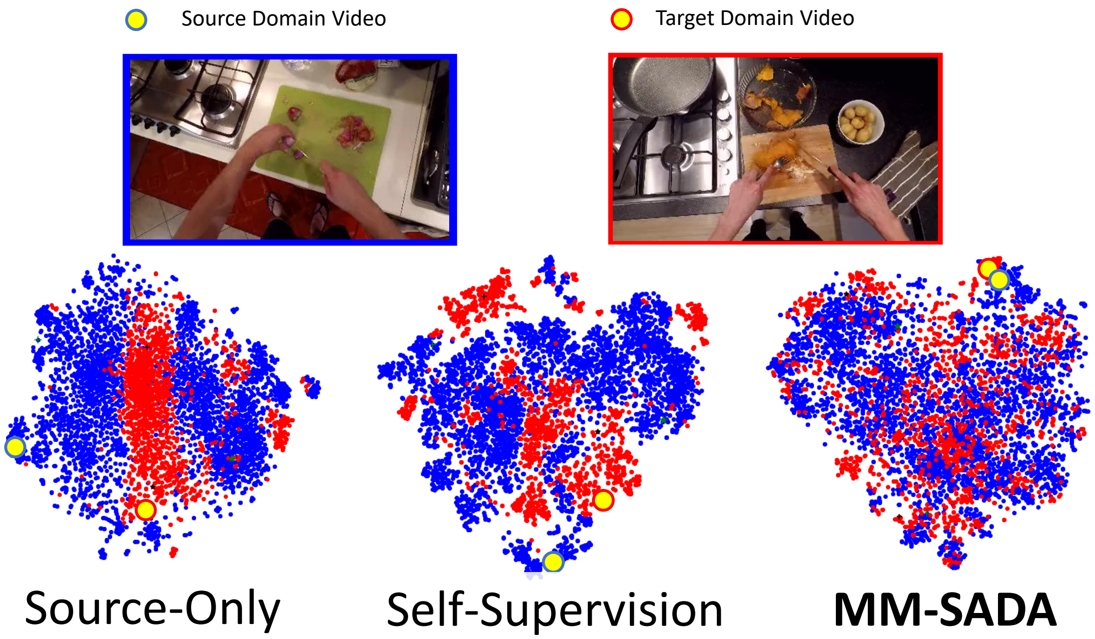


Figure 4.12: A source and target video of the same action class, cut, highlighted in the RGB feature space of source-only, self-supervised and MM-SADA models. The videos are better aligned with MM-SADA.

(D1), in the feature space produced by the source only, self-supervised and MM-SADA models. Both videos depict the same action class, cut. The change in location and objects leads to the videos being far apart in the features space of the source-only model. However, with self-supervision the model becomes more robust to environmental changes and the videos become closer in feature space, but not fully aligned. By combining self-supervision with adversarial alignment, MM-SADA, the domains are aligned. Other methods that could show the additional invariance of MM-SADA to environmental changes is Grad-CAM [232]. Grad-CAM highlight which regions of the image are used by the network for the action recognition task. We hypothesise the network would focus more on pixels where actions take place, rather than the background environment.

4.3.6 Performance of individual modalities

In order to assess the performance of individual modalities, the multi-modal action classification loss is replaced with a separate loss per modality. The classification loss for RGB and Flow are defined in Eq. 4.5. The self-supervision loss is still applied to both modalities.

$$\mathcal{L}_y^m = \sum_{x \in \{\mathbf{S}\}} -y \log \sigma\big(G^m(F^m(x^m))\big) \qquad \forall m \in \{RGB, FLOW\} \tag{4.5}$$

Baselines

We compare against recent video domain adaptation methods[4], DAAA [2] and TA3N [1], which align the RGB modality. Note that DAAA directly applied adversarial training to 3D convolutions, so it is equivalent to our adversarial-only ablation on the RGB modality. TA3N applies domain adversarial training to parts of a Temporal Relation Module [20], which takes extracted features as input. For a fair comparison we extract features from our I3D source-only network trained on the RGB modality as input features for TA3N.

[4] An explanation of these methods is provided in Chapter 3, Sec. 3.3

	$D2 \rightarrow D1$	$D3 \rightarrow D1$	$D1 \rightarrow D2$	$D3 \rightarrow D2$	$D1 \rightarrow D3$	$D2 \rightarrow D3$	Mean
RGB source-only	33.9	39.0	38.8	46.4	34.2	37.9	38.4
RGB (AdaBN)	34.0	39.8	42.4	45.7	35.9	40.6	39.7
RGB (self-supervised-only)	37.4	**43.2**	45.0	**50.1**	40.9	45.0	43.6
DAAA / RGB (adversarial-only)	42.1	41.6	**47.2**	44.2	38.9	44.5	43.1
RGB (MM-SADA)	**44.9**	42.1	43.8	48.8	**43.6**	**48.6**	**45.3**
TA3N (source-only)	34.9	41.1	38.8	44.6	36.5	40.0	39.5
TA3N	35.9	**43.2**	41.1	43.5	40.0	41.4	40.9
Flow source-only	42.8	47.0	51.1	61.2	40.4	47.7	48.4
Flow (AdaBN)	44.4	48.6	51.6	60.6	43.3	48.0	49.4
Flow (self-supervised-only)	45.5	51.0	50.9	59.7	45.1	51.1	50.6
Flow (adversarial-only)	**47.9**	49.8	**54.7**	**61.6**	**48.3**	**54.9**	**52.9**
Flow (MM-SADA)	46.8	**51.7**	53.2	60.0	47.8	52.5	52.0

Table 4.7: Ablation of MM-SADA on individual modalities, comparing to previous domain adaptation works on RGB, TA3N [1] and DAAA [2].

Results

Table 4.7 shows the impact of the proposed method on the performance of the modalities individually. Predictions are taken from each modality separately before late fusion. RGB, the less robust modality, benefits most from MM-SADA, improving over source-only by 6.9% on average, whereas Flow improves by 1.6%. The inclusion of multi-modal self-supervision provides 5.2% and 2.2% improvements for RGB and Flow, compared to only using adversarial alignment. This shows the benefit of employing self-supervision from multiple modalities during alignment. While Flow is generally more robust than RGB, it still benefits from multi-modal supervision with the less robust modality, as well as adversarial training. Future research should focus on how to effectively combine adversarial training with self-supervision, as the Flow modality with MM-SADA has worse performance than with adversarial training alone.

Table 4.7 also compares the various components of MM-SADA to TA3N. Note that target performance is larger for TA3N trained only on source (TA3N source-only) than MM-SADA trained only on source. This is because TA3N models longer temporal structures with the temporal relation module. However, TA3N is less effective for domain adaptation with target data than both the proposed self-supervised and adversarial alignment strategies. This shows that back-propagating alignment losses through the feature extractor leads to better target domain performance, as TA3N acts on extracted features. Note that the proposed self-supervised loss alone outperforms TA3N.

Given that Flow is expensive to compute, it is advantageous to avoid Flow for time critical applications with a small computation cost budget. This section showed that exploiting optical flow greatly benefits domain adaptation for RGB. For deployment, action recognition can be performed purely on the RGB modality, while benefiting from the robustness learned from Flow during training.

4.4 Open Set Adaptation Experiments with RGB, Optical Flow and Audio

Audio has been shown to improve first-person action recognition due to the distinct sounds from different actions [23]. Many sounds will also be robust to participant and environmental changes, as actions performed with objects of similar materials will produce similar sounds. This section shows that self-supervision with audio improves domain adaptation. In addition, experiments are conducted in the more realistic open-set setting, where the number of classes is not restricted. The self-supervised domain adaptation strategy improves the performance of recognising both the type of action (verbs) and the interacted objects (nouns).

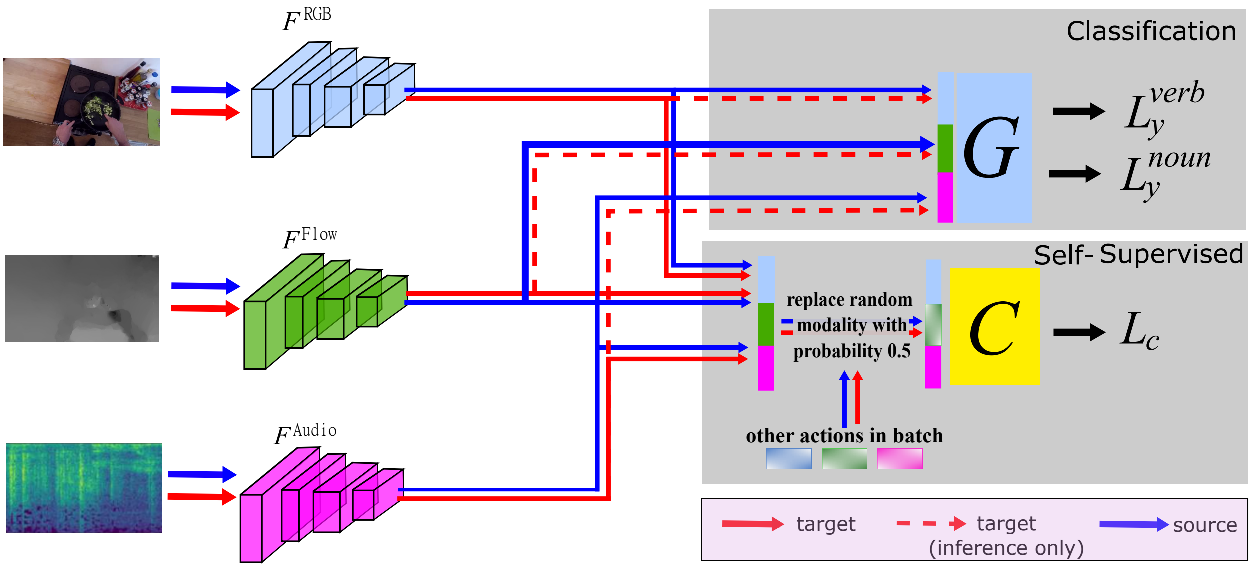


Figure 4.13: Temporal Binding Window Network (TBN) [23] with multi-modal self-supervision. A random modality is replaced with another instance from the batch to generate non-corresponding examples.

4.4.1 Experimental Setup

Multi-modal self-supervision is applied to the state-of-the art multi-modal architecture, TBN [23]. Fig. 4.13 shows the experimental setup with self-supervision applied to RGB, optical flow and audio. Note that the adversarial alignment component was removed due to instability during training with the chosen architecture, which led to inconsistent results between training runs. The minimax game played between the feature extractor and the domain discriminator is known to suffer from instability issues [233]. There are several key changes from the architecture in Fig. 4.2: the classifier, G, acts on the concatenation of all modalities, and instances of non-corresponding modalities are sampled in feature space instead of at the input. Individual components of the architecture and training procedure are discussed in more detail as follows:

Backbone. The backbone for each modality, F, is BN-inception. F^{RGB} and F^{Audio} are pretrained on ImageNet and F^{Flow} is pretrained on Kinetics. This was the pretraining strategy used in TBN [23], which was shown to be effective. Audio is extracted as in TBN [23], creating a 2D spectrogram centered from a 1.28 second window around the current frame. RGB and optical flow frames were provided publicly [30]. Only the first batch normalisation layer is updated in F, the rest are frozen and use the statistics from the pretrained models.

Action and Correspondence Classifiers. Both the action classifier G, and correspondence classifier, C, are multi-layer perceptions with hidden dimensions of size 512. They take as input a concatenated representation from all modalities followed by a multi-layer perception. G produces 125 logits for verbs and 352 logits of nouns. C is a binary classifier.

Random Modality Replacement. In order to generate non-corresponding examples, a random modality is replaced by another instance in the mini-batch. Replacement is only applied to half the batch, with the other half containing corresponding modalities. This enables all source inputs to be used for action classification, while still generating non-corresponding instances for the self-supervised task.

Training. The network was trained with SGD and momentum, with a learning rate of 0.01 for 120 epochs and 0.001 for a further 40 epochs. The momentum value was 0.9. Dropout of probability 0.5 was used on the activations of the hidden layers of C and G. A batch size of 128 was used for both source and target domains. The self-supervised weighting was $L_c = 5$ and no adversarial domain alignment was used, $L_d = 0$. During training, a single video segment is used to reduce GPU memory requirements of the model, rather than the consensus of multiple segments used in TBN [23]. At test time, the consensus of 25 segments in the video is used. The network is trained with the same augmentation strategy as TBN [23].

4.4.2 Results

Table 4.8 shows the action recognition accuracy on the target domains for verbs and nouns, respectively. The addition of the self-supervised objective (MM-SADA), improves performance

Metric	Method	$D2 \to D1$	$D3 \to D1$	$D1 \to D2$	$D3 \to D2$	$D1 \to D3$	$D2 \to D3$	mean
Verb	Source-only	28.3	27.0	32.1	45.8	27.6	42.1	33.8
	MM-SADA	25.4	33.2	44.1	49.3	35.6	43.7	38.5▲+4.70
Noun	Source-only	11.3	10.2	11.8	19.0	10.3	19.3	13.6
	MM-SADA	12.7	14.0	14.1	24.2	10.7	21.5	16.2▲+2.60

Table 4.8: Impact of multi-modal self-supervision (RGB, flow and audio) on the open-set domain adaption benchmarks. Both verb and noun classification improve with the self-supervised loss.

compared to source-only by 4.7% for verbs and 2.6% for nouns, on average. MM-SADA improves verb classification performance for all but one benchmark, $D2 \to D1$, however, noun classification performances increases for all benchmarks. This shows that self-supervision is beneficial for multiple tasks, and is useful in the open-set setting, where less than 85% of target videos have a noun label from a class in the source domain.

To asses the impact of self-supervision on each modality, several experiments were conducted to answer two questions: which modalities should be included for domain adaptation, and what is the performance of each modality on the self-supervised task? To reduce computation, these experiments were only conducted on the $D2 \to D3$ open set benchmark.

Which modalities should be included for domain adaptation?

Table 4.9a reports the target domain performance of the TBN architecture trained with two modalities as input (RGB and audio or RGB and Flow), and with all three modalities (RGB, Flow and audio). For each architecture, target performance with our proposed self-supervised loss is compared to no self-supervision (source-only). The proposed self-supervised objective provides the largest improvement over source-only with RGB and audio as input, with an 8.7% and 7.1% increase in accuracy for verbs and nouns. Self-supervision between RGB and Flow also improves

Metric	Method	RGB+Audio	RGB+Flow	RGB+Flow+Audio
Verb	Source-only	29.19	32.85	42.11
	MM-SADA	37.89▲+8.70	35.45▲+2.60	43.74▲+1.63
Noun	Source-only	13.50	16.18	19.27
	MM-SADA	20.57▲+7.07	17.64▲+1.46	21.54▲+2.27

(a) Accuracy

Metric	Method	RGB+Audio	RGB+Flow	RGB+Flow+Audio
Verb	Source-only	4.58	6.77	10.95
	MM-SADA	9.39▲+4.81	8.87▲+2.10	12.69▲+1.63
Noun	Source-only	4.71	6.32	7.66
	MM-SADA	7.25▲+2.54	6.98▲+0.66	8.60▲+0.94

(b) Macro-averaged Recall

Table 4.9: Action recognition performance on the target domain for different combinations of modalities. Experiment conducted on the open-set benchmark: $D2 \rightarrow D3$. All three modalities provide the highest performance, and self-supervision consistently outperforms source-only.

target performance, albeit not as much as RGB and audio. This shows the benefit of including audio in the self-supervision objective. All three modalities as input to the model achieves the best target domain performance. This is due to the better modelling of actions in the source domain. Table 4.9b reports Macro-averaged Recall on the target domain, such that all classes are given equal weighting in the metric. Macro-averaged Recall is consistently larger than source-only, therefore, self-supervision must improve both the minority and majority classes of verbs and nouns.

What is the performance of each modality on the self-supervised task?

Fig. 4.14 ablates the accuracy of the self-supervised task for each modality. For 50% of instances, the ablated modality is replaced with another instance in the batch, and the remaining instances contain corresponding modalities. The analysis is conducted on the model trained with RGB, Flow and audio as input. Ablating RGB and Flow produces a similar accuracy, which averaged across both domains is 83.6% and 84.3%, respectively. The correspondence accuracy of audio is smaller, with 71.8% target accuracy, showing that learning the temporal alignment between audio with the other modalities is a harder task to learn than between RGB and Flow.

4.5 Conclusion

This chapter proposed a multi-modal domain adaptation approach for fine-grained action recognition, utilising multi-modal self-supervision and adversarial training per modality. The self-supervised task of predicting the correspondence of multiple modalities was shown to be an effective domain adaptation method. On its own, this can outperform domain alignment methods [5, 73], by jointly optimising for the self-supervised task over both domains. In addition, self-supervision can be applied to a number of different modalities, showing improved target performance with RGB, optical flow and audio, on both the classification of verbs and nouns. Together

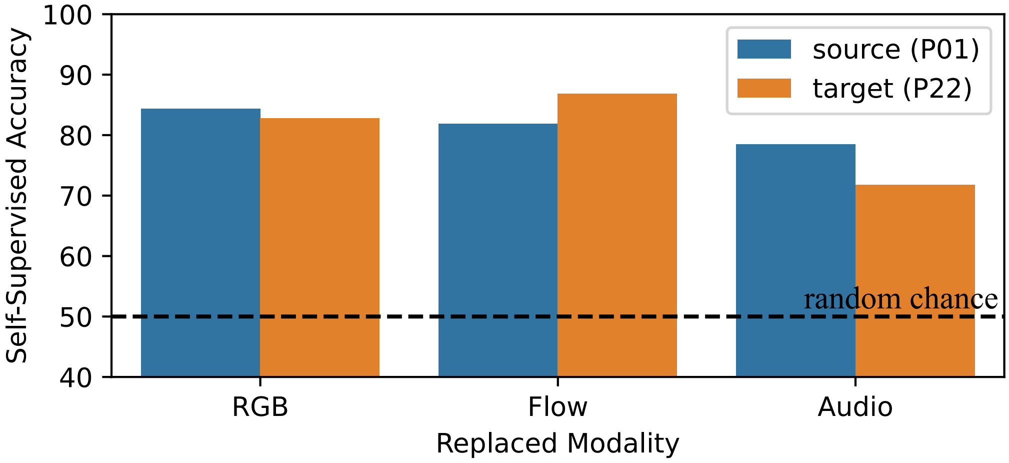


Figure 4.14: Accuracy of correspondence classifier when replacing a specified modality for 50% of the input instances. Experiment conducted on the open-set benchmark: $D2 \rightarrow D3$.

with adversarial training, the proposed approach outperforms non-adapated models by 4.8% on verb classification in the closed-set setting. In conclusion, aligning individual modalities whilst learning a self-supervision task on source and target domains can improve the ability of action recognition models to transfer to unlabelled environments.

This chapter has assumed each multiple-modality has a differing robustness to domain changes. Our self-supervised alignment loss may not be beneficial if all modalities have a similar level of robustness to domain changes. Additionally, we assumed each action was segmented, i.e. an action detector already exists in the target domain. This is exploited by adversarial alignment objective, which would be impacted by additional source and target classes in both domains. Our self-supervised object does not align marginal distributions and is applied on a per-instance basis, so will be more robust to additional classes in each domain.

Chapter 5

The EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition

The majority of fine-grained action recognition datasets collect data over a short recording period [12, 13, 14, 15, 30, 31], where video footage from each participants is collected over a few days. The data can be split into training and test sets in different ways. Commonly the same recording environment, recording equipment and activities are present in both splits [12, 13, 14, 15, 31]. The advantage of that strategy is that the train and test sets are sampled from the same distribution, but the disadvantage is practicality. Participants may change homes, interact with novel objects and tools, or be filmed with a different type of camera, after the initial dataset collection. Therefore, the performance of action recognition models may drop over time. To reduce the cost of annotating footage over later recording periods, this chapter proposes to adapt models to environmental changes by collecting unlabelled data.

While Chapter 4 focused on bias introduced by filming in different locations, this chapter focuses on different domain shifts that naturally occur due to the collection of fine-grained action recognition dataset. This chapter uses a greater variety of different locations within each domain, therefore, location bias is no longer the leading causes of the domain shift. However, changes in camera equipment and interactions with novel objects and tools still has a large impact on the target domain performance. This chapter highlights the need for domain adaptation methods to consider these visually subtle causes of domain shift, and provides a benchmark for the research community.

The EPIC-KITCHENS-100 dataset collects videos over two distinct recording periods, taken two years apart, which introduces a domain gap within the dataset (Sec. 5.1). This chapter proposes the Unsupervised Domain Adaptation Challenge (Sec. 5.2), where the task is to adapt action recognition models, trained on the initial recording period, to the footage taken two years later. A carefully chosen train, validation and test split is proposed. How to form a validation set is a particular issue for unsupervised domain adaptation, since no labelled data from the target domain should be used during training. The difficulty of the benchmark is showcased across a series of baselines, evaluated across the multiple modalities available in the dataset: appearance, motion and audio (Sec. 5.3). The domain adaptation baseline, TA3N [1], was adapted for the challenge and acts on extracted features that are made publicly available. A public competition for the challenge was hosted on Coda-Lab, where the winning solutions proposed multi-modal solutions (Sec. 5.4).

5.1 The Domain Gap in EPIC-KITCHENS-100

EPIC-KITCHENS-100 extended EPIC-KITCHENS-55 [41] with additional footage collected two-years later. Videos in the extension were recorded by 16 subjects that participated in EPIC-KITCHENS-55. Interestingly, half (8 subjects) had moved homes over the past two years. All participants were asked to collect 2–4 days of their typical kitchen activities, as in EPIC-KITCHENS-55. Footage was captured using a head mounted GoPro Hero7 black. This is two generations

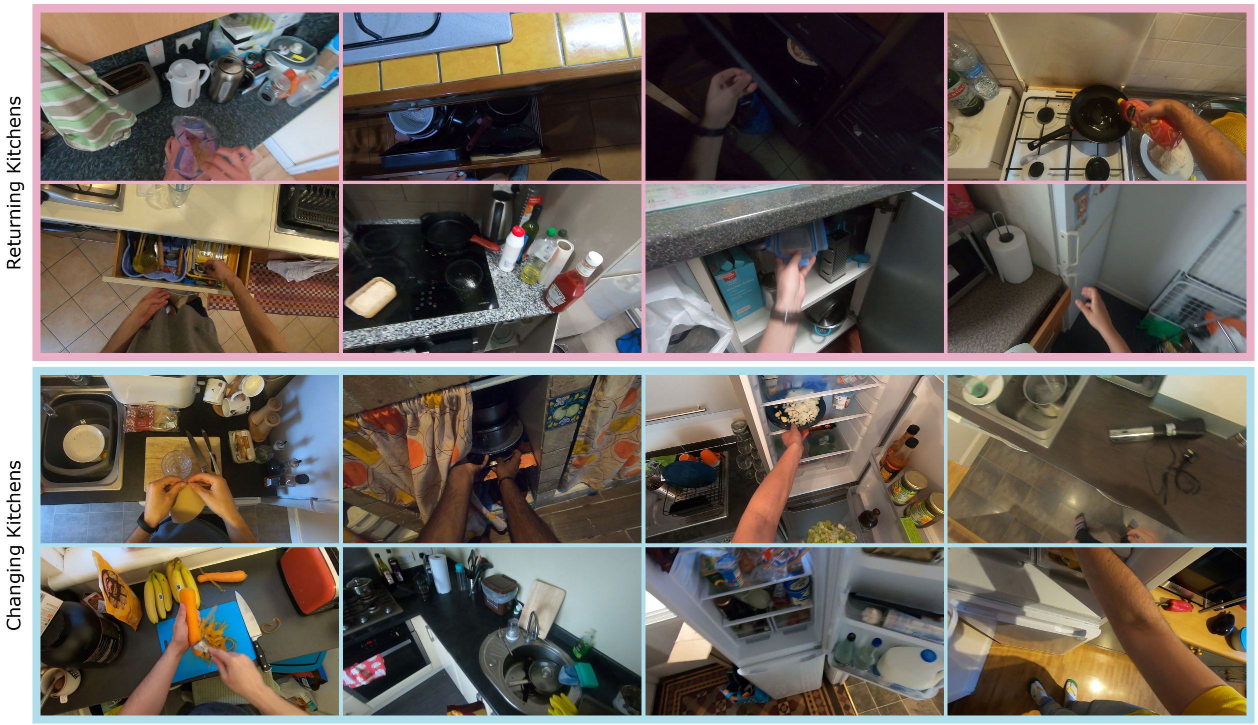


Figure 5.1: Frames from EPIC-KITCHENS-100 showcasing returning participants, who filmed in the same (top) or different locations (bottom) between recording periods.

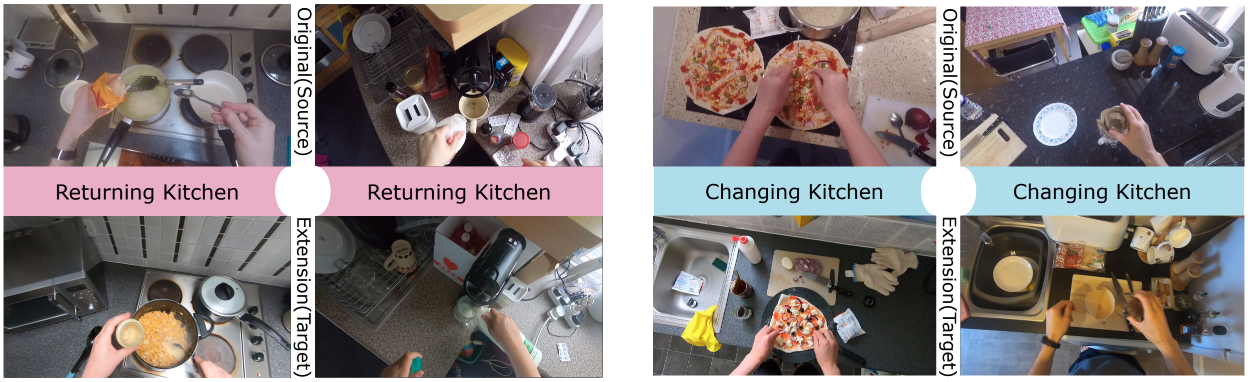


Figure 5.2: Frames from the original footage (Source) and the extension two-years later (Target), for participants who changed or stayed in the same location between recording periods.

newer than the camera used in EPIC-KITCHENS-55, with a built-in feature for HyperSmooth video stabilisation. Sample frames are shown in Fig. 5.1.

In EPIC-KITCHENS-100, there is a domain gap between the videos in the extension (Target) and the videos in the initial collection of footage (Source) due the following properties:

- Hardware and capturing. Extended footage uses a newer camera model with onboard video stabilisation.

- Locations. 8 subjects have moved home, resulting in changed surroundings.

- Long-term temporal offsets. EPIC-KITCHENS-100 is filmed 2 years after EPIC-KITCHENS-55. In the same environment, changes such as wear and tear, new objects and different object positions are observed. Participant behaviour can also change over time.

To encourage the development of methods which tackle the domain gap, Sec. 5.2 outlines the proposed benchmark on unsupervised domain adaptation for action recognition.

5.2 The Unsupervised Domain Adaptation Challenge

This section defines the proposed unsupervised domain adaptation challenge for EPIC-KITCHENS-100 (Sec. 5.2.1). A particular emphasis was given to designing training, validation and test splits suited for domain adaptation (Sec. 5.2.2). This is an open-set domain adaptation problem as there is not a complete overlap of action classes between the source and the target domains (Sec. 5.2.3). Finally, the proposed benchmark is compared to other domain adaptation benchmarks, in images and video (Sec. 5.2.4).

5.2.1 Definition

Each video in EPIC-KTICHENS-100 is associated with a noun, describing the main object of interaction, and a verb, describing the object manipulation. Verbs were manually clustered into verb classes, C_v, with synonyms, describing the same manipulation, assigned to the same class. Similarly, nouns were clustered into noun classes, C_n. Given a set of training videos, with an action label assigned to each, $a = (v \in C_v, n \in C_n)$, the task is to predict the action labels for a set of unseen videos. However, when the test set is collected in a later recording period to the training set, a domain gap may be present. Therefore, we propose the unsupervised domain adaptation challenge, which is defined next.

Unsupervised Domain Adaptation (UDA) utilises a labelled source domain and learns to adapt to an unlabelled target domain. Videos recorded in 2018 are used as the labelled source, and newly collected videos are used as the target. Both source and target videos are available during training, but the target videos are not annotated with action labels. The task is to predict the action labels of target video not seen during training. The difficulty of this challenge stems from the fact that the source and the the target domain come from distinct training distributions, due to the collection of videos two years later.

5.2.2 Splits

This challenge assesses models' ability to adapt to additional footage without labels. Thus, the following splits are defined; *Source*: labelled training data from 16 participants (collected in 2018) and *Target:* unlabelled footage from the same 16 participants collected in 2020. This ensures the gap in the domains is related to the capturing of the data 'two years later'.

For unsupervised domain adaptation, labels from the target domain should not be used for both training and validation. However, without a labelled validation set, it is difficult to select optimal hyper-parameters for any proposed domain adaptation solution. This can limit the ability

				No. Classes		
Split	Domain	No. Participants	No. Instances	Verb	Noun	Action
Train	Source	12	16115	86	219	1663
	Target	12	26115	87	227	2031
Val	Source	4	5002	69	147	821
	Target	4	7906	77	170	1089
Test	Source	12	4298	65	170	802
	Target	12	5909	80	161	937

Table 5.1: Training, validation and test splits for the unsupervised domain adaptation challenge.

to provide a fair comparison between different baselines.

The train, validation and test splits are shown in Table 5.1. 4 participants are reserved for validation and the remaining 12 participants are used for training and testing. This allows hyper-parameters to be selected using the validation participants, without accessing labelled target data from the training participants. Next, the train and test splits are discussed, followed by a detailed description of the validation procedure.

Train and Test Splits

Train and test splits are defined for each domain, and contain the same 12 participants. Half of these participants have changed kitchen between domains, while the other participants collected footage in the same kitchen.

Target videos are split into: *Target Train* and *Target Test*. The first are unlabelled videos used during domain adaptation, while the second are videos used for evaluation, as in the VisDA 2017

Challenge [88]. Similarly, source videos are split into: *Source Train* and *Source Test*. The first provides labelled videos used to learn the action recognition task, while the second is used to evaluate the performance of action recognition without the domain shift. The number of action instances per split are reported in 5.3. The challenge is evaluated by top-1 action accuracy on *Target Test*.

Validation Splits for Hyper-parameter Tuning

As the target domain is unlabelled, no labelled data is available for hyper-parameter tuning. Therefore, the training data is split to create *Source Val* and *Target Val*, with data collected by 4 of the 16 participants. Similar to the training splits, half of the participants are of returning kitchens and half are of changing kitchens.

For hyper-parameter tuning, models are trained on labelled data from *Source Val* and unlabelled from *Target Val*. The performance on *Target Val* can be used to asses the impact of different hyper-parameters. Note that action labels are available for *Target Val*, but *Target Train* is unlabelled. This ensures that no labelled target data is used from the participants in the train and test splits for validation.

To obtain the results for the challenge, a new model is trained on *Source Train* and unlabelled *Target Train*, using the hyper-parameters optimised from the validation split. This model is evaluated on *Target Test* to obtain results.

5.2.3 Open-set Adaptation due to Zero-Shot Actions

Due to the unscripted nature of the data collection, a negligible number of verb and noun classes in the target domain are not present in the source domain. Table 5.2 shows the number of target videos that do not contain a verb, noun or action label present in *Source Train*. The vast majority of verb or noun labels in *Target Train* are present in the *Source Train*. Only 0.4% and 3.0% of the

	No. Target Instances not in *Source Train*			No. Target Classes not in *Source Train*		
Split	Verbs	Nouns	Actions	Verbs	Nouns	Actions
Target Train	99 (0.4%)	783 (3.0%)	3566 (13.7%)	8	40	1073
Target Test	6 (0.1%)	178 (3.0%)	792 (13.4%)	4	20	353

Table 5.2: The number of target verb, noun and actions labels that are not present in Source Train. This showcases the open-set nature of the dataset, providing additional challenges for domain adaptation.

verb and noun labels in *Target Train* are absent in *Source Train*, respectively. These have not been removed from the dataset. However, 13.7% actions (exact verb-noun combinations) did not exist in *Source Train*, these are referred to as the zero-shot actions. Note that it is still possible to predict these actions as both verbs and nouns were present in the source domain. In addition, the zero-shot actions belong to 1073 action classes or 63% of all action classes in *Target Train*.

The open-set nature of the dataset brings additional challenges to the dataset. How methods align the source and the target domain, given many target videos depict actions that are not present in the source, remains an interesting research direction.

5.2.4 Related Datasets

This section provides a brief comparison of the proposed benchmark with other unsupervised domain adaptation (UDA) benchmarks. UDA benchmarks have traditionally used images [60, 78, 88, 234], with recent attempts to use video [1, 2, 235] adapting across public datasets (*e.g.*UCF to Olympics). EPIC-KITCHENS-100 is the first to propose a within-dataset domain adaptation challenge in video. Video-based UDA raises additional challenges, such as aligning temporal information across domains [2], attending to relevant transferable frames [1], and avoiding non-informative background frames [37].

Table 5.3 shows that EPIC-KITCHENS-100 provides several advantages over other video-based benchmarks, such as the largest number of instances, classes, and the fact it is multi-

	Dataset	**Train**		**Test**	**Classes**	**M**	**AAL**	**Year**	**Real/Syn**
Image	Office [60]	4110		N/A	31	1	N/A	2010	Real
	Office↔ Home [78]	15500		N/A	65	1	N/A	2017	Real
	VisDA-C [88]	280157		N/A	12	1	N/A	2017	Real/Syn
	DomainNet [234]	363534		37706	345	1	N/A	2019	Real/Syn
		Source	**Target**	**Test***					
Video	UCF↔ HMDB (small) [36]	482	350	150	5	1	4.7 ± 2.5	2018	Real
	UCF↔ Olympic [2]	601	250	54	6	1	6.6 ± 4.5	2018	Real
	UCF↔ HMDB (full) [1]	1438	840	360	12	1	4.0 ± 5.8	2019	Real
	IEMOCAP→ AFEW [235]	6611	795	N/A	4	2	N/A	2018	Real
	Kinetics↔ Gameplay [1]	43378	2625	749	30	1	N/A	2019	Real/Syn
	EPIC-KITCHENS-100	16115	26115	5909	3369	3	2.8 ± 5.2	2020	Real

Table 5.3: Comparison of domain adaptation classification datasets. M: Modalities. AAL: Average Action Length. *: Note that Test* refers to *Target Test*

modal. In addition, all the video-based UDA datasets focus on closed-set adaptation, whereas EPIC-KITCHENS-100 has additional challenges due the partial overlap of action labels between domains.

5.3 Evaluation

This section evaluates several baselines, defined in Sec. 5.3.1, on the UDA challenge. Implementation details are provided in Sec. 5.3.2, and the results are presented in Sec. 5.3.3, for RGB, Flow and Audio modalities.

5.3.1 Baselines

The 'Source-Only' baseline provides a lower-bound of domain adaptation performance, where labelled source data is used for training and no adaptation to target data is attempted. Two possible upper-bounds are used, 'Target-Only' and 'Source+Target', where *labelled* target data is used. 'Target-Only' only uses target data, whereas 'Source+Target' uses both source and target data. Neither of these are UDA methods, but offer an insight into the domain gap.

Temporal Attentive Alignment (TA3N) [1] is used for the domain adaptation baseline. This trains a Temporal Relation Module [20] with adversarial domain alignment and a domain attention mechanism. TA3N was reviewed in Chapter 3, Sec. 3.3.

5.3.2 Implementation and Training Details

The TBN feature extractor [23], is trained on the union of *Source Train* and *Source Val*. We make these features publicly available. The available code from Chen *et al*. [1] is used to train and evaluate all baselines. The code is modified to consider multi-modal input, by concatenating the features from all modalities as input. This automatically increased the number of parameters in the first fully connected layer.

The performance of TA3N is improved by initialising the domain discriminators before the gradients are reversed and back-propagated. In our implementation, the domain discriminators' hyper-parameters, which weight the gradient back-propagated by the gradient reversal layer, are annealed similar to that in [4]:

$$\eta = \frac{2}{1 + exp(-p)} - 1 \tag{5.1}$$

where p is the training progress that linearly increases from 0 to 1. The domain discriminator hyperparameters are annealed up to the value specified in TA3N, i.e. $\lambda^s = 0.75\eta$, $\lambda^r = 0.5\eta$ and $\lambda^t = 0.75\eta$. TA3N applies categorical entropy minimisation to the output of the action classifier, for target videos at input. The weighting of the entropy loss is set to $\gamma = 0.003$. Models are trained for 30 epochs at a learning rate of $3e^{-3}$, reduced by a factor of 10 at epochs 10 and 20.

5.3.3 Results

Table 5.4 reports the results for the baselines on *Target Test*. These show significant performance improvements when using multi-modal data compared to RGB (11.7% increase in top-1 verb accuracy and 4.1% in nouns). For individual modalities, Flow achieves highest verb performance and RGB achieves highest noun performance. The domain gap is evident when comparing the lower and upper bounds. ‘Source-only’ has consistently worse performance, for all input modalities, compared to models trained with target labelled data. TA3N is able to partially decrease this gap, providing a 2.5% improvement in verb accuracy and 2.4% in nouns. This dataset provides a UDA benchmark that is less saturated than existing datasets [2].

Figure 5.3 visualises the feature space of a model trained on all modalities, showing limited overlap between source and target. TA3N aligns the features demonstrating the capability of UDA. Figure 5.4 visualises the feature space for models trained on individual modalities. TA3N shows improved alignment for RGB. For all modalities, target videos are not well clustered into action classes, compared to that of source videos, in both the source-only and adapted models. Future research could improve local alignment of action classes in source and target, in addition to improving the clustering of target videos.

Table 5.5 reports the accuracy on *Target Test* separately for instances recorded by participants who moved home, or stayed in the same location. The performance increase of TA3N over ‘source-only’ is greatest for participants who moved home. However, the performance of TA3N is far from the upper-bound performance, regardless of whether the participant moved or stayed in the same location between recording periods. This indicates that other factors have a larger impact on the domain shift, *i.e.* the different camera used and temporal progression of two-years between

		Target Accuracy					
		Top-1 (%)			Top-5 (%)		
Modality	Baseline	Verb	Noun	Act.	Verb	Noun	Act.
RGB	Source-Only	32.8	21.2	10.7	72.6	43.9	22.6
	TA3N [1]	32.1	21.6	11.1	71.7	44.1	22.5
	Target-Only	39.7	32.3	18.3	80.8	56.2	34.0
	Source+Target	41.1	33.0	18.8	80.4	58.5	35.2
Flow	Source-Only	42.8	19.2	12.7	74.5	38.5	23.8
	TA3N [1]	43.2	20.1	12.8	74.5	41.2	25.0
	Target-Only	53.8	26.7	20.2	84.2	49.1	34.5
	Source+Target	52.4	27.3	19.8	82.4	50.3	35.4
Audio	Source-Only	31.4	12.8	8.5	64.8	28.4	16.0
	TA3N [1]	32.0	13.3	8.9	66.0	29.1	16.5
	Target-Only	41.7	19.1	13.4	77.2	39.6	23.6
	Source+Target	41.8	19.8	13.8	77.1	40.6	24.3
RGB+Flow +Audio	Source-Only	44.4	25.3	16.8	69.7	48.4	29.1
	TA3N* [1]	46.9	27.7	19.0	72.7	50.7	30.5
	Target-Only	59.1	40.3	30.4	85.0	65.0	47.8
	Source+Target	59.4	41.9	31.3	85.3	66.6	49.2

Table 5.4: Unsupervised domain adaptation results on *Target Test* with lower (source-only) and upper (target-only and source+target) bounds.

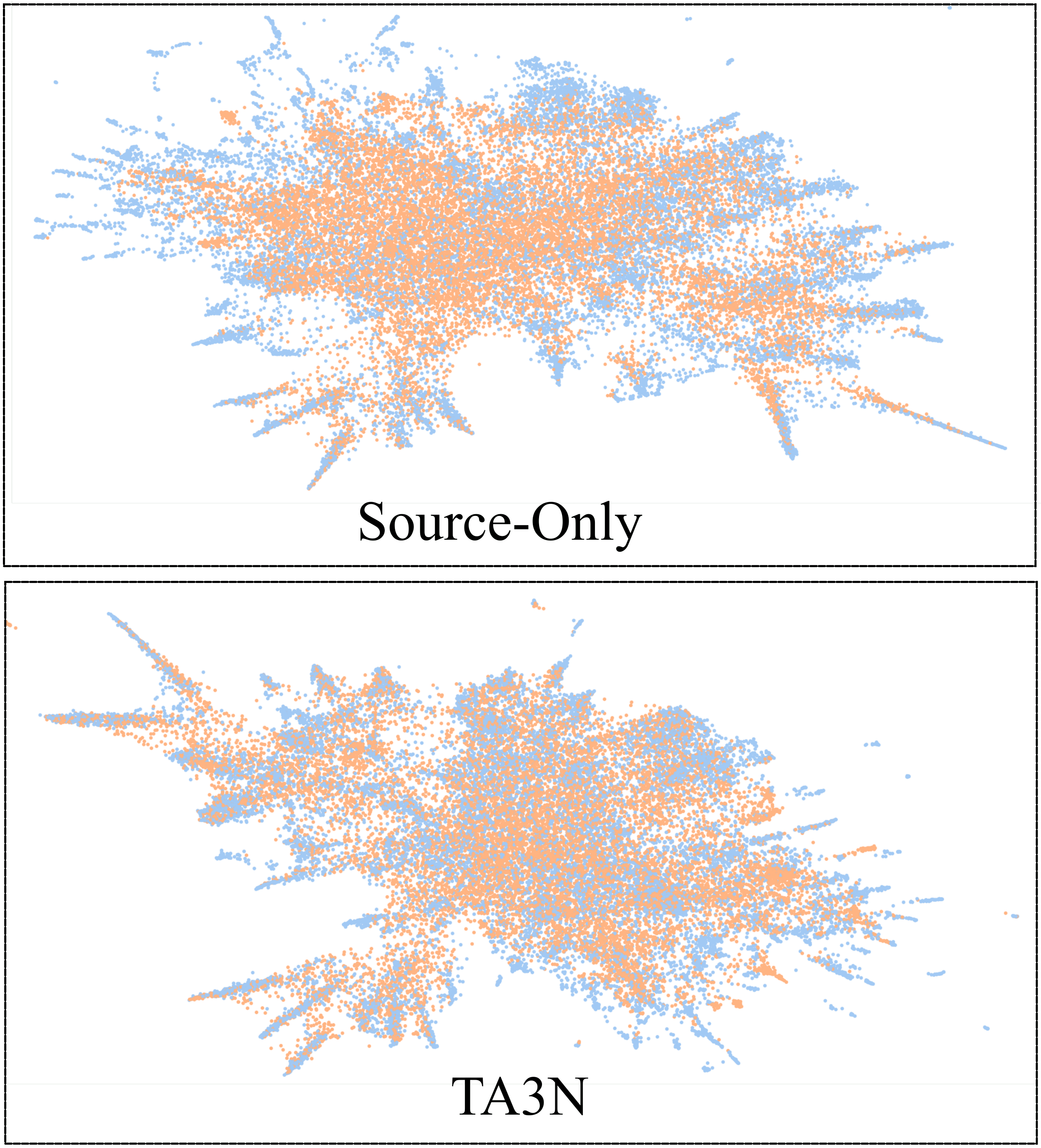
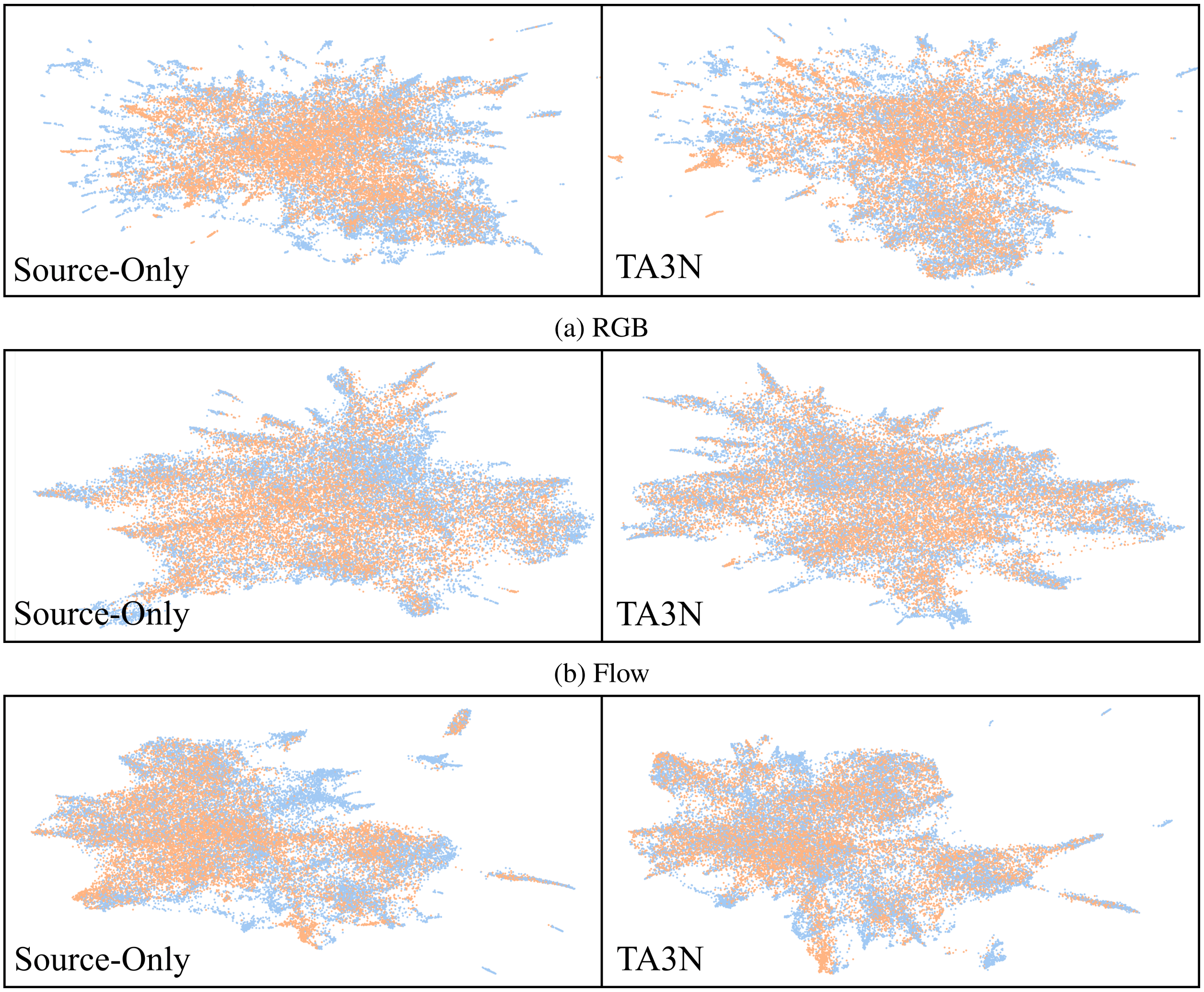


Figure 5.3: UMAP [24] of feature spaces for models train on all modalities. The UDA baseline shows better alignment. Source videos are shown as blue, Target are shown are orange.



(a) RGB

(b) Flow

(c) Audio

Figure 5.4: UMAP [24] of feature spaces for models trained on individual modalities. Source videos are shown as blue, Target are shown are orange.

	Target Top-1 Accuracy (%)					
	Returning Kitchen			Changing Kitchen		
Baseline	Verb	Noun	Act.	Verb	Noun	Act.
Source-Only	44.0	25.5	16.7	45.2	24.7	17.0
TA3N [1]	45.9	27.6	18.6 ▲+1.9	49.2	27.9	19.8 ▲+2.8
Target-Only	57.9	40.8	29.8	62.0	39.2	31.8
Source+Target	58.0	41.7	30.6	62.6	42.4	33.0

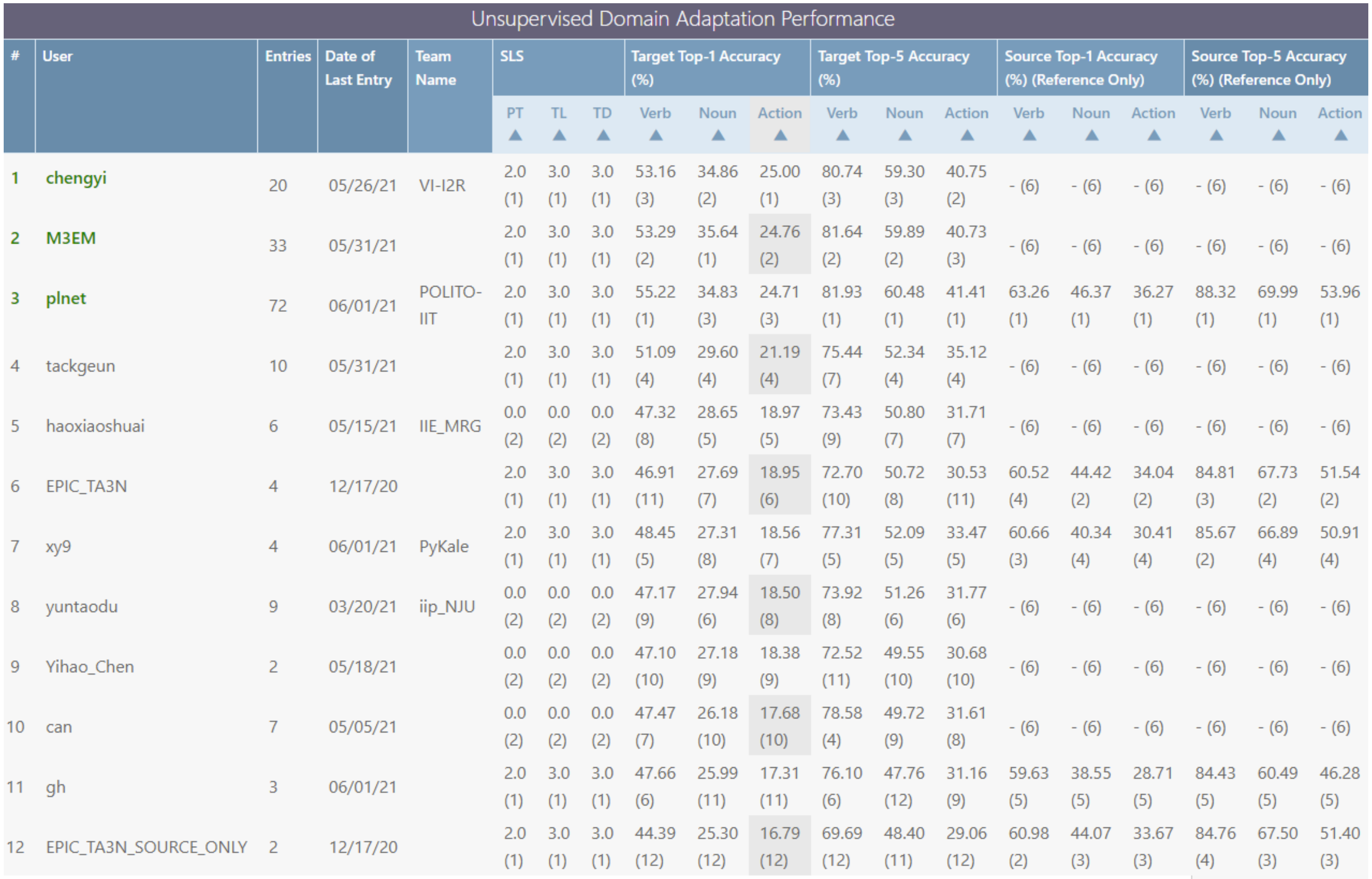
Table 5.5: Accuracy on *Target Test* broken down by participants who recorded in the same kitchen, or changed kitchen across domains. Models were trained with all modalities as input (RGB, Flow and Audio).

recordings periods.

5.4 Challenge Submissions

A public competition for the Unsupervised Domain Adaptation challenge was hosted on CodaLab. In total, there were 166 submissions, across 10 different teams. 4 teams submitted technical reports which are summarised later in this section.

Figure 5.5 shows the results achieved from the participants. All submissions outperformed the baseline model trained only on the source domain (EPIC_TA3N_SOURCE_ONLY) and three submissions outperformed the UDA baseline (EPIC_TA3N) by at least 5.76% action accuracy. The majority of submissions did not submit predictions for *Source Test*, which were optional for the submission on CodaLab. This would have provided additional insights into how much each submission improves action recognition in general compared to overcoming the domain gap. We



Unsupervised Domain Adaptation Performance

#	User	Entries	Date of Last Entry	Team Name	SLS			Target Top-1 Accuracy (%)			Target Top-5 Accuracy (%)			Source Top-1 Accuracy (%) (Reference Only)			Source Top-5 Accuracy (%) (Reference Only)		
					PT ▲	TL ▲	TD ▲	Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲	Verb ▲	Noun ▲	Action ▲
1	chengyi	20	05/26/21	VI-I2R	2.0 (1)	3.0 (1)	3.0 (1)	53.16 (3)	34.86 (2)	25.00 (1)	80.74 (3)	59.30 (3)	40.75 (2)	- (6)	- (6)	- (6)	- (6)	- (6)	- (6)
2	M3EM	33	05/31/21		2.0 (1)	3.0 (1)	3.0 (1)	53.29 (2)	35.64 (1)	24.76 (2)	81.64 (2)	59.89 (2)	40.73 (3)	- (6)	- (6)	- (6)	- (6)	- (6)	- (6)
3	plnet	72	06/01/21	POLITO-IIT	2.0 (1)	3.0 (1)	3.0 (1)	55.22 (1)	34.83 (3)	24.71 (3)	81.93 (1)	60.48 (1)	41.41 (1)	63.26 (1)	46.37 (1)	36.27 (1)	88.32 (1)	69.99 (1)	53.96 (1)
4	tackgeun	10	05/31/21		2.0 (1)	3.0 (1)	3.0 (1)	51.09 (4)	29.60 (4)	21.19 (4)	75.44 (7)	52.34 (4)	35.12 (4)	- (6)	- (6)	- (6)	- (6)	- (6)	- (6)
5	haoxiaoshuai	6	05/15/21	IIE_MRG	0.0 (2)	0.0 (2)	0.0 (2)	47.32 (8)	28.65 (5)	18.97 (5)	73.43 (9)	50.80 (7)	31.71 (7)	- (6)	- (6)	- (6)	- (6)	- (6)	- (6)
6	EPIC_TA3N	4	12/17/20		2.0 (1)	3.0 (1)	3.0 (1)	46.91 (11)	27.69 (7)	18.95 (6)	72.70 (10)	50.72 (8)	30.53 (11)	60.52 (4)	44.42 (2)	34.04 (2)	84.81 (3)	67.73 (2)	51.54 (2)
7	xy9	4	06/01/21	PyKale	2.0 (1)	3.0 (1)	3.0 (1)	48.45 (5)	27.31 (8)	18.56 (7)	77.31 (5)	52.09 (5)	33.47 (5)	60.66 (3)	40.34 (4)	30.41 (4)	85.67 (2)	66.89 (4)	50.91 (4)
8	yuntaodu	9	03/20/21	iip_NJU	0.0 (2)	0.0 (2)	0.0 (2)	47.17 (9)	27.94 (6)	18.50 (8)	73.92 (8)	51.26 (6)	31.77 (6)	- (6)	- (6)	- (6)	- (6)	- (6)	- (6)
9	Yihao_Chen	2	05/18/21		0.0 (2)	0.0 (2)	0.0 (2)	47.10 (10)	27.18 (9)	18.38 (9)	72.52 (11)	49.55 (10)	30.68 (10)	- (6)	- (6)	- (6)	- (6)	- (6)	- (6)
10	can	7	05/05/21		0.0 (2)	0.0 (2)	0.0 (2)	47.47 (7)	26.18 (10)	17.68 (10)	78.58 (4)	49.72 (9)	31.61 (8)	- (6)	- (6)	- (6)	- (6)	- (6)	- (6)
11	gh	3	06/01/21		2.0 (1)	3.0 (1)	3.0 (1)	47.66 (6)	25.99 (11)	17.31 (11)	76.10 (6)	47.76 (12)	31.16 (9)	59.63 (5)	38.55 (5)	28.71 (5)	84.43 (5)	60.49 (5)	46.28 (5)
12	EPIC_TA3N_SOURCE_ONLY	2	12/17/20		2.0 (1)	3.0 (1)	3.0 (1)	44.39 (12)	25.30 (12)	16.79 (12)	69.69 (12)	48.40 (11)	29.06 (12)	60.98 (2)	44.07 (3)	33.67 (3)	84.76 (4)	67.50 (3)	51.40 (3)

Figure 5.5: Results on the Unsupervised Domain Adaptation challenge

encourage next year's submissions to consider providing the *Source Test* scores.

5.4.1 Technical Reports

The technical reports for the Unsupervised Domain Adaptation challenge, in order of their overall rank on the public leaderboard, are given in this section. Most solutions exploited multiple modalities for domain adaptation, and the best performing solutions used additional backbone architectures compared to the baselines which used TBN.

VI-I2R (Rank 1) This method is based on the Temporal Attentive Adversarial Adaptation Network (TA3N) [1], augmented with hand-centric features. To locate hands, a Multi-level Entropy Attention Alignment (MEAA) [236] is used to train the detector. Additional hand-labelled hand bounding boxes are used, in comparison to other methods which use pre-trained hand detectors.

M3EM (Rank 2) The main idea is that early fusion between modalities can help improve features across modalities. This is handled by a Multi-Modal Mutual Enhancement Module (M3EM). This contains a Semantic Mutual Refinement (SMR) module which finds the most transferable features, and a Cross Modality Consensus (CMC) module which finds the most transferable regions. The best result uses an ensemble which also includes object features, and is guided by a hand feature extractor.

PoliTO-IIT (Rank 3) The approach is based on an audio-visual Relative Norm Alignment Network (RNA-Net) [44] with added flow, applied to source and target separately. Both TA3N and RNA are used for adaptation. Additional losses are incorporated—Temporal Hard Norm Alignment (T-HNA) and Min-Entropy consistency (MEC) to encourage consistency between different modalities.

PyKale (Rank 5) The approach uses transformer encoding on top of the input features to give whole video embeddings. Adversarial domain classification is used after fusing results from each modality, treating each domain separately. Results are not significantly better than TA3N, which shows that just using a standard non-video domain adaptation technique is insufficient.

These works showed the benefit of domain adaptation through the fusion of multiple modalities (M3EM), in addition to aligning the feature norms of individual modalities (PoliTO). The use of local visual ques, such as object features (M3EM) and hand-centric features (VI-I2R), have also been shown to benefit domain adaptation. However, some issues were not addressed by these works. The partial-overlap of actions classes is still an open challenge, in addition to the long-tailed distribution of classes in source and target. All works focused on aligning the marginal distributions across domains, which may not achieve discriminative clustering of target videos. Local alignment of classes through pseudo-labelling approaches (see Sec 2.3.6) may improve domain adaptation.

5.5 Conclusion

This chapter showcased the domain gap between the videos captured two-years apart in the EPIC-KITCHENS-100 dataset, that occurs due to subtle changes in the environment and recording conditions. The proposed Unsupervised Domain Adaptation challenge is the first to tackle this domain gap for action recognition, and provides one of the largest domain adaptation benchmarks for video. While action recognition performance on Target can be improved by recent video domain adaptation solutions [1], accuracy is far from the upper-bound when models are trained on labelled Target data. The challenge has led to the development of several novel multi-modal domain adaptation solutions, proposed by the research community, with scope for future research to further improve domain adaptation performance.

Chapter 6

Domain Adaptation for Text-to-Video Retrieval

Using natural language queries to search for video sequences is the most intuitive interface for video search engines. Example applications include finding short and precise action sequences in long instructional videos, *e.g.* for educational purposes. Text-to-video retrieval is typically approached by training a joint video-text embedding following a learning-to-rank objective. Thanks to this, retrieval comprises of computing distances between representations of textual queries and representations of videos in a gallery. Building a reliable joint embedding space requires large amounts of accurately annotated video-caption pairs [22, 152, 204, 205, 206, 207, 208, 210], where captions come from manual annotations, query logs or transcribed audio. These works assume the training set allows for learning a robust representation that generalises to the test/target videos. However, the appearance of certain actions can change due to new environments or recording equipment. This introduces a visual domain gap between the captioned video sequences and the gallery set to retrieve relevant videos from, rendering the trained joint embedding space sub-optimal.

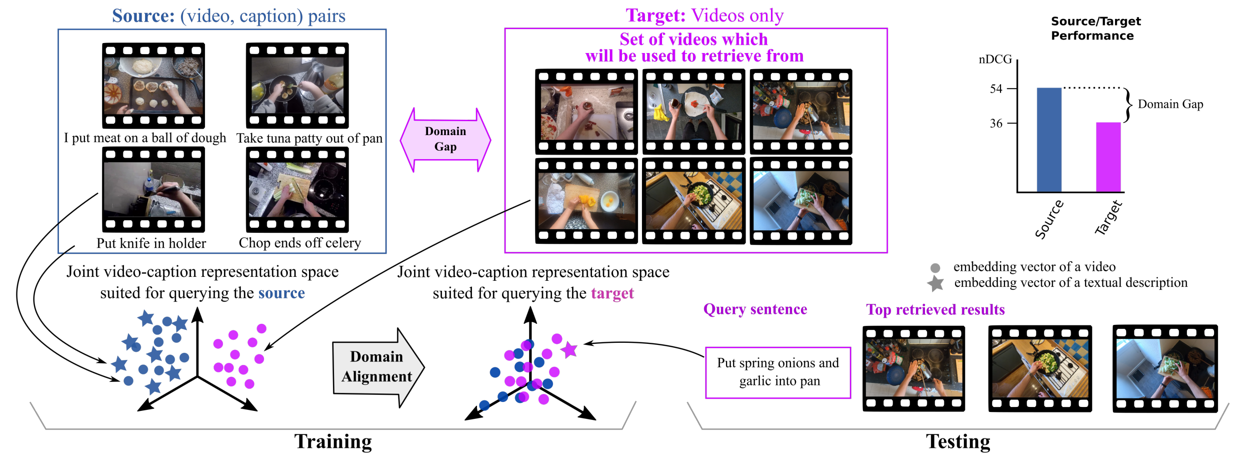

Source: (video, caption) pairs
I put meat on a ball of dough
Take tuna patty out of pan
Put knife in holder
Chop ends off celery
Domain Gap
Target: Videos only
Set of videos which will be used to retrieve from
Source/Target Performance
nDCG
54
36
Domain Gap
Source
Target
Joint video-caption representation space suited for querying the source
Joint video-caption representation space suited for querying the target
Domain Alignment
● embedding vector of a video
★ embedding vector of a textual description
Query sentence
Top retrieved results
Put spring onions and garlic into pan
Training
Testing

Figure 6.1: Given video-text pairs (denoted by circles and stars) in source (blue), and video-only target set (purple), the proposed alignment method reduces the domain gap between the source videos and the target videos using pseudo-labels and cross-domain ranking. The learnt and aligned space can then be used for retrieving a ranked list of target videos using previously unseen text queries.

This chapter addresses the challenge of a visual domain gap in the context of text-to-video action retrieval, assuming that no captions are available for the target gallery. Experiments are conducted on the EPIC-KITCHENS-100 dataset [10], composed of two distinct sets of videos featuring fine-grained actions. Chapter 5 showed a significant domain shift which alters performance when tested for action recognition. Instead, this work provides the first attempt to use this dataset for domain alignment in action retrieval.

Experiments show that the marginal alignment of the distribution over videos [57, 73], independent of the text, is insufficient to fix the domain gap. Instead, this chapter proposes to leverage pseudo-labels for the target videos in a unified learning-to-rank approach which effectively combines *captioned source* video sequences and *uncaptioned target* videos (see illustration in Figure 6.1). Given the domain gap, pseudo-labelling is noisy. Therefore, this chapter proposes a robust confidence measure to determine the reliability of pseudo-labels, combined with a sampling strategy that ensures all actions are aligned, rather than only the most confident or common actions.

For better adaptation, disentangled embeddings are learnt, where each embedding specialises in a different parts-of-speech (*i.e.* verb and nouns) within the caption.

To summarise, the contributions of this chapter are threefold. First, the problem of Unsupervised Domain Adaptation (UDA) for text-to-video retrieval is introduced. In this setup, training only has access to i) a source domain composed of video-caption pairs and to ii) videos (without captions) from the target gallery set. Second, a cross-domain learning-to-rank strategy is proposed which mixes samples from the source and the target domains thanks to an iterative pseudo-labelling and robust sampling strategy. Third, extensive experiments are conducted on the EPIC-KITCHENS-100 dataset which validate our approach. Our method outperforms state-of-the-art UDA approaches [130, 133, 134] which were adapted for action retrieval.

6.1 Method

This work tackles the problem of text-to-video retrieval, where the goal is to retrieve relevant videos from a *target* gallery of videos V^t, given any text query w^t. To achieve this, a *source* gallery of paired videos, V^s, with their captions, W^s, is used to learn a cross-modal embedding. However, a domain shift is known to exist between V^s and V^t. The aim is to leverage the paired *source* set, and the single-modality *target* set (*i.e.*videos only), for training a text-to-video retrieval model which will be applied to the *target* set.

An overview of the proposed method is provided in Fig. 6.2. This section presents a text-to-video retrieval system which performs well when aligned video-caption pairs are available for training, *e.g.* as in our source domain (Sec. 6.1.1). Our work builds on this baseline with a domain adaptation strategy (Sec. 6.1.2), which learns a cross-domain embedding to align source and target videos that depict similar actions. To overcome the absence of the text modality in the target domain, a robust pseudo-labelling strategy is applied to the target gallery (Sec. 6.1.4). This relies on prototype-based confidence estimation to reduce noise in the labelling process. For better adapta-

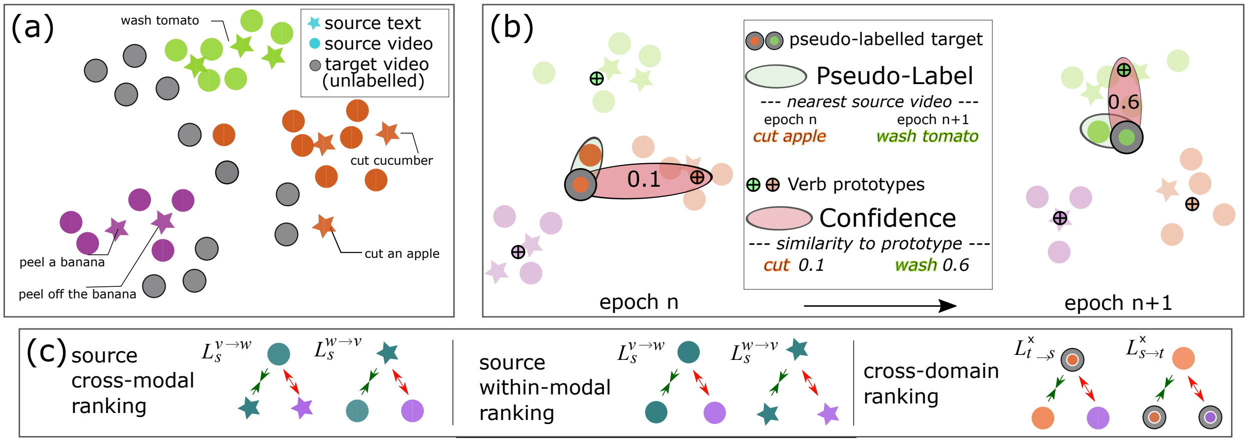


Figure 6.2: Proposed alignment strategy for the disentangled embedding space of verbs. (a) Domain gap between target videos (grey) and video-caption source pairs (colored according to their verb - wash/green, cut/orange, peel/purple) (b) Target videos inherit pseudo-labels from their nearest source video, with a confidence defined by the verb prototypes. Over epochs, pseudo-labelling is updated. We show one video pseudo-labelled as cut in epoch n, and wash in n+1 with increased confidence. (c) Cross-modal losses (L_s) are jointly trained with cross-domain losses $L^{\times}_{s\to t}$, $L^{\times}_{t\to s}$. Relevant (green arrow) and irrelevant (red arrow) relations are used to train the embedding space with only the most confidently pseudo-labelled target videos.

tion, an embedding is learnt for each part-of-speech in the text caption, and alignment losses are applied to each (Sec. 6.1.5).

6.1.1 Training for Text-to-video Retrieval with Captioned Videos

Given a set of videos, V^s, and their corresponding captions, W^s, one can learn a joint video-text embedding space, following a learning-to-rank approach with the hinge-based triplet ranking loss. Two embedding functions $f : V^s \to \Omega$ and $g : W^s \to \Omega$ which respectively produce representations for videos and text in the embedding space Ω are learned. The triplet ranking loss is defined in Eq. 6.1, where a is the anchor, b^+ and b^- are relevant and irrelevant samples for that anchor, γ is a constant margin, $d(\cdot,\cdot)$ is the distance function and $h \in \{f, g\}$ is the appropriate embedding function, f or g, depending on the modality. Therefore:

$$H(a, b^+, b^-) = max(\gamma + d\big(h(a), h(b^+)\big) - d\big(h(a), h(b^-)\big), 0) \tag{6.1}$$

Classical text-to-video retrieval approaches [204, 205, 207, 208] sample triplets within video-caption pairs from (V^s, W^s) and optimize cross-modal losses Eq 6.2, where $v_i, v_k \in V^S$ and $w_i, w_j, \in W^S$ and θ are the parameters of the embedding functions h.

$$L(\theta) = \sum H(v_i, w_i, w_j) + \sum H(w_i, v_i, v_k), \tag{6.2}$$

While these approaches typically already perform well, videos/captions of similar actions are not explicitly encouraged to be close in the embedding space. Inspired by previous approaches which learn a cross-modal embedding [22, 237], we use both cross-modal and within-modal losses between sets of relevant items. When the anchor is a video, combining the cross-modal and within-modal losses produces:

$$\begin{gathered} L_s^{v \to w}(\theta) = \sum H(v_i, w_j, w_k) + \sum H(v_i, v_l, v_m) \\ \text{with } w_j \in W^s_{i+}, w_k \in W^s_{i-}, v_l \in V^s_{i+}, v_m \in V^s_{i-} \end{gathered} \tag{6.3}$$

where V^s_{i+} are all videos relevant to the anchor video v_i and V^s_{i-} are irrelevant ones. Similarly, W^s_{i+}/W^s_{i-} is the set of all captions relevant/irrelevant to the anchor video v_i.

In addition, cross-modal and within-modal losses are defined where the anchor is a text caption:

$$\begin{gathered} L_s^{w \to v}(\theta) = \sum H(w_i, v_j, v_k) + \sum H(w_i, w_l, w_m) \\ \text{with } v_j \in V^s_{i+}, v_k \in V^s_{i-}, w_l \in W^s_{i+}, w_m \in W^s_{i-} \end{gathered} \tag{6.4}$$

where V^s_{i+}/V^s_{i-} are all videos relevant/irrelevant to the anchor text w_i, and W^s_{i+}/W^s_{i-} is the set of all captions relevant/irrelevant to the anchor text w_i.

The training for the cross-modal embedding is then defined as:

$$L_s = L_s^{v \to w} + L_s^{w \to v} \tag{6.5}$$

The following section shows how an embedding space trained on the source domain $S = (V^s, W^s)$ can be adapted for the target video gallery V^t.

6.1.2 Cross-domain Cross-modal Embedding

Reducing the domain shift by aligning marginal distributions of the source and the target video sequences [4, 73] often leads to sub-optimal solutions, as it does not consider the alignment of local distributions between the two galleries. This work proposes a cross-domain cross-modal approach instead, which aligns videos in the source and the target domain that depict similar actions. To overcome the lack of captions in the target gallery V^t we instead propose using pseudo-labels to model cross-modal relevance.

For simplicity, we refer to the source video $v_i^s \in V^s$ as s_i, and the target video $v_i^t \in V^t$ as t_i. We define the source-to-target and target-to-source cross-domain ranking objectives as:

$$\begin{aligned} L^{\times}_{s \to t}(\theta) &= \sum H(s_i, t_j, t_k)|\ t_j \in V^t_{s_i+}, t_k \in V^t_{s_i-} \\ L^{\times}_{t \to s}(\theta) &= \sum H(t_i, s_j, s_k)|\ s_j \in V^s_{ti+}, s_k \in V^s_{ti-} \end{aligned} \tag{6.6}$$

where $V^t_{s_i+}/V^t_{s_i-}$ are relevant/irrelevant sets of target videos for the source video s_i and V^s_{ti+}/V^s_{ti-} are the relevant/irrelevant sets of source videos for the target video t_i. These relevant sets are defined by the pseudo-labelling strategy described in Sec. 6.1.4.

We then train the source embedding objectives (see Section 6.1.1) along with the cross-domain ranking objectives:

$$L = L_s + \lambda_1 L^{\times}_{s \to t} + \lambda_2 L^{\times}_{t \to s} \tag{6.7}$$

where λ_1 and λ_2 are weights to balance the corresponding cross-domain losses.

Importantly, note that this is an iterative process during training. As the joint embedding space is trained, target videos are pseudo-labelled, and are used to generate triplets for cross-domain ranking. Target videos can potentially be assigned to different pseudo-labels at each iteration. For robust learning, we adjust the assignments to the relevance sets at the end of each training epoch.

6.1.3 Defining Relevancy

Most benchmarks in text-to-video retrieval consider each video in the gallery to be relevant to a single caption [150, 151, 152] . However, there may be multiple captions in the dataset that could describe the action depicted in the video equally well. In fact, many videos in EPIC-KITCHENS-100 [10] have identical captions. This work considers a video relevant to a set of captions, in line with recent works [215, 238, 239]. Finding these sets of relevant videos/captions requires additional knowledge, such as visual clustering [224], attributes [240], or semantic similarity between captions [215, 238, 239].

In our experiments, we use the verb and noun classes within EPIC-KITCHENS [10] to define the relevancy between videos and captions. The verb and noun classes were formed by grouping synonyms available within the dataset. Therefore, two captions are relevant to each other if both are assigned to the same verb class and the same noun class. Similarly, the videos attached to the captions will also be relevant to each other. This enables relevancy sets to be defined for the source domain losses in Eq. 6.5.

Cross-domain losses require relevancy sets between source and target videos. However, in the absence of the text modality in the target domain, relevancy between source and target videos cannot be determined using the same strategy. Therefore, we utilise a pseudo-labelling method to determine these relevancy sets, which is outlined next.

6.1.4 Pseudo-labelling and Prototype-based Sampling

In order to sample triplets for the cross-domain losses in Eq. 6.6, we need pairs of relevant videos between the source and the target sets. Therefore, we propose: i) a pseudo-labelling strategy to determine relevant video pairs across domains, and ii) an associated prototype-based confidence measure to select which target videos to use for training.

In standard DA, which often assumes a single modality and a classification task, typical approaches use the softmax scores of the source classifier to produce pseudo-labels [104, 126], or label propagation on the nearest neighbour graph built with the joint source and target sets [124, 241]. Taking inspiration from these strategies, and replacing *pseudo-labelling* with the process of assigning to each target video a relevant set of source videos, we proceed as follows. As in classical label propagation, the target video inherits the relevance property of the closest source video, *i.e.* a relevant/irrelevant query to the nearest source video is considered relevant/irrelevant to the target video. Formally, if $s_{\hat{n}}$ is the nearest source video to the target video, t_i, *i.e.* $\hat{n} = \arg\min_n d(f(t_i), f(s_n))$, we assign:

$$V^s_{t_i+} = \{V^s_{s_{\hat{n}}+} \cup \{s_{\hat{n}}\}\}; V^s_{t_i-} = V^s_{s_{\hat{n}-}} \tag{6.8}$$

Similarly, the target video is considered relevant to each source video in its relevant set. However, this process which associates videos across the source and the target domains might be error prone, and relying on too many erroneous relevance pairs would degrade the model. On the other hand, trusting too few would not be sufficient to align the domains. Consequently, to handle such a trade-off, we propose to robustly measure the confidence of the labelling process and to use these confidence scores to select target videos.

Prototype Confidence (PC). We propose to calculate confidence scores based on the distance between the target videos and the *source prototypes* of their closest source video $s_{\hat{n}}$ in the embedding

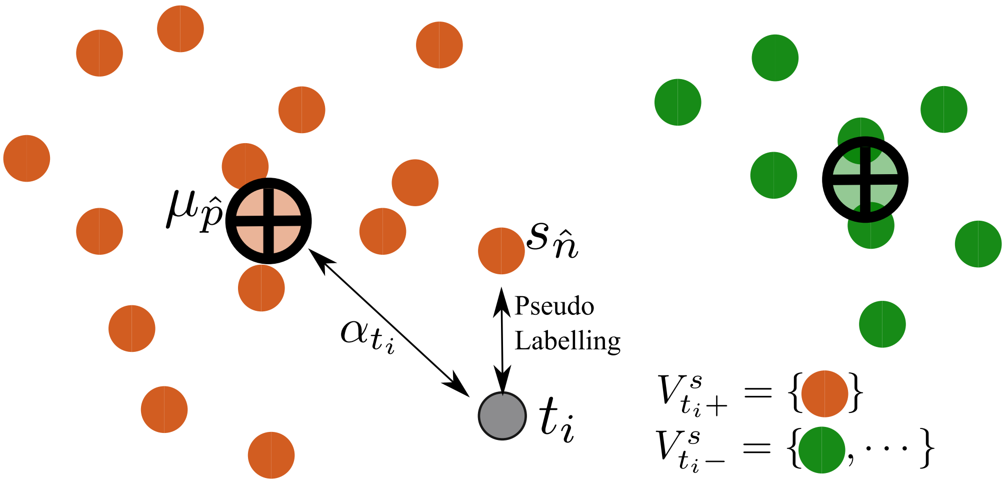


Figure 6.3: Pseudo-Labelling with Prototype Confidence (PC). A target video, t_i, is assigned a relevant set of source videos (all points in orange), from the nearest source video, $s_{\hat{n}}$. The confidence of the pseudo-labelling, α_{t_i} is determined by the distance to the source prototype $\mu_{\hat{p}}$.

space, as shown in Fig. 6.3. A source *prototype* is defined as the barycentre of the sets of relevant videos as described in Sec. 6.1.3. All videos within one set $V^s_{s_i+}$ are relevant to each other and share the same relevance properties, *e.g.* they show the same action. We calculate the prototype for this set V^s_p by:

$$\mu_p = \sum_{s_i \in V^s_p} f(s_i) \tag{6.9}$$

Let $\mu_{\hat{p}}$ be the prototype corresponding to the source video $s_{\hat{n}}$, *i.e.* $s_{\hat{n}} \in V^s_{\hat{p}}$, and therefore $V^s_{s_{\hat{n}}+} = V^s_{\hat{p}}$. Then, the confidence measure for pseudo-labelling the target video, t_i, is calculated as a function of the distance to the prototype $\mu_{\hat{p}}$, more precisely:

$$\alpha_{t_i} = e^{-d(f(t_i), \mu_{\hat{p}})}. \tag{6.10}$$

Note that we propose to calculate the distance between the target video and the closest prototype $\mu_{\hat{p}}$, rather than the closest source video $s_{\hat{n}}$, so we increase the robustness by avoiding outliers and considering the distance of the target video to all relevant source videos.

Importantly, the embeddings $f(t_i)$, $f(s_{\hat{n}})$ and prototypes $\mu_{\hat{p}}$ depend on the learnt embedding function f, and hence they continuously change over training iterations. Subsequently, the pseudo-labelling and the confidence scores are also continuously changing, but it would be extremely costly to update them for all target videos after every model update. Fortunately, this change is progressive and updating them at the end of every epoch is sufficient.

Prototype Based Sampling (PBS). Using the proposed prototype confidence scores, we can sample the most reliable target videos, and use them for training our cross-domain losses (Eq. 6.6). However, selected target videos do not only need to be reliable, they also need to widely cover the different actions. Without sufficient coverage, some actions can be poorly aligned. As training progresses, the variety of selected actions may decrease as the labelling gets biased towards actions that are already aligned.

To avoid only aligning a few confident actions, and ignoring the others during domain alignment, we leverage the source prototypes. Let us consider the set of all target videos that are pseudo-labelled using the source prototype p, *i.e.* their relevant source videos are V_p^s. We would like to select the most confident of these, per prototype, to maintain coverage. We thus propose to sample the top $x\%$ most confident target videos assigned to each prototype, *i.e.* we rank the videos assigned to V_p^t based on the confidence scores α_{t_i} and take the top $x\%$ of the list, for each prototype. Thus, Our proposal can deal with imbalanced sets, maintaining the distribution of pseudo-labels, when sampling the most confident ones.

To summarise, we align an embedding space (Sec. 6.1.1) by pseudo-labelling, sampling the most confident target videos per prototype, and iterating.

6.1.5 Disentangled Embedding for Domain Adaptation

Recent works in text-to-video retrieval have shown that learning multiple embedding spaces for different syntax in the textual caption may be beneficial [22, 204]. These learned a separate em-

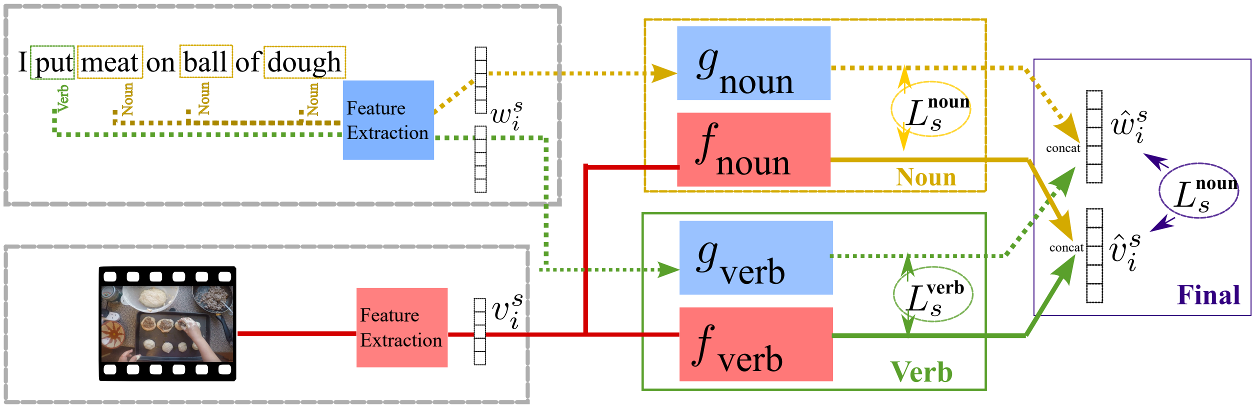


Figure 6.4: The JPOSE architecture learns a separate embedding between each disentangled parts-of-speech in the caption (*i.e.* verbs and nouns) and the video features.

bedding for the nouns, verbs and sentences in the text [22, 204]. Due to our focus on fine-grained actions, we follow the JPoSE architecture [22].

The advantage of JPoSE is that it separates the learning of objects (*i.e.* nouns) from their manipulations (*i.e.* verbs). This is in contrast to a single embedding which is more likely to overfit to certain object interactions (*i.e.* noun-verb pairs), which could affect the ability of the retrieval system to generalise to new noun or verb categories. An overview of JPoSE is given in Fig 6.4. The caption is parsed into the verbs and nouns, and a separate embedding for each is trained. This disentangles the learning of objects and manipulations in the video, since the same visual feature are used in both embeddings. The embedding vectors in all disentangled spaces (visual or textual) are then concatenated to form the final embedding space used for retrieval. Each embedding is trained using its own set of triplets involving embedding-specific relevance relations. In the 'verb' embedding, all captions involving the verb (*e.g.* 'cut') are considered relevant to one another, regardless of the object. Conversely, in the 'noun' embedding, all captions involving the same noun are considered relevant.

Source domain losses (Eq 6.5 in Sec. 6.1.1) enforce the alignment between text and video in each embedding space. Given L_s^{p} is the loss for part-of-speech p, the combined loss for the source domain is:

$$L_s = L_s^{\text{VERB}} + L_s^{\text{NOUN}} + L_s^{\text{FINAL}} \tag{6.11}$$

Aligning Target Videos in a Disentagled Embedding

The disentangled spaces learn generic concepts about actions, *i.e.* objects and manipulations, that are likely to be common to both source and target domains. This is in contrast to the precise verb-noun combinations (the action) learnt in the final embedding space. Therefore, it is hypothesised that alignment of disentangled spaces would achieve better domain adaptation than aligning the final embedding space.

Each embedding space is adapted to the target domain with the cross-domain alignment losses (Sec. 6.1.2) and the target video pseudo-labelling strategy (Sec. 6.1.4). Target pseudo-labelling is applied independently on each embedding space. For example, in the 'verb' embedding, the target video will inherit a relevancy set of video that share the same verb prototype as the nearest source video. The alignment losses are jointly optimised with source domain losses. Our proposed alignment strategy can be applied to any number of embedding spaces:

Ours-Final Source and target videos are aligned in the final embedding space used to conduct retrieval.

$$L = L_s + \lambda_1^F L_{s\to t}^{\times^{FINAL}} + \lambda_2^F L_{t\to s}^{\times^{FINAL}} \tag{6.12}$$

Ours-PoS Source and target videos are aligned in the disentangled embedding spaces.

$$\begin{aligned} L = L_s + &\lambda_1^V L_{s\to t}^{\times^{VERB}} + \lambda_2^V L_{t\to s}^{\times^{VERB}} \\ &\lambda_1^N L_{s\to t}^{\times^{NOUN}} + \lambda_2^N L_{t\to s}^{\times^{NOUN}} \end{aligned} \tag{6.13}$$

Ours Source and target videos are aligned in all embedding spaces.

$$\begin{aligned} L = L_s + & \lambda_1^V L_{s\to t}^{\times^{VERB}} + \lambda_2^V L_{t\to s}^{\times^{VERB}} \\ & \lambda_1^N L_{s\to t}^{\times^{NOUN}} + \lambda_2^N L_{t\to s}^{\times^{NOUN}} \\ & \lambda_1^F L_{s\to t}^{\times^{FINAL}} + \lambda_2^F L_{t\to s}^{\times^{FINAL}} \end{aligned} \tag{6.14}$$

The advantage of **Ours** is that alignment is conducted in multiple views of the same video features. Alignment in the disentangled embedding space may correct pseudo-labelling errors in the final embedding space, and vice versa.

6.2 Experimental Evaluation

First, the proposed domain adaptation benchmark for text-to-video retrieval is defined in Sec 6.2.1, followed by the implementation details of our method which is are detailed in Sec 6.2.2, and the comparative baselines are presented in Sec. 6.2.3. The comparative results of our method to the baselines is reported in Sec. 6.2.4, as well as an ablation of aligning different embedding spaces. Next, the design choices of pseudo-labelling approach are analysed in Sec. 6.2.5. The benefits of learning the disentangled embedding spaces over a single embedding are showcased in Sec. 6.2.6. Finally, the qualitative results of our method are presented in Sec. 6.2.7, including feature visualisations of the embedding spaces.

6.2.1 Dataset and Evaluation

Dataset. The proposed method is evaluated on EPIC-KITCHENS-100 [10], which provides videos associated with free-form narrated captions of egocentric actions. The Source/Target split is evaluated in the context of action recognition in Chapter 5, but considers the class labels and not the

Split	Domain	Participants	Video Gallery	Text Queries
TRAIN	Source / Target	12 / 12	16115 / 26115	4756 / [5907†]
VAL	Source / Target	4 / 4	5002 /7906	1805 / [2822*]

Table 6.1: Proposed EPIC-KITCHENS benchmark. Note that [target text queries] are **never** used for training. †: only used for evaluation *: only used for optimising hyper-parameters.

captions. The domain shift results from recording footage in different environments and recording periods, and is discussed in detail in Chapter 5.

Proposed Benchmark. This chapter proposes the first benchmark for domain alignment in fine-grained text-to-video retrieval (see Table 6.1). The proposed benchmark considers all videos from the TRAIN/VAL UDA action recognition challenge (Chapter 5), as these contain released captions[1]. The VAL split of the UDA challenge is only used to select the hyperparameters. Once the hyperparameters are decided, the model is trained on the TRAIN split. The model is trained with Source videos and their captions, but only Target videos are used (no captions). At test time, the model is evaluated by ranking the Target videos for each Target text query.

Evaluation. For evaluation, the gallery of videos is ranked in order of similarity to each text query in the final embedding space. Given many videos can be relevant to the same caption, we consider two metrics that determine the quality of the entire returned ranking: Mean Average Precision (mAP) and Normalised Discounted Cumulative Gain (nDCG), as proposed by the multi-instance action retrieval challenge for EPIC-KITCHENS-100 [10]. First, we outline how the relevancy is computed between text queries and videos, before defining the metrics.

Actions in EPIC-KITCHENS-100 can be fully described by the verbs and nouns in the text captions, therefore, we use the verb and noun classes to determine relevancy. We denote w_i^V and w_i^N as the verb and noun classes for a textual query, w_i. Similarly, v_j^V and v_j^N are the verb and

[1] The Test split in the challenge does not have released captions publicly available

noun classes for a video in the gallery, v_j. The relevancy $\mathcal{R}$ between w_i and v_j is defined as the averaged Intersection-over-Union of the verb and noun classes:

$$\mathcal{R}(w_i, v_j) = \frac{1}{2}\left(\frac{|w_i^v \cap v_j^v|}{|w_i^v \cup v_j^v|} + \frac{|w_i^N \cap v_j^N|}{|w_i^N \cup v_j^N|}\right) \tag{6.15}$$

Mean average precision (mAP) is the mean of the Average Precision (AP) evaluated for each query, w_i. AP is the area under the precision-recall curve, given a ranked gallery of videos, V_r, containing N relevant videos to w_i. This requires calculating the precision value of the first k retrieved videos in the ranking, $P@k$, for every kth relevant video in the ranking.

$$AP(w_i, V_r) = \frac{1}{N}\sum_{k=1}^{|V_r|} P@k(V_r) \times I_k \tag{6.16}$$

The disadvantage of mAP is that it assumes binary relevancy, as represented by the indicator function, I. For the kth video in the ranking, v_k, the indicator function, I_k, is set to 1 if $\mathcal{R}(w_i, v_k) > 0.5$ and 0 otherwise.

The advantage of nDCG is that it allows non-binary relevancy. Given a query text, w_i, and a ranked gallery of videos, V_r, nDCG is defined as the Discounted Cumulative Gain (DCG) over the Ideal Discounted Cumulative Gain (IDCG):

$$nDCG(w_i, V_r) = \frac{DCG(w_i, V_r)}{IDCG(w_i, V_r)} \tag{6.17}$$

with the DCG being given by:

$$DCG(w_i, V_r) = \sum_{j=1}^{|V_r|} \frac{\mathcal{R}(w_i, v_j)}{log(j+1)} \tag{6.18}$$

IDCG represents the maximum possible nDCG score, which normalises the metric. Given $\hat{V_r}$ is the gallery of videos sorted in order of relevance, then $IDCG(w_i, V_r) = DCG(w_i, \hat{V_r})$.

6.2.2 Implementation Details

Model and Pretraining. For our experiments, we use the public implementation of JPoSE [22] as our base network and modify it for domain adaptation as described in Sec. 6.1.2 and 6.1.4. As often in DA, first the network is pre-trained on source and use it to initialise the proposed domain alignment network. Then the proposed model is trained with cross-modal and cross-domain ranking losses, where the source-target associations are obtained with the proposed target video pseudo-labelling and sampling strategy. At the end of each epoch, the relevance sets are updated.

Architecture and Input Features. Video input features, V, are extracted from a TBN [23] model pre-trained on the source domain videos from Train and Val splits. The 3072 dimensional feature vector is a concatenation of RGB, Flow and Audio features—where audio is the natural sound recorded with the video. The text features are a Word2Vec [217] descriptor of length 200, trained on the Wikipedia corpus [242].

The base architecture is defined in JPOSE [22] where $f(v_i)$ and $g(w_i)$ are multi-layer perceptrons with hidden layer sizes of 228 and 1664, respectively. The output video and text embeddings are vectors of length 256. The action descriptor is of length 512, which is the concatenation of the video and text embeddings.

For optimisation, weights are optimised by SGD with a learning rate of 0.01 and momentum of 0.9. We also adopt a hard-negative mining strategy for the source-domain losses. This considers only negative examples from the nearest 30% of action prototypes to the action prototype of the query. This decreased the number of epochs required for training the Source-Only model, in addition to a small gain in text-to-video retrieval performance, as shown in appendix B.1.

Hyper-parameter Selection. The weights for the alignment losses, $\lambda^{\times}_{s,t} = \lambda^{\times}_{t,s} = 0.1$ were found best when tuning on Val with values $\lambda^{\times}_{s,t}, \lambda^{\times}_{t,s} \in \{0.01, 0.1, 1.0\}$. The weights of the source losses,

λ_s, were those defined in JPoSE [22]. The distance metric, d, used for pseudo-labelling and confidence measures was the cosine distance.

To obtain results on the test set (Train Target), the network is then trained with all losses using the hyper-parameters selected from the Val set. The network is trained for n epochs, where n indicates the number of epochs used to train the best performing model on the Val set.

6.2.3 Baselines

As a lower bound, we include the non-adapted **Source-Only** results. Additionally, we implement per-domain standardisation **PDS** which is a simple alignment technique where the input distribution of each domain is standardised separately[2]. PDS decreases the domain shift. All methods —including ours—are trained with these standardised features. We compare our method to classification-based (*i.e.* typical) domain alignment methods, that we modified for the task of cross-modal video retrieval. All methods were hence trained and tested with the exact same protocol. We list them below.

From shallow models we test **CORAL** [57], which aligns second order statistics, and from deep models we consider:

MMD [73] which minimises the discrepancy between source and target embeddings in Reproducing Kernel Hilbert Space (see Sec. 2.3.2 for more details).

GRL [4] which optimizes a domain discrimination loss in an adversarial manner (see Sec. 2.3.3 for more details).

Deep alignment methods, MMD and GRL, are optimised jointly with the source cross-modal embedding. The weighting of the MMD loss, $\lambda_{mmd} = 0.01$, and GRL, $\lambda_{GRL} = 0.0001$ were found using a grid-search in the range of $[0.0001, 0.001, 0.01, 0.1, 1.0]$.

[2] For **Source-Only**, both Source and Target videos are standardised with the statistics of Train Source, whereas **PDS** standardises Target videos with the statistics of Train Target

We also compare to more recent conditional alignment approaches (discussed in detail in Sec. 2.3.6), and adapt these by replacing the notion of classes with groups of relevant videos that share the same prototype. In particular, we evaluate:

MSTN [133] which minimizes the Euclidean distance between the source and target prototypes.

TPN [134] which minimises the MMD between the source and target examples per prototype.

CDD [130] where the MMD is minimised within each prototype while maximised across examples belonging to differing prototypes.

For fairness, all conditional alignment methods use the same pseudo-labelling strategy as our method and, to make sure conditional alignment baselines have a sufficient number of instances per prototype, we compare alignment of the disentangled Part-of-Speech (PoS) embeddings.

The conditional alignment approaches (MSTN, TPN and CDD) utilise our proposed pseudo-labelling strategy. First the networks are trained on the source domain, similarly to Ours. For these baselines, we implement class-aware sampling [130] which ensures sufficient number of examples for each prototype. This samples 30 instances from 32 different classes each mini-batch. For classes containing less than 30 instances all instances of the class are used and classes containing less than 3 samples are not sampled. The weighting for each loss is $\lambda_{MSTN} = 0.1$, $\lambda_{TPN} = 0.1$, $\lambda_{CDD} = 0.005$ for MSTN, TPN and CDD, respectively. These were found using a grid-search in the range of $[0.0001, 0.001, 0.005, 0.01, 0.1, 1.0]$.

6.2.4 Comparative Results

Table 6.2(a) shows how the proposed method performs against models trained solely on source (Source-Only) and both marginal and conditional alignment baselines. We report retrieval results on Target VAL (referred to as Val) and on Target TRAIN (referred to as Test[3]). Surprisingly, the simpler marginal alignment methods (PDS and CORAL) yield better performance than the

[3] We report best Val epoch based on nDCG and use it as early stopping criteria when testing on Target TRAIN.

	Method	Alignment Space				Val (Target VAL)		Test (Target TRAIN)	
		PoS	Final	M	C	nDCG	mAP	nDCG	mAP
(a)	Source-Only	×	×	×	×	40.45	7.32	35.25	5.05
	PDS	✓	×	✓	×	40.78	7.65	35.96	5.42
	CORAL [57]	✓	×	✓	×	40.70	7.79	36.32	5.38
	MMD [73]	✓	×	✓	×	40.60	7.49	36.26	5.43
	GRL [4]	✓	×	✓	×	42.45	5.20	36.64	5.61
	TPN [134]	✓	×	✓	✓	42.19	8.27	36.86	5.73
	CDD [130]	✓	×	✓	✓	42.00	8.48	36.48	5.80
	MSTN [133]	✓	×	✓	✓	42.85	**8.63**	37.62	5.87
	Ours-PoS	✓	×	✓	✓	**43.14**	8.59	**38.06**	**6.21**
(b)	Ours-Final	×	✓	✓	✓	43.07	8.85	37.74	6.20
	Ours	✓	✓	✓	✓	**44.00**	**9.10**	**38.21**	**6.34**

Table 6.2: (a) Comparison with alignment baselines (M: Marginal alignment, C: Conditional alignment). TPN, CDD, MSTN use our sampling strategy for fair comparison. (b) Which space(s) to use for alignment: Disentangled Part-of-Speech or the Final embedding (see Sec. 6.1.5).

deep models (MMD) for our benchmark, but overall they provide marginal improvements over the Source-Only model. Adversarial training (GRL) provides a larger improvement, however, we find this to be highly unstable as training progresses. The conditional alignment approaches (TPN, CDD, MSTN) perform better than marginal alignment, with MSTN performing the best. Our approach (Ours-PoS), using cross-domain alignment, outperforms all baselines on Test.

Table 6.2(b) shows two alternatives of the proposed model: Ours-Final, where the domains are aligned in the final embedding space, and Ours, where all embedding spaces are aligned. Even with single-view domain alignment (Ours-Final), our approach outperforms all baselines. Finally, by aligning all embeddings (Ours) jointly, we observe further improvements over Ours-PoS.

(a) Labelling Method

Method	Val	Test
Proto	43.55	37.92 ± 0.12
Ours	44.00	38.21 ± 0.08

(b) Confidence Method

Method	Val	Test
Neighbour	43.79	37.64 ± 0.30
Ours	44.00	38.21 ± 0.08

(c) Target Video Sampling Method

Method	Val	Test
All	42.66	36.53 ± 0.14
Uniform (60%)	43.57	37.88 ± 0.08
Ours (60%)	44.00	38.21 ± 0.08

Table 6.3: Ablation Studies on labelling, confidence and sampling of pseudo-labels for text-to-video Retrieval (nDCG). Mean and standard deviation over 3 runs, reported on the Test set.

6.2.5 Analysis of Target Pseudo-labelling Design Choices

This section verifies several design choices of our proposed target pseudo-labelling strategy . First, the proposed pseudo-labelling, confidence and sampling strategies are compared to alternative approaches. Next, the impact of varying the proportion of pseudo-labelled target data during training is assessed. Finally, an analysis of the pseudo-labels produced by the proposed and alternative strategies is presented.

Table 6.3 answers the following questions on the target pseudo-labelling strategy:

How to pseudo-label? Table 6.3a shows that inheriting from the nearest source video performs better than using the nearest prototype's label (Proto) because it makes fewer assumptions on the target distribution.

How to assign confidence? Table 6.3b shows that the proposed confidence measure α—based on the distance to the prototype—is more robust than using a confidence based on the distance to the closest source video instead (Neighbour).

How to sample target videos? In Table 6.3c we compare the proposed sampling strategy to uniform sampling (Uniform). By sampling confident examples per prototype, this ensures a variety of actions is covered, thus improving the domain alignment. Without our sampling strategy, target videos are sampled from fewer actions as training progresses. Both sampling approaches, Uniform and Ours, outperform training with all pseudo-labelled target videos.

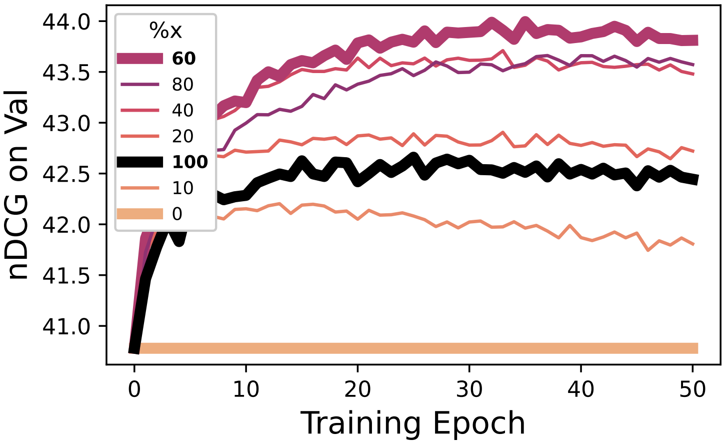


Figure 6.5: Performance as we vary the proportion of target sampled.

How many are relevant? Figure 6.5 shows the impact of training with different proportions of labelled target examples. The results are shown on Val. Removing the least confident examples significantly improves target performance and that training with the most confident 60% performs best. As expected, using all target instances (100%), which would include erroneous pseudo-labels, leads to a substantial drop in performance. This validates our sampling proposal.

Table 6.4 compares the impact of the proposed method if pseudo-labelling or alignment uses source text, instead of source videos. In both experiments cross-domain alignment is conducted in

(a) Pseudo-Labelling Modality

Source Modality	Val	Test
Text	42.54	37.41 ± 0.13
Video (Our-PoS)	43.14	38.06 ± 0.12

(b) Alignment Modality

Source Modality	Val	Test
Text	42.81	38.08 ± 0.06
Video (Our-PoS)	43.14	38.06 ± 0.12

Table 6.4: Ablation on which source modality to use for pseudo-labelling and alignment, reporting retrieval performance on Target (nDCG). Mean and standard deviation over 3 runs reported on the Test set.

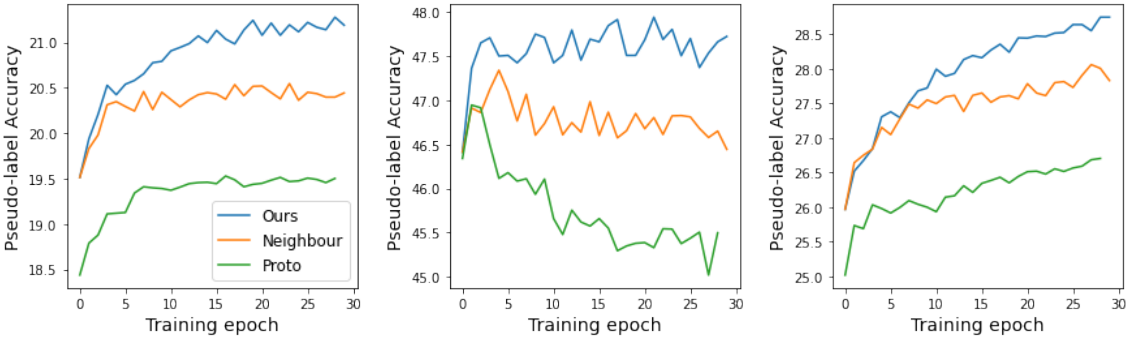
the disentangled part-of-speech embedding spaces, and is compared to Ours-PoS.

Which source modality should be used for pseudo-labelling? Table 6.3a shows that it is best to pseudo-label target videos based on the nearest source videos, rather than the nearest source text. Pseudo-labelling using source text introduces multiple sources of errors, induced by the visual domain shift, and the misalignment between text and video. Pseudo-labelling using source videos is more accurate, as only the visual domain shift is a source of error.

Which source modality should be used for alignment? Table 6.3b compares alignment of target videos with relevant source texts or source videos. Similar performance on Test is achieved by aligning target videos with source text or source videos sharing the same prototypes. The smaller validation set (Val) shows improved performance by aligning target videos to source videos. This is due to the fact there are fewer captions than videos, such that aligning source and target videos is less likely to over-fit domain alignment.

Next, the accuracy and diversity of the pseudo-labels is analysed as training progresses. Figure 6.6 shows the pseudo-label accuracy over training epochs, which compares Ours to the alternative pseudo-labelling approaches, Neighbour and Proto, defined in the discussion of Table. 6.3. Note that pseudo-label accuracy improves over training iterations as the domains are better aligned. Ours, with the robust prototype-based confidence measure, outperforms using the distance to the

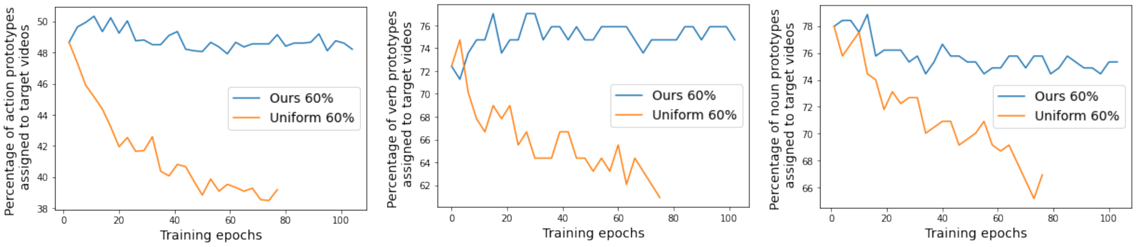
Analysis of the accuracy of the pseudo-labels assigned to target videos



(a) Action (Final Embedding) (b) Verb (c) Noun

Figure 6.6: Pseudo-label accuracy for target videos in the final, verb and noun embedding as training progresses. Ours (BLUE) produces more accurate labels as training progresses compared to alternative pseudo-labelling strategy, Proto (GREEN), and confidence measure, Neighbour (ORANGE).

Analysis of the diversity of target video pseudo-labels



(a) Action (Final Embedding) (b) Verb (c) Noun

Figure 6.7: The percentage of prototypes (pseudo-labels) assigned to target videos in the final, verb and noun embedding as training progresses. We see that Ours (BLUE) maintains a greater variety of labels than uniform sampling (ORANGE).

Architecture	Method	Val (nDCG)	Test (nDCG)
SE	PDS	33.76	31.26 ± 0.04
	Ours	34.58	31.67 ± 0.09
DE	PDS	40.78	35.96 ± 0.14
	Ours	44.00	38.21 ± 0.08

Table 6.5: Comparison between a disentagled embeddding for each part-of-speech (DE) *vs.* a single embedding (SE). PDS is trained only with source-domain losses, and Ours is trained with the addition of cross-domain alignment losses. Sec. 6.2.3 for details on PDS.

nearest source video as the confidence measure (Neighbour). Pseudo-labelling based on the nearest source prototype (Proto) incorrectly labels more target examples.

Our prototype sampling strategy produces more diverse pseudo-labels, compared to uniform sampling. Figure 6.7 shows the percentage of prototypes assigned to all target examples as pseudo-labels. Our proposed sampling strategy ensures that the target videos are assigned to a large variety of prototypes, however, using uniform sampling favors assignments to prototypes corresponding to large clusters, introducing an increasing bias towards them.

6.2.6 Single Embedding *vs*. Disentangled Embeddings

The architecture used throughout this thesis (including Ours) consists of a disentangled embedding for each part-of-speech (DE). This section compares the performance of DE to a single embedding (SE). SE is implemented with the same architecture as a single disentangled embedding (verb or noun). Table 6.5 reports retrieval performance on Target for each architecture. Comparing to each architecture trained with source-domain losses (PDS), DE produces higher retrieval performance on the target gallery of videos compared to SE. This shows that learning disentangled embeddings is more suited for retreival than a single embedding, even without domain alignment. With the addition of cross-domain alignment losses (Ours), retrieval performance for both architectures improves. However, the relative improvement of domain adaptation on Test is 2.3% for DE and 0.4% for SE. This shows that our method is beneficial for both architectures, but is most effective when disentangled embedding spaces are learnt.

6.2.7 Qualitative Results

First, this section shows examples of videos retrieved from the embedding learned with Source-Only, and the proposed method, Ours. Next, feature spaces are visualised showing the alignment of source and target videos, in addition to the alignment of target videos to target text queries.

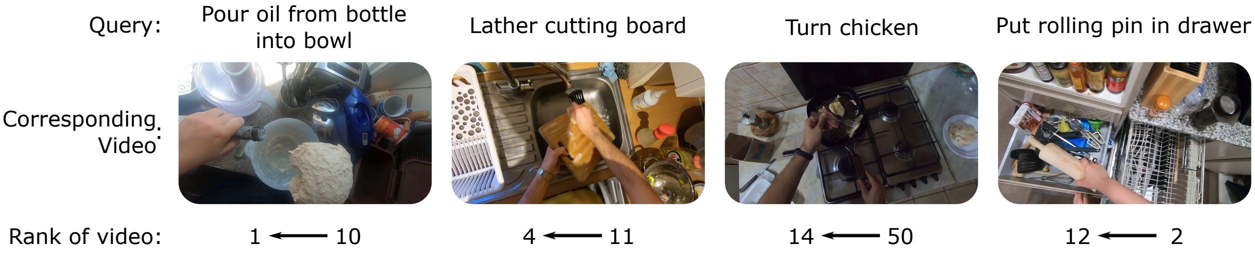


Figure 6.8: Qualitative results of text-to-video retrieval. We show the query caption and the corresponding video along with the change in rank, $A \leftarrow B$, of the proposed method (A) and Source-Only baseline (B).

Figure 6.8 shows how the rank of the first relevant video to specified text queries changes from the Source-Only model and Ours. Ours results in more relevant videos retrieved higher in the ranking, however, some failure cases exist. The video of '*put rolling pin in drawer*' is retrieved lower in the ranking, with videos depicting actions of moving utensils retrieved higher in the ranking.

Fig. 6.10 shows the UMAP visualisation of the final embedding space for Source-Only (Top) and Ours (Bottom). Ours not only shows greater alignment of source (BLUE) and target (ORANGE) videos but more distinct clusters of videos depicting the same action/part-of-speech. The

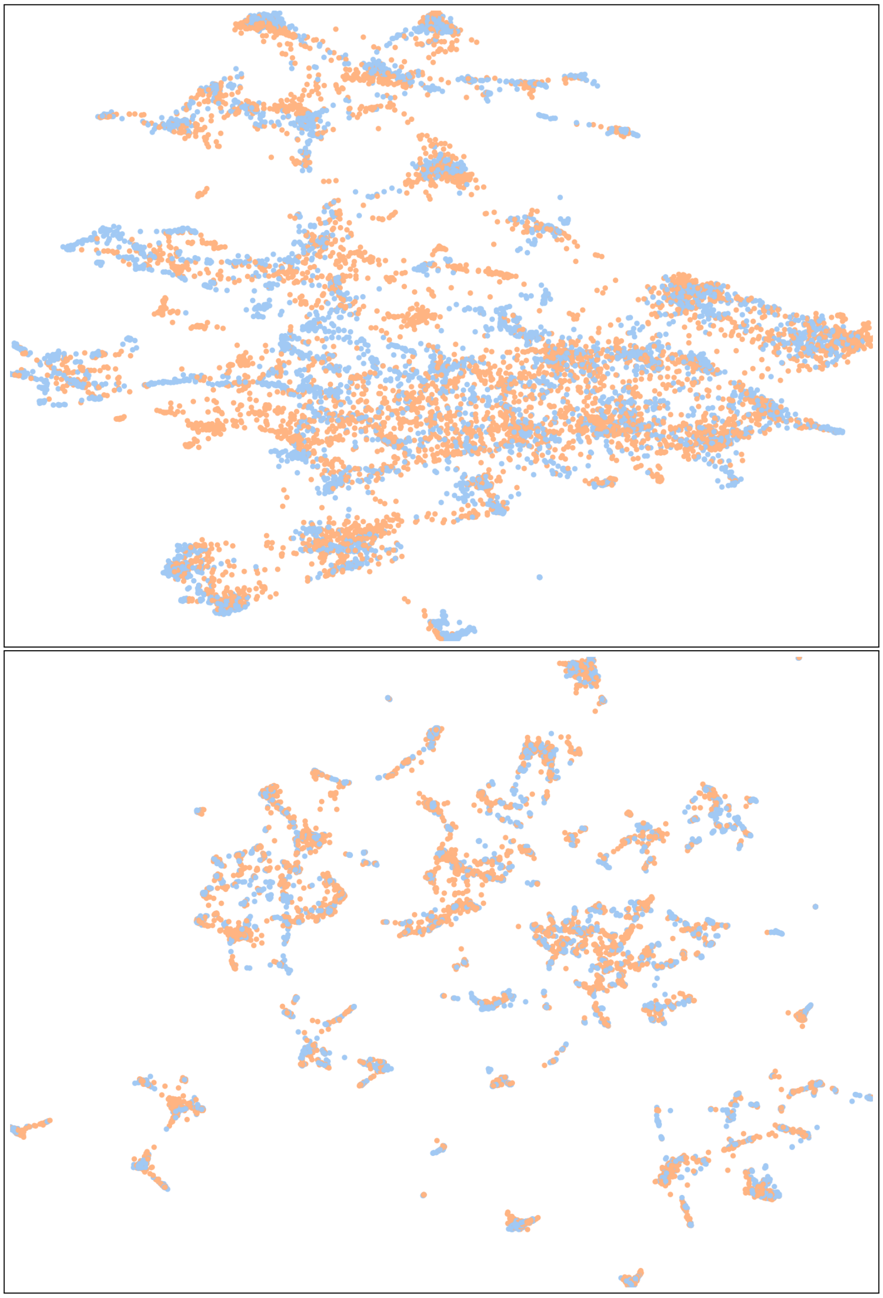
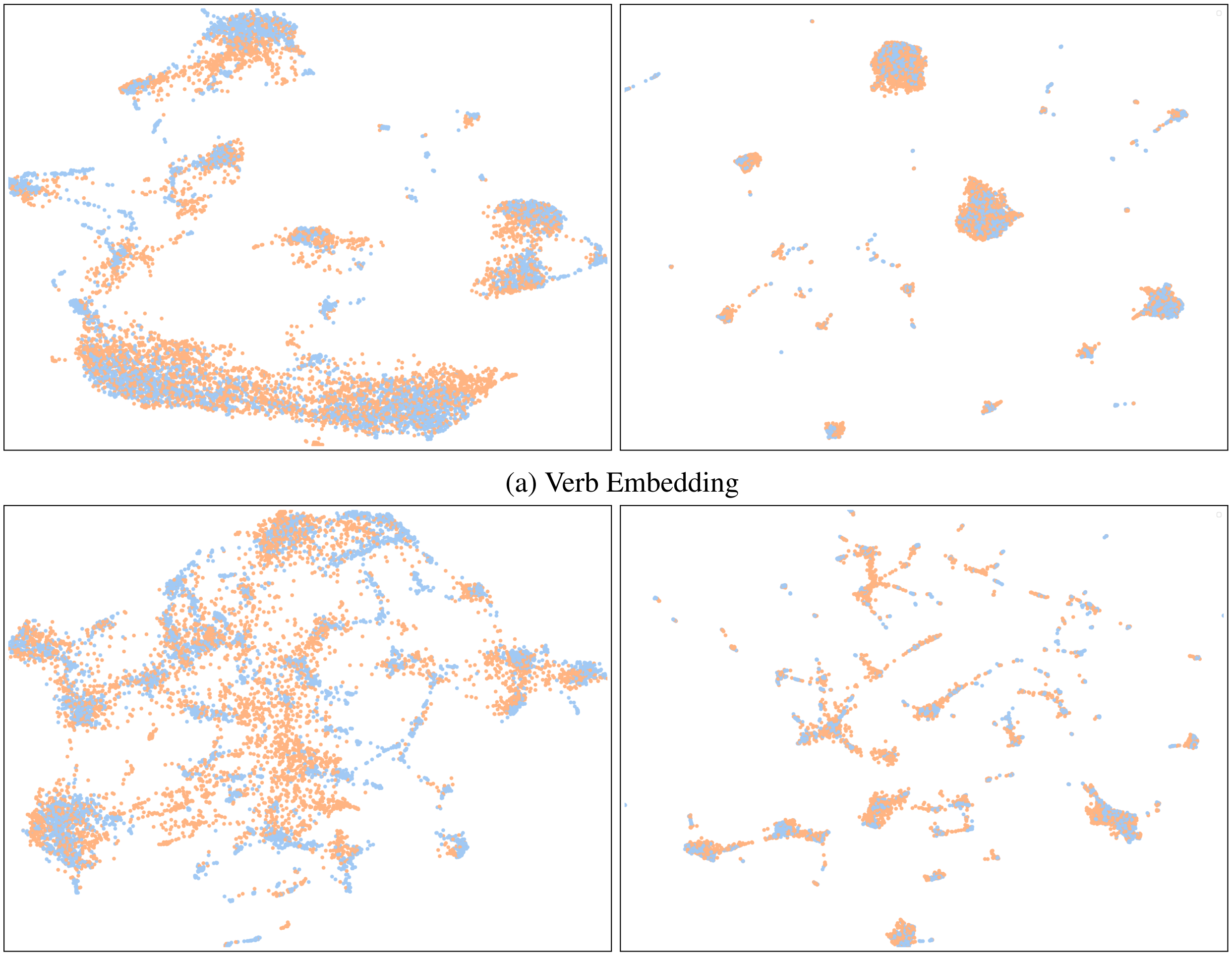


Figure 6.9: UMAP visualisation of final embedding, Target in ORANGE and Source in BLUE. Top: Source-Only, Bottom: Ours



(a) Verb Embedding

(b) Noun Embedding

Figure 6.10: UMAP visualisation of the disentagled part-of-speech embeddings, Target in ORANGE and Source in BLUE. LEFT: Source-Only, RIGHT: Ours

UMAP visualisations in the disentangled part-of-speech embedding space is shown in Fig. 6.10. The embedding spaces for Source-Only (LEFT) show a larger domain shift between videos in the noun embedding than the verb embedding. However, our proposed method (RIGHT) is able to align both verb and noun embedding spaces.

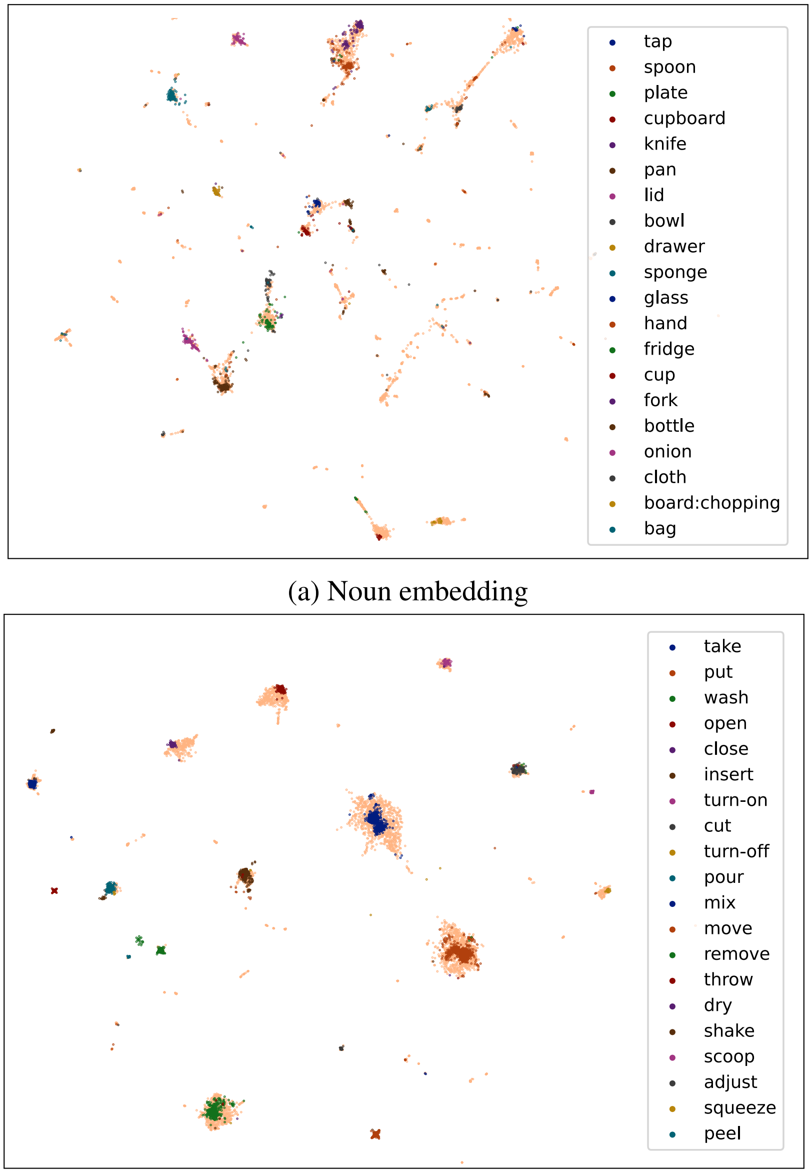
Fig. 6.9 shows the UMAP visualisation of the target videos (ORANGE) and target text (other colours). Target-text were used as queries to evaluate text-to-video retrieval, but were not present during training. For clarity, we only show the text embedding from the 20 verb/noun prototypes with the most videos. Target text is aligned with target video clusters.

6.3 An Exploration into Cross-dataset Adaptation for Fine-grained Action Retrieval

The source and target domain in this chapter contained videos of cooking actions from a first person perspective. However, one may want to retrieve video snippets of fine-grained actions from instructional videos on the web. This work would adapt a text-to-video retrieval system trained on EPIC-KITCHENS to a smaller dataset of instructional cooking videos taken from the web, YouCook2 [150].

YouCook2 is a dataset of cooking videos taken from the web, with annotated actions. In total, 15,000 descriptions of short action sequences have temporal annotations, in contrast to the 90,000 annotated fine-grained actions in EPIC-KITCHENS-100. Due to the larger size, EPIC-KITCHENS will be the source domain, while YouCook2 will be the target domain. The difficulty of this text-to-video retrieval task is the large domain shift between the two datasets in both the video and text modalities:

The Visual Domain Gap YouCook2 contains instructional videos, while EPIC-KITCHENS contains video footage of participants unscripted, household activities. This leads to a large



(a) Noun embedding

(b) Verb embedding

Figure 6.11: UMAP visualisation of chosen target text queries and all target videos. Visualising text queries from 20 part-of-speech prototypes which have the most videos. Target videos are shown in ORANGE.

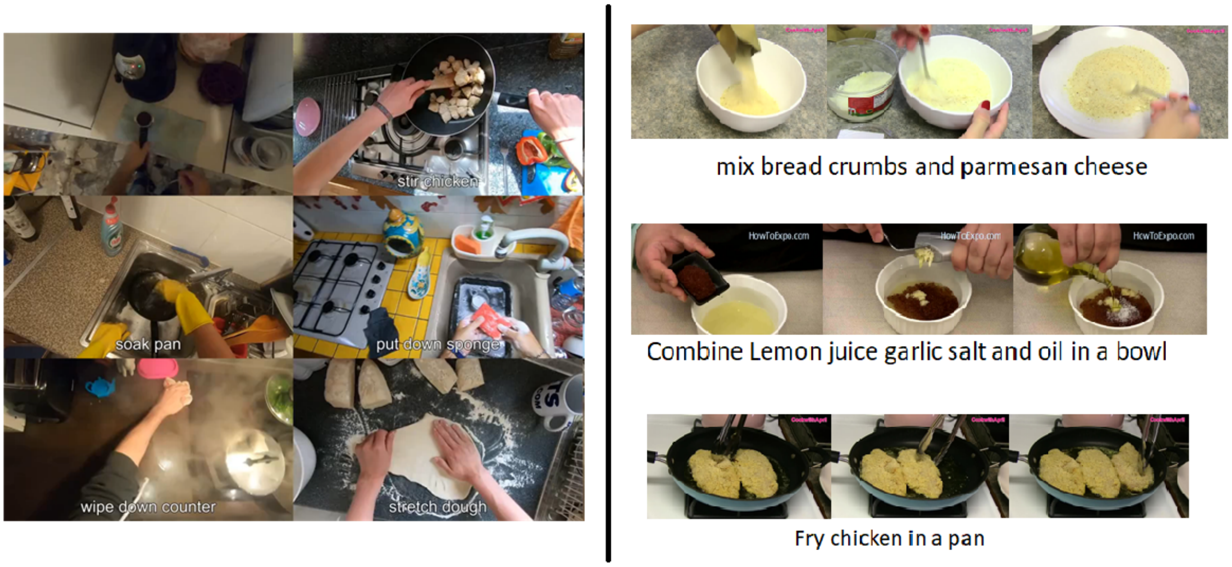


Figure 6.12: Example frames from EPIC-KITCHENS-100 (Left) and YouCook2 (Right)

difference in the video content and the type of camera shots, as shown in Fig. 6.12. While the camera shots vary in YouCook2, most action sequences are extreme closeups on the action, taken from a third person perspective. This is in contrast to the first person footage in EPIC-KITCHENS. In addition, the videos in YouCook2 can contain transitions between different camera shots, and text is sometimes overlaid onto the video.

Careful consideration will be needed to choose the feature extractor for the visual modalities, as this must perform well on both first person and third person video. Future research could focus on training a feature extractor to perform well on downstream tasks for both first-person and third person video. This was explored recently in Ego-Exo [243], which utilised knowledge distillation losses to learn egocentric ques from third person video when training models on Kinetics [26].

Different Temporal Granularity The annotated temporal bounds of actions in YouCook2 contain multiple fine-grained actions. The mean length of an action sequence is 19.6s. In contrast, EPIC-KITCHENS contains a single annotated fine-grained action that lasts on average 3.7s. A cross-modal embedding trained on single actions (*i.e.* EPIC-KITCHENS), may not generalise to the multiple actions described in captions of YouCook2's video sequences.

Future research could combine consecutive instances from EPIC-KITCHENS, in order to model action sequences instead of a single action. The input video would span the length of multiple actions, while the text modality could contain the nouns and verbs describing all actions depicted in the video .

Semantic Shift The meaning of verbs between YouCook2 and EPIC-KITCHENS differs greatly. YouCook2 annotations typically describe instructions, similar to that in a recipe book (*e.g.* '***add*** *oil to pan*' and '***season*** *water*'). While annotations in EPIC-KITCHENS describe how the action is taking place (*e.g.* *'**pour** oil into pan*' and '***sprinkle*** *salt into pan*').

In order to simplify the problem, a small portion of the YouCook2 videos should be re-annotated in the style of EPIC-KITCHENS, and used for evaluation. Initial research can then focus on overcoming the visual domain gap, while avoiding the issue of the semantic shift between the text captions. Later research could utilise a limited number of YouCook2 captions to learn how to map EPIC-KITCHENS captions into the style of YouCook2.

Given the challenges we found with this exploration, this direction was not further explored and next steps are left for future work.

6.4 Conclusion

In this work, we have introduced the problem of unsupervised domain adaption for text-to-video retrieval. Given a source dataset of captioned videos for training, the goal is to perform retrieval on a distinct target set of uncaptioned videos. Our experiments show that marginal alignment approaches commonly used for domain adaptation lead to underwhelming results on this task, as they are unable to align the local structure of the embeddings. At the heart of our proposed approach lies an iterative pseudo-labelling process, which associates source and target samples at train time. These pseudo-labelled samples fuel our cross-domain cross-modal learning-to-rank approach which aligns the source and the target video galleries. Our experiments validate this strategy and highlight the importance of selecting confident target examples, using source prototypes, during alignment.

This work is not without limitations. Our work assumes the text in the source and target domains is written in a similar style, with a large overlap in their vocabulary. However, in addition to video, text could also exhibit a large domain shift. Future work could explore the domain shift between the text in the source and the target domain.

This work has helped mitigate the impact of misalignment due to differing classes in source and target domains with our prototype sampling strategy, which aims to sample on classes that are shared between the source and the target domains. In addition, aligning individual parts-of-speech helps in this open-set setting as verbs and nouns are more likely to be common to both domains, compared to their combinations (actions) which will be less likely to be shared in both domains. The limitation of our approach is that it still assumes a large overlap of verb and noun classes. Future work could look into the open-set setting where there is severely limited overlap in classes.

Chapter 7

Conclusion

Dataset bias remains an issue for fine-grained action understanding, since datasets collect long, untrimmed videos, with a limited number of participants, locations and cameras. Surprisingly, little attention has been given to the domain shift between videos during training, and videos encountered when models are deployed. This thesis explored unsupervised domain adaptation in order to improve the performance of fine-grained action understanding tasks under a domain shift.

Several domain shifts were shown to affect the performance of fine-grained action understanding tasks. Chapter 4 showed that training fine-grained action recognition models with videos captured in a single location, greatly impacts the ability of models to generalise to new environments. In addition, Chapter 5 highlighted a domain shift that occurs between the videos of the same participants captured two-years apart. The wear and tear in the environment, new objects, and different cameras used to film the participants, all contribute to the domain shift between the footage taken two-years apart. This affects the performance of both action recognition (Chapter 5) and text-to-video retrieval (Chapter 6).

Video provides additional opportunities for domain adaptation compared to image based approaches, including the presence of multiple time-synchronised modalities. Chapter 4 showed that three modalities available to video (appearance, motion and sound) have differing levels of robustness to changes in the environment, which can be exploited for domain adaptation. The proposed multi-modal self-supervised objective improves the recognition of both the verbs and nouns used to describe actions, when evaluated in the target environment. This was the first work in video to use multiple modalities for domain adaptation, and has inspired further research into multi-modal domain adaptation [42, 44, 129].

Self-supervision is an effective domain adaptation method in the open-set setting, as shown in Chapter 4. Unlike common approaches in domain adaptation, which assume the ideal feature representation is one where the source and target videos aligned (Chapter 2), self-supervision does not impose assumptions on the distributions of each domain. This is important for open-set adaptation as the domains do not share the same label space. Self-supervision provides a promising direction for future open-set domain adaptation research.

Chapter 6 explored domain adaptation in the context of text-to-video. This showed that aligning videos that depict the same action, outperforms methods that align the marginal distributions of the source and target domains. In the absence of text descriptions of actions in the target domain, a pseudo-labelling strategy was proposed. A robust confidence measure was proposed to select the most confidently pseudo-labelled target videos for training. This improved retrieval performance by reducing the impact of noisy pseudo-labels. Rather than aligning visual representations of actions (verb-noun combinations), Chapter 6 showed that separating learning of objects (nouns) from manipulations that are independent of the object (verbs) can improve target domain performance. This improvement is due to the learning of more generic semantic concepts that are more likely to be shared by videos in both source and target domains.

The collection of untrimmed, unscripted video provides a practical setting for domain adaptation, whereas most domain adaptation benchmarks are created from curated datasets [1, 2, 60, 78, 88, 234]. Many action recognition datasets [25, 26, 146] contain carefully selected videos, such that the vast majority are relevant for the task of action recognition. In addition, the videos are often trimmed such that the action is depicted in most of the video. Prior domain adaptation benchmarks use these datasets as an unlabelled target domain [1, 2]. However, this ignores the fact that these videos are chosen specifically for the target task, and will differ greatly from uncurated real-world footage. Datasets of fine-grained actions provide long, untrimmed videos, by observing the natural interactions of participants with their environments. This provides a promising direction for evaluating unsupervised domain adaptation.

7.1 Future Work

This thesis, in-particular Chapter 4, showed the benefits of using multiple modalities of video, via self-supervision, to improve domain adaptation for fine-grained action recognition. However, several limitations of the current approach exists which are discussed further in Section 7.1.1. In addition, Chapter 6 showed the benefits of aligning local structures with the action understanding task in mind, using pseudo-labels to reason about the action labels for target videos. However, this didn't fully exploit the multiple modalities of video, which we discuss in more detail in Section 7.1.2.

Several outstanding challenges remain for domain adaptation for fine-grained action understanding. The first is the assumption of actions can be detected effectively in the target domain and the task is to only recognise the actions. This is discussed in more detail in Section 7.1.3. The second is the requirement of target data that is needs to be available, and is discussed in Section 7.1.4. Finally, the same semantics of labels are needed for both the source and target domain classes, and is discussed further in Section 7.1.5.

7.1.1 Additional or Improved Self-supervised Objectives

Chapter 4 showed that self-supervision is an effective strategy to utilise multi-modal video data for domain adaptation. However, the separate loss for the distribution alignment and self-supervision do not always work well together. Future work can look into how to combine the distribution alignment and multi-modal self-supervision into a single loss.

The training of the temporal correspondence objective could be improved with the use of *hard-negatives* to increase the difficulty of the self-supervised task. The proposed self-supervised loss showed little signs of over-fitting, despite the saturated training loss. *Hard-negatives* may be able to further improve the temporal correspondence between modalities, as this would focus the loss on non-corresponding modalities that are temporally close in time.

Additional self-supervised objectives may be useful for domain adaptation of fine-grained actions. Recent work has also shown that predicting the temporal order of frames can improve domain adaptation for course-grained actions [43]. Improved target performance could be achieved by jointly optimising multiple self-supervised objectives, which applied to both source and target, may enable the model to learn different properties of video that are common to both domains.

7.1.2 Classifier Adaptation with Multi-modal Agreement

Chapter 4 focused on utilising multiple modalities to align the feature representations from both source and target domains. However, the classifier has not been refined with target data to improve action recognition. Additionally, Chapter 6 showed the benefits of refining models with pseudo-labelled target videos to improve action understanding.

Many pseudo-labels (classifier predictions) will be incorrect and it is hypothesised that the consensus from multiple modalities can increase the proportion of pseudo-labels matching the ground-truth. Given the classifier response, $p^m(x^t)$, from $m \in M$ modalities, a subset of target examples, X_c^t can be selected to refine the multi-modal classifier.

$$X_c^t = \{x \in X^t \quad | \quad \operatorname{argmax} p^m(x^t) = \operatorname{argmax} p^{m'}(x^t) \quad \forall m \in M \wedge m' \in M\} \tag{7.1}$$

Recent work [129] has shown large improvements using a similar multi-modal consensus strategy on more balanced datasets. However, our initial experiments training with a selected subset produced by the consensus of Flow and RGB modalities worsened action recognition performance, as it exaggerated the bias towards the overly confident majority classes. A future research direction could find out how to overcome the large class imbalance, when generating pseudo-labels with multi-modal data.

7.1.3 Domain Adaptation for Action Detection

The main limitation of this thesis is the assumption that start and end times of actions in the target domain are available during training. In practice, this would require an additional annotation effort, or a generic action detector to determine the start and end times. Ideally, domain adaptation methods should be able to adapt to untrimmed video in the target domain.

7.1.4 Domain Generalisation

For some applications, it may be impractical to collect any video data from the target domain. This would require models to generalise to new domains, without observing target data. Future research could train models on multiple source domains (*e.g.* video from multiple participants) in order to generalise to an unseen target domain (*e.g.* an unseen participant).

7.1.5 Cross-dataset Domain Adaptation

Finally, this work only considered the domain gap between videos within EPIC-KITCHENS-100, however, cross-dataset adaptation also introduces additional challenges. The semantics between the action labels and/or descriptions will vary due the ambiguous nature of verbs. In addition there will exist a larger visual shift, especially in the case of a viewpoint change.

Appendix A

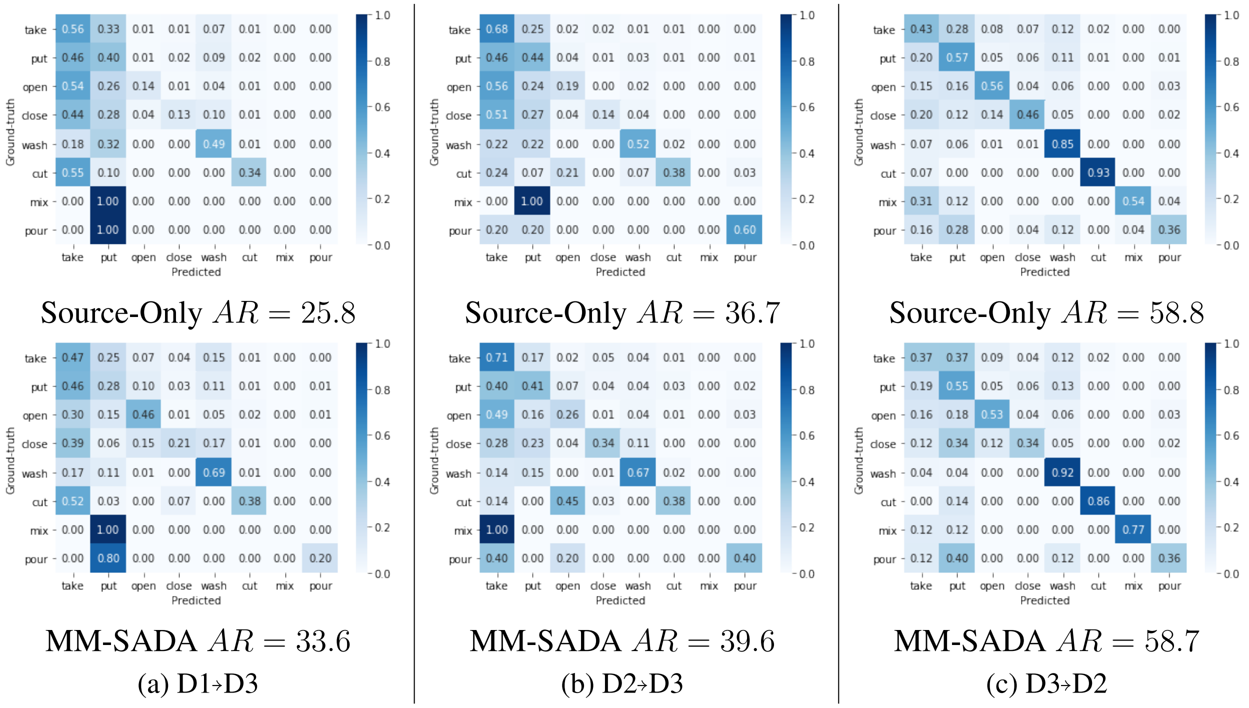
MM-SADA Results

A.1 Individual Performance of Modalities

The performance of the individual modalities of the source-only, adversarial-only and MM-SADA models are shown in Fig A.1. Note that these models were trained with the late-fusion of modalities, and their multi-modal performance is shown in Table 4.5. MM-SADA outperforms both adversarial-only and source-only, on both RGB and Flow.

A.2 Additional Confusion Plots

The confusion plots for MM-SADA and source-only, evaluated on the domain pairs not provided in Fig. 4.7, are shown in Fig. A.1



Source-Only $AR = 25.8$

	take	put	open	close	wash	cut	mix	pour
take	0.56	0.33	0.01	0.01	0.07	0.01	0.00	0.00
put	0.46	0.40	0.01	0.02	0.09	0.02	0.00	0.00
open	0.54	0.26	0.14	0.01	0.04	0.01	0.00	0.00
close	0.44	0.28	0.04	0.13	0.10	0.01	0.00	0.00
wash	0.18	0.32	0.00	0.00	0.49	0.01	0.00	0.00
cut	0.55	0.10	0.00	0.00	0.00	0.34	0.00	0.00
mix	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
pour	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00

MM-SADA $AR = 33.6$

	take	put	open	close	wash	cut	mix	pour
take	0.47	0.25	0.07	0.04	0.15	0.01	0.00	0.00
put	0.46	0.28	0.10	0.03	0.11	0.01	0.00	0.01
open	0.30	0.15	0.46	0.01	0.05	0.02	0.00	0.01
close	0.39	0.06	0.15	0.21	0.17	0.01	0.00	0.00
wash	0.17	0.11	0.01	0.00	0.69	0.01	0.00	0.00
cut	0.52	0.03	0.00	0.07	0.00	0.38	0.00	0.00
mix	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
pour	0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.20

(a) D1→D3

Source-Only $AR = 36.7$

	take	put	open	close	wash	cut	mix	pour
take	0.68	0.25	0.02	0.02	0.01	0.01	0.00	0.00
put	0.46	0.44	0.04	0.01	0.03	0.01	0.00	0.01
open	0.56	0.24	0.19	0.00	0.02	0.00	0.00	0.00
close	0.51	0.27	0.04	0.14	0.04	0.00	0.00	0.00
wash	0.22	0.22	0.00	0.00	0.52	0.02	0.00	0.00
cut	0.24	0.07	0.21	0.00	0.07	0.38	0.00	0.03
mix	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
pour	0.20	0.20	0.00	0.00	0.00	0.00	0.00	0.60

MM-SADA $AR = 39.6$

	take	put	open	close	wash	cut	mix	pour
take	0.71	0.17	0.02	0.05	0.04	0.01	0.00	0.00
put	0.40	0.41	0.07	0.04	0.04	0.03	0.00	0.02
open	0.49	0.16	0.26	0.01	0.04	0.01	0.00	0.03
close	0.28	0.23	0.04	0.34	0.11	0.00	0.00	0.00
wash	0.14	0.15	0.00	0.01	0.67	0.02	0.00	0.00
cut	0.14	0.00	0.45	0.03	0.00	0.38	0.00	0.00
mix	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
pour	0.40	0.00	0.20	0.00	0.00	0.00	0.00	0.40

(b) D2→D3

Source-Only $AR = 58.8$

	take	put	open	close	wash	cut	mix	pour
take	0.43	0.28	0.08	0.07	0.12	0.02	0.00	0.00
put	0.20	0.57	0.05	0.06	0.11	0.01	0.00	0.01
open	0.15	0.16	0.56	0.04	0.06	0.00	0.00	0.03
close	0.20	0.12	0.14	0.46	0.05	0.00	0.00	0.02
wash	0.07	0.06	0.01	0.01	0.85	0.00	0.00	0.00
cut	0.07	0.00	0.00	0.00	0.00	0.93	0.00	0.00
mix	0.31	0.12	0.00	0.00	0.00	0.00	0.54	0.04
pour	0.16	0.28	0.00	0.04	0.12	0.00	0.04	0.36

MM-SADA $AR = 58.7$

	take	put	open	close	wash	cut	mix	pour
take	0.37	0.37	0.09	0.04	0.12	0.02	0.00	0.00
put	0.19	0.55	0.05	0.06	0.13	0.00	0.00	0.00
open	0.16	0.18	0.53	0.04	0.06	0.00	0.00	0.03
close	0.12	0.34	0.12	0.34	0.05	0.00	0.00	0.02
wash	0.04	0.04	0.00	0.00	0.92	0.00	0.00	0.00
cut	0.00	0.14	0.00	0.00	0.00	0.86	0.00	0.00
mix	0.12	0.12	0.00	0.00	0.00	0.00	0.77	0.00
pour	0.12	0.40	0.00	0.00	0.12	0.00	0.00	0.36

(c) D3→D2

Figure A.1: Confusion matrices of MM-SADA (bottom) and Source-Only (top) for domain pairs not shown in Fig. 4.7.

	$D2 \rightarrow D1$	$D3 \rightarrow D1$	$D1 \rightarrow D2$	$D3 \rightarrow D2$	$D1 \rightarrow D3$	$D2 \rightarrow D3$	Mean
RGB source-only	37.0	36.3	36.1	44.8	36.6	33.6	37.4
RGB (Adversarial-only)	37.8	41.1	45.7	45.1	38.1	41.2	41.5
RGB (MM-SADA)	41.7	42.1	45.0	48.4	39.7	46.1	43.9
Flow source-only	44.6	44.4	52.2	54.0	41.1	50.0	47.7
Flow (Adversarial-only)	45.5	46.8	51.1	54.6	44.2	47.1	48.2
Flow (MM-SADA)	45.0	45.7	49.0	58.9	44.8	52.1	49.3

Table A.1: Ablation of MM-SADA on individual modalities, reporting predictions from each modality stream before late fusion. Different from Table. 4.7, modalities are trained with a single multi-modal action classification loss.

Appendix B

Text-to-Video Retreival

B.1 Hard Negative Mining for Source Domain Losses

The negative examples for the source domain triplet losses (Eq. 6.5), are randomly selected from instances assigned to the nearest 30% of prototypes to the anchor, after the fifth epoch. This percentage found best through a grid search in range: $[10\%, 30\%, 50\%, 70\%, 90\%, 100\%]$. For the first 5 epochs, negative instances are randomly sampling from all prototypes, as distances between examples will not be semantically meaningful before an embedding is learnt.

Figure B.1 compares the proposed hard-negative sampling strategy to randomly sampling from all prototypes as done in JPOSE [22]. The main improvement of hard-negative mining is the faster convergence, only 10 epochs is needed to reach the same performance of JPOSE [22] after 50 epochs.

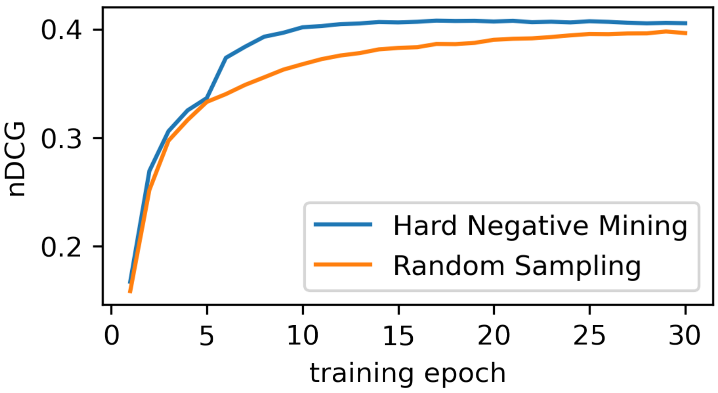


Figure B.1: Comparison of hard-negative mining negatives *vs*. random sampling.

Bibliography

[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 *International Conference on Computer Vision*, 2019.

[2] Arshad Jamal, Vinay P Namboodiri, Dipti Deodhare, and KS Venkatesh. Deep domain adaptation in action space. In *British Machine Vision Conference*, 2018.

[3] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

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

[5] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

[7] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The ”Something Something” video database for learning and evaluating visual common sense. In *International Conference on Computer Vision*, 2017.

[8] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016.

[9] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *IJCV*, 2021.

[11] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[12] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[13] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013.

[14] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and

actions in first person video. In *European Conference on Computer Vision*, pages 619–635, 2018.

[15] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, 2015.

[16] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[17] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[18] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *ArXiv*, 2014.

[19] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, 2016.

[20] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *European Conference on Computer Vision*, 2018.

[21] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *International Conference on Computer Vision*, 2017.

[22] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *International Conference on Computer Vision*, 2019.

[23] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *International Conference on Computer Vision*, 2019.

[24] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *ArXiv*, 2018.

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, 2012.

[26] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *International Conference on Computer Vision*, 2019.

[28] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[29] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *ArXiv*, 2021.

[30] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The EPIC-KITCHENS dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[31] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *European Conference on Computer Vision*, 2012.

[32] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[33] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In Juergen Gall, Peter Gehler, and Bastian Leibe, editors, *German Conference on Pattern Recognition*, 2015.

[34] Yun He, Soma Shirakabe, Yutaka Satoh, and Hirokatsu Kataoka. Human action recognition without human. In *European Conference on Computer Vision Workshop*, 2016.

[35] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *European Conference on Computer Vision*, 2018.

[36] Waqas Sultani and Imran Saleemi. Human action recognition across datasets by foreground-weighted histogram decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[37] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. *AAAI Conference on Artificial Intelligence*, 2020.

[38] Jinwoo Choi, Gaurav Sharma, Manmohan Chandraker, and Jia-Bin Huang. Unsupervised and semi-supervised domain adaptation for action recognition from drones. In *IEEE Winter Conference on Applications of Computer Vision*, 2020.

[39] Michael Wray and Dima Damen. Learning visual actions using multiple verb-only labels. In *BMVC*, 2019.

[40] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[41] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *European Conference on Computer Vision*, 2018.

[42] Xiaolin Song, Sicheng Zhao, Jingyu Yang, Huanjing Yue, Pengfei Xu, Runbo Hu, and Hua Chai. Spatio-temporal contrastive domain adaptation for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[43] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *European Conference on Computer Vision*, 2020.

[44] Mirco Planamente, Chiara Plizzari, Emanuele Alberti, and Barbara Caputo. Cross-domain first person audio-visual action recognition through relative norm alignment. *ArXiv*, 2021.

[45] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 2012.

[46] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 2010.

[47] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2010.

[48] Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems*, 2006.

[49] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *Annual Meeting of the Association of Computational Linguistics*, 2007.

[50] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Von Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems*, 2007.

[51] Arthur Gretton, Alex J Smola, Jiayuan Huang, Marcel Schmittfull, Karsten M Borgwardt, and Bernhard Schöllkopf. Covariate shift by kernel mean matching. In *Dataset shift in machine learning*. The MIT Press, 2009.

[52] Hal Daumé III. Frustratingly easy domain adaptation. In *45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[53] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *International Conference on Computer Vision*, 2011.

[54] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2010.

[55] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *International Conference on Computer Vision*, 2013.

[56] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision*, 2013.

[57] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2016.

[58] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[59] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, pages 120–128, 2006.

[60] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, 2010.

[61] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[62] Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation for discrete optimal transport. In *Advances in Neural Information Processing Systems*, 2016.

[63] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *ArXiv*, 2017.

[64] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2014.

[65] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *International Conference on Computer Vision*, 2013.

[66] Gabriela Csurka, Boris Chidlovskii, and Florent Perronnin. Domain adaptation with a domain specific class means classifier. In *European Conference on Computer Vision*, 2014.

[67] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *International Conference on Machine Learning*, 2011.

[68] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, 2014.

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[70] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision*. Springer, 2006.

[71] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. Domain adaptive neural networks for object recognition. In *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2014.

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

[73] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, 2015.

[74] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning*, 2017.

[75] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. In *International Conference on Learning Representations*, 2017.

[76] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision Workshop*, 2016.

[77] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. In *European Conference on Computer Vision*, 2018.

[78] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[79] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[80] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[81] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, pages 1989–1998, 2018.

[82] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim. Image to image translation for domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[83] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[84] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018.

[85] Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[86] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2018.

[87] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International Conference on Machine Learning*, 2019.

[88] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. VISDA: The visual domain adaptation challenge. *ArXiv*, 2017.

[89] Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *International Conference on Computer Vision*, 2019.

[90] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *International Conference on Learning Representations*, 2018.

[91] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2020.

[92] Barbara Caputo and Novi Patricia. Overview of the imageclef 2014 domain adaptation task. Technical report, University of Rome, 2014.

[93] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[94] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *International Conference on Computer Vision*, 2019.

[95] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning Representations*, 2017.

[96] Slawomir Bak, Peter Carr, and Jean-Francois Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. In *European Conference on Computer Vision*, 2018.

[97] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Instance-guided context rendering for cross-domain person re-identification. In *International Conference on Computer Vision*, 2019.

[98] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[99] Yixiao Ge, Feng Zhu, Rui Zhao, and Hongsheng Li. Structured domain adaptation for unsupervised person re-identification. *ArXiv*, 2020.

[100] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[101] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive Batch Normalization for practical domain adaptation. *Pattern Recognition*, 2018.

[102] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[103] Alessandro Bergamo and Lorenzo Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In *Advances in Neural Information Processing Systems*, 2010.

[104] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, 2017.

[105] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. AudioDIAL: Automatic domain alignment layers. In *International Conference on Computer Vision*, 2017.

[106] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[107] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, 2016.

[108] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *ArXiv*, 2016.

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

[110] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *ArXiv*, 2019.

[111] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, 2020.

[112] Silvia Bucci, Antonio D'Innocente, Yujun Liao, Fabio Maria Carlucci, Barbara Caputo, and Tatiana Tommasi. Self-supervised learning across domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[113] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *International Conference on Computer Vision*, 2019.

[114] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[115] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *European Conference on Computer Vision*, 2018.

[116] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-T approach to unsupervised domain adaptation. *ArXiv*, 2018.

[117] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *International Conference on Computer Vision*, 2019.

[118] Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *International Conference on Learning Representations*, 2020.

[119] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 2004.

[120] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[121] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 2017.

[122] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.

[123] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, 2018.

[124] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, 2016.

[125] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[126] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *International Conference on Computer Vision*, 2019.

[127] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[128] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[129] Weichen Zhang, Dong Xu, Jing Zhang, and Wanli Ouyang. Progressive modality cooperation for multi-modality domain adaptation. *IEEE Transactions on Image Processing*, 30:3293–3306, 2021.

[130] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[131] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, 2016.

[132] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[133] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International Conference on Machine Learning*, 2018.

[134] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[135] Yitong Li, Michael Murias, Samantha Major, Geraldine Dawson, and David Carlson. On target shift in adversarial domain adaptation. In *Proceedings of Machine Learning Research*, 2019.

[136] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *International Conference on Computer Vision*, 2017.

[137] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *European Conference on Computer Vision*, September 2018.

[138] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[139] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self-supervision. In *Advances in Neural Information Processing Systems*, 2020.

[140] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *International Conference Pattern Recognition*, 2004.

[141] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[142] Daniel Weinland, Edmond Boyer, and Remi Ronfard. Action recognition from arbitrary views using 3D exemplars. In *International Conference on Computer Vision*, 2007.

[143] Mikel Rodriguez, Javed Ahmed, and Mubarak Shah. Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[144] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[145] Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[146] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision*, 2011.

[147] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[148] Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine vision and applications*, 2013.

[149] Lucas Smaira, João Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset. *ArXiv*, 2020.

[150] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 2018.

[151] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[152] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *International Conference on Computer Vision*, 2019.

[153] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *German conference on pattern recognition*, 2014.

[154] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *International Conference on Computer Vision*, 2017.

[155] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 2005.

[156] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[157] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *European Conference on Computer Vision*, 2008.

[158] Heng Wang, A. Klaser, C. Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[159] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *International Conference on Computer Vision*, 2013.

[160] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *European Conference on Computer Vision Workshop*, 2004.

[161] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, 2010.

[162] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[163] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*, 2008.

[164] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, 2009.

[165] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*, 2006.

[166] Mihir Jain, Hervé Jégou, and Patrick Bouthemy. Better exploiting motion for better action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[167] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 2016.

[168] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *International Conference on Computer Vision*, 2015.

[169] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[170] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *International Conference on Computer Vision*, 2017.

[171] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[172] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *International Conference on Computer Vision*, 2019.

[173] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[174] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[175] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[176] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *International Conference on Computer Vision*, 2019.

[177] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[178] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks

for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[179] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[180] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 2016.

[181] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[182] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[183] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video Jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[184] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[185] Will Price and Dima Damen. Retro-Actions: Learning 'close' by time-reversing 'open' videos. In *International Conference on Computer Vision Workshop*, 2019.

[186] Pulkit Agrawal, João Carreira, and Jitendra Malik. Learning to see by moving. In *International Conference on Computer Vision*, 2015.

[187] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[188] Mirco Planamente, Andrea Bottino, and Barbara Caputo. Self-supervised joint encoding of motion and appearance for first person action recognition. In *International Conference Pattern Recognition*, 2021.

[189] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *European Conference on Computer Vision*, 2018.

[190] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems*, 2018.

[191] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. AudioSet: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[192] Hossein Rahmani and Ajmal Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[193] Yu Kong, Zhengming Ding, Jun Li, and Yun Fu. Deeply learned view-invariant features for cross-view action recognition. *IEEE Transactions on Image Processing*, 2017.

[194] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 2017.

[195] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Unsupervised learning of view-invariant action representations. In *Advances in Neural Information Processing Systems*, 2018.

[196] Qiang Nie, Jiangliu Wang, Xin Wang, and Yunhui Liu. View-invariant human action recognition based on a 3d bio-constrained skeleton model. *IEEE Transactions on Image Processing*, 2019.

[197] Shruti Vyas, Yogesh S Rawat, and Mubarak Shah. Multi-view action recognition using cross-view video prediction. In *European Conference on Computer Vision*, 2020.

[198] Liangliang Cao, Zicheng Liu, and Thomas S Huang. Cross-dataset action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[199] N Faraji Davar, Teofilo de Campos, David Windridge, Josef Kittler, and William Christmas. Domain adaptation in the context of sport video action recognition. In *Domain Adaptation Workshop, in conjunction with NIPS*, 2011.

[200] Fan Zhu and Ling Shao. Enhancing action recognition by cross-domain dictionary learning. In *British Machine Vision Conference*, 2013.

[201] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision*, 2010.

[202] Junsong Yuan, Zicheng Liu, and Ying Wu. Discriminative video pattern search for efficient action detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[203] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The Jester dataset: A large-scale video dataset of human gestures. In *International Conference on Computer Vision Workshop*, 2019.

[204] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[205] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, 2020.

[206] Meera Hahn, Andrew Silva, and James M Rehg. Action2Vec: A crossmodal embedding approach to action learning. In *British Machine Vision Conference*, 2019.

[207] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *ArXiv*, 2019.

[208] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[209] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *ArXiv*, 2018.

[210] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ACM International Conference on Multimedia Retrieval*, 2018.

[211] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *International Conference on Computer Vision*, 2015.

[212] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[213] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: ClipBERT for video-and-language learning via sparse sampling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[214] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference*, 2018.

[215] Michael Wray, Hazel Doughty, and Dima Damen. On semantic similarity in video retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[216] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, 2010.

[217] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *ArXiv*, 2013.

[218] Ran Xu, Caiming Xiong, Wei Chen, and Jason Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI Conference on Artificial Intelligence*, 2015.

[219] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, 2018.

[220] Andrea Burns, Reuben Tan, Kate Saenko, Stan Sclaroff, and Bryan A Plummer. Language features matter: Effective language representations for vision-language tasks. In *International Conference on Computer Vision*, 2019.

[221] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference*, 2018.

[222] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014.

[223] Hehe Fan, Lian Zheng, Chenggang Yan, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2018.

[224] Yang Fu, Yunchao Wei, Guanshuo Wang, Yuqian Zhou, Honghui Shi, and Thomas S. Huang. Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In *International Conference on Computer Vision*, 2019.

[225] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[226] Xinyu Zhang, Jiewei Cao, Chunhua Shen, and Mingyu You. Self-training with progressive augmentation for unsupervised cross-domain person re-identification. In *International Conference on Computer Vision*, 2019.

[227] Yuxin Peng and Jingze Chi. Unsupervised cross-media retrieval using domain adaptation with scene graph. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[228] Qingchao Chen, Yang Liu, and Samuel Albanie. Mind-the-gap! unsupervised domain adaptation for text-video retrieval. In *AAAI Conference on Artificial Intelligence*, 2021.

[229] Yang Liu, Qingchao Chen, and Samuel Albanie. Adaptive cross-modal prototypes for cross-domain visual-language retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[230] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *European Conference on Computer Vision*, 2018.

[231] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.

[232] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[233] Casey Chu, Kentaro Minami, and Kenji Fukumizu. Smoothness and stability in gans. In *International Conference on Learning Representations*, 2020.

[234] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *International Conference on Computer Vision*, 2019.

[235] Fan Qi, Xiaoshan Yang, and Changsheng Xu. A unified framework for multimodal domain adaptation. In *ACM International Conference on Multimedia*, 2018.

[236] Dang-Khoa Nguyen, Wei-Lun Tseng, and Hong-Han Shuai. Domain-adaptive object detection via uncertainty-aware distribution alignment. In *ACM International Conference on Multimedia*, 2020.

[237] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[238] Albert Gordo and Diane Larlus. Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[239] Mo Zhou, Zhenxing Niu, Le Wang, Zhanning Gao, Qilin Zhang, and Gang Hua. Ladder loss for coherent visual-semantic embedding. In *AAAI Conference on Artificial Intelligence*, 2020.

[240] Kenan E. Ak, Ashraf A. Kassim, Joo Hwee Lim, and Jo Yew Tham. Learning attribute representations with localization for flexible fashion search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[241] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[242] The wikipedia corpus. Accessed:2020-11-11.

[243] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-Exo: Transferring visual representations from third-person to first-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.