Deep Learning based Domain Adaptation

Yusuf Tas

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

November 2021

Declaration

I hereby declare that this submission is my own work (based on publications in collaboration with the co-authors where due acknowledgement is made) and that, to the best of my knowledge, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma at ANU or any other educational institution, except where due acknowledgment has been made. The content of this thesis is mainly based on the publications during my Ph.D. as listed below:

1. Piotr Koniusz*, Yusuf Tas*, Fatih Porikli: "Domain Adaptation by Mixture of Alignments of Second- or Higher-Order Scatter Tensors". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017.
2. Piotr Koniusz*, Yusuf Tas*, Hongguang Zhang, Mehrtash Harandi, Fatih Porikli, and Rui Zhang. "Museum exhibit identification challenge for the supervised domain adaptation and beyond." In Proceedings of the European Conference on Computer Vision (ECCV), pp. 788-804. 2018. (Oral presentation, ∼2% acceptance rate)
3. Yusuf Tas, Piotr Koniusz. "CNN-based action recognition and supervised domain adaptation on 3D body skeletons via kernel feature maps". In British Machine Vision Conference (BMVC) 2018. (Spotlight presentation, ∼6% acceptance rate)
4. Yusuf Tas, Piotr Koniusz. "Simple Dialogue System with AUDITED". Accepted by British Machine Vision Conference (BMVC) 2021.

* indicates shared credit, equal contributions.

Yusuf Tas
19 November 2021

To my wife Ozge Tas

Acknowledgements

First and foremost, I am very grateful to Allah for offering me paths that have led me to where I am now. After that, I would like to express my sincere gratitude to my panel chair Prof. Fatih Porikli, for accepting me to his group in the first place. I am very thankful to you for being immensely helpful in many ways even when you moved away from Australia.

I am equally grateful to my primary supervisor, Dr. Piotr Koniusz. Thank you for giving me valuable pointers and ideas, shaping my research and being part of all my work and research. Thank you for pushing and supporting me mentally whenever required during difficult times. I am indebted to Dr. Piotr for teaching me skills I needed to complete this research and bearing with me in this long journey.

I also would like to thank the other researchers at Data61 and ANU for their feedback, ideas, and discussions we had about research and life. Primarily, I would like to express my gratitude to Dr. Saeed Anwar who has helped me in many ways both academic and personal, Ehab Salahat, Dr. Samitha Herath, Dr. Salman H. Khan, Dr. Zeeshan Hayder, Dr. Arash Shahriari, Fatima and Hongguang.

I acknowledge and appreciate the financial support from the Australian National University, Data61 (previously NICTA) and the Australian Government for my Ph.D. research. Without their scholarships, travel grants and resources, I would not have been able to focus my Ph.D without worrying about my financial situation.

I can not deny the help I have received from Scientific Computing Services at CSIRO. Primarily, I would like to offer my thanks to Ondrej Hlinka and (Tim) Ka Ho for their help with the GPU cluster whenever we needed. Without your help, we would not have managed to run thousands of experiments in this thesis. I also thank NVIDIA for the donation of additional GPUs for our research.

I want to thank to my mother, Songul Tas, for upbringing and looking after me by herself for many years since my father has passed away and to my father, Bedrettin Tas, who unfortunately will not get to see these days. Mom and Dad, thank you for raising me to who I am right now and always supporting my education even when we were struggling financially.

Finally, very special thanks and gratitude to my wife, Ozge Tas. You were always with me in this long path, always believed in me, patiently waited for this day to come. I can never thank you enough for the things you have done for me for years. Thank you for accepting to be a part of my life even if you knew the challenges we would face. I am so lucky to have you, your love and support by my side. This thesis is dedicated to you and our baby who is yet to come.

Abstract

Recent advancements in Deep Learning (DL) have helped researchers achieve fascinating results in various areas of Machine Learning (ML) and Computer Vision (CV). Starting with the innovative approach of [Krizhevsky et al., 2012] where they have utilized processing powers of graphical processing units (GPU) to make training large networks a viable choice in terms of training time, DL has had its place in different ML and CV problems over the years since. Object detection and semantic segmentation [Girshick et al., 2014; Girshick, 2015; Ren et al., 2015], image super-resolution [Dong et al., 2015], action recognition [Simonyan and Zisserman, 2014a] *etc.* are a few examples of that. Over years, many more new and powerful DL architectures have been proposed: VGG [Simonyan and Zisserman, 2014b], GoogleNet [Szegedy et al., 2015], ResNet [He et al., 2016] are examples to most commonly used network architectures in the literature. Our main focus is on the specific task of Supervised Domain Adaptation (SDA) using Deep Learning. SDA is a type of domain adaptation where target and source domains contain annotated, labeled data.

Firstly we look at SDA as a domain alignment problem. We propose a mixture of alignment approach based on second- or higher-order scatter statistics between source and target domains. Although they are different, each class has two distinct representations in source and target domains. The proposed mixture alignment approach reduces within-class scatters to align the same classes from source and target while maintaining between-class separation. We design and construct a two-stream Convolutional Neural Network (CNN). One stream receives source data, and the second gets the target with matching classes to implement the within-class alignment. We achieve end-to-end training of our two-stream network together with alignment losses.

Next, we propose a new dataset called Open Museum Identification Challenge (Open MIC) for SDA research. Office dataset [Saenko et al., 2010] is a very common dataset in SDA literature. But one main drawback of this dataset is that results have started to saturate, reaching 90+% accuracy. The limited number of images is one of the leading causes of high accuracy results. Open MIC aims to provide a large dataset for SDA while providing challenging tasks to be addressed by researchers. We extend our mixture of alignment loss from Frobenius norm distance to Bregman divergences and the Riemannian metric to learn the alignment in different feature spaces.

In the subsequent study, we propose a new representation to encode 3D body skeleton data into texture images using kernel methods for the Action Recognition problem. We utilize

these representations in our SDA two-stream CNN pipeline. We improve our mixture of alignment losses to work with partially overlapping datasets to let us use other Action Recognition datasets as additional source domains even if they only partially overlap with the target set.

Finally, we move to a more challenging domain adaptation problem: Multimodal Conversation Systems. Multimodal Dialogue dataset (MMD) [Saha et al., 2018] provides dialogues between a shopper and retail agent. In these dialogues, the retail agent may also answer with specific retail items such as clothes, shoes *etc*. Hence, the conversation flow is a multimodal setting where utterances can contain both text and image modalities. Two-level RNN encoders are used to encode a given context of utterances. We propose a new approach to this problem by adapting additional data from external domains. For improving the text generating capabilities of the model, we utilize French translation of the target sentences as an extra output target. To improve the model's image ranking capabilities, we use an external dataset and find the nearest neighbors of target positive and negative images. We set up new encoding methods for these nearest neighbors for assigning them to the correct target class, positive or negative.

In summary, we focus on Deep Learning based Supervised Domain Adaptation problems. We have proposed a new approach to domain alignment using class scatter tensors from second or higher-order statistics. We have created a new large dataset for SDA research and demonstrated learning new metrics with our mixture of alignment loss. We have extended our research to Action Recognition and modified mixture of alignment losses to work with any given two domains even if their classes do not fully overlap. Finally, we move to the multimodal conversation. We propose new methods to get and encode additional data from external domains to improve multimodal dialogue agents.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Deep Learning

Machine Learning (ML) is the problem of teaching computers to classify, detect, separate, segment, translate, speak, and many other human-like tasks. ML accommodates various methods and algorithms applicable to many different learning problems: Support Vector Machines (SVM), Artificial Neural Networks (ANN), regression, Bayesian Networks. ANN is constructed from neuron-like units where one neuron connects to neurons of the previous layer and the next layer. Layer by layer structure of the ANN imitates the brain's neural process of feed-forward style where inputs are processed in one group of neurons and forwarded to other neurons connected to it. Learning in ANN is done through gradient backpropagation, where the error is calculated with respect to the target and network output. This error is backpropagated by following the chain rule of derivatives from the output layer back to the first layer. [Rumelhart et al., 1985] introduced this error-propagation idea and proved that Artificial Neural Networks, in other words, Multi-Layer Perceptrons can learn the input representations by updating its internal states with the backpropagated error. [LeCun et al., 1998] showed many years later than its introduction that backpropagation trained networks can be used in a real-life task, document recognition.

ANNs suffered an inconvenient problem at that time, vanishing gradient at early levels of the model. When the error is propagated towards the earlier layers, the gradient will become smaller and smaller due to multiplicative nature of backpropagation. Gradients are used to update the network's weights; ending with near-zero gradients would cause the network to stall and not learn anything. Another major problem of ANNs was that it was not easy to train and required a lot of processing power, preventing researchers from trying out different structures, larger models. It took several years to address these issues and overcome the limitations of ANNs.

Then [Krizhevsky et al., 2012] has started the new era, Deep Learning. They showed that utilizing Graphical Processing Units (GPU) would provide the required processing power to train large networks efficiently. They have demonstrated that Rectified Linear Unit (ReLU) [Nair and Hinton, 2010] with its non-saturating property would make the training several times faster compared to saturating sigmoid activation functions used in traditional neurons. Layers in ANNs are called fully connected layers since each neuron is connected to every other neuron in the previous and the next layer. Considering each connection as a weight to learn, this fully connection nature of ANNs would cause the network to contain many weights it needs to learn. But Convolutional Neural Network (CNN) used in [Krizhevsky et al., 2012] would make use of convolution operation and hence reduce the number of weights in one layer to the size of the convolution filter. With all these improvements, they were able to train a large CNN with five layers on ImageNet classification challenge [Russakovsky et al., 2015] of 1.2 million images within six days. They were able to increase the state of the art results of the time by a large margin, about 11%. This outstanding improvement and the ideas proposed got many researchers' attention, igniting the Deep Learning era in ML literature.

Deep Learning can be essentially defined as the ability to efficiently train large networks on large datasets in reasonable time frames. Over time, deep networks became larger and deeper. [Simonyan and Zisserman, 2014b] from 5 layered structure of [Krizhevsky et al., 2012] and proposed new models with 16 and 19 layers. Then [Szegedy et al., 2015] was able to train networks with 22 layers. [He et al., 2016] took the word *"deep"* to another level and trained networks with even more than 100 layers deep. Over time, as predicted by [Krizhevsky et al., 2012], achieving better results for image classification was inevitable with better and powerful GPU and deeper models.

Although it was introduced in CNN for the image classification task, many more deep architectures were designed and applied to various problems in ML research. R-CNN and its subsequent iterations achieved impressive results on object detection and semantic segmentation tasks [Girshick et al., 2014; Girshick, 2015; Ren et al., 2015]. [Dong et al., 2015] utilized a CNN-based model in image super-resolution task to generate a higher resolution image from the given lower resolution one and achieved much better results than the traditional bicubic interpolation based image upscaling. [Simonyan and Zisserman, 2014a] proposed a two-stream network-based model for the action classification task.

Achievements of Deep Learning were limited to image-related problems. It was not unusual to see Deep Learning types of approaches used in any area of Machine Learning. Long Short-Term Memory (LSTM) type of recurrent networks was used by [Sutskever et al., 2014] to model and learn sequence to sequence nature of automated language translation task. Embeddings from Language Models (ELMo) [Peters et al., 2018] and Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018] revolutionized natural language processing literature by providing powerful models that can extract text-based features and achieve state of the art results in various text-related tasks. Deep Recurrent Neural Networks (RNN) was utilized in [Graves et al., 2013] for speech recognition from audio signals.

Deep Learning's successful applications in various ML fields are also related to easy access to readily available GPU-based implementations. Since its introduction, various frameworks have provided efficient GPU-based implementation of core elements such as convolution layers, pooling, activation functions, and the ability to construct and train models easily. This ease of prototyping and rapid development helped researchers model and develop new architectures for various problems smoothly. To name a few of these: Theano [Al-Rfou et al., 2016], Tensorflow [Abadi et al., 2015], Pytorch [Paszke et al., 2019], Matconvnet [Vedaldi and Lenc, 2015], Caffe [Jia et al., 2014] *etc*. provided easy to use frameworks in various programming languages. One advantage found in Theano, Tensorflow, Torch, and Pytorch was implementing an auto differentiation tool in the framework. Every operation in a network requires gradient calculations for its input and weights to apply backpropagation. Auto differentiation helped researchers skip this load and automatically calculate any new operation's gradients as long as it is within the boundaries of the framework.

A simple illustration of a CNN can be found in Figure 1.1. The figure demonstrates a couple of the most used layer types found in CNN: convolution, pooling, and fully connected layers. These layers can be found in many different state of the art models. These layers can be explained briefly as follows:

1. **Convolution layer:** A predetermined number of randomly and differently initialized 2d kernels convolved over each channel of the image to calculate the output response of these filters, kernels. Randomly initialized starting points of these filters let them explore different feature spaces and learn interesting various shape structures. Generally, early convolutional layers' kernels tend to learn edge and corner like basic shape defining features, but in later and deeper levels, they learn more sophisticated shapes.

2. **Pooling Layer:** Pooling is the operation of downsampling of the input feature map. This operation reduces the spatial size of the feature maps, which consecutively reduces the total number of operations required in the following layers to counter overfitting. In literature, various types of downsampling operations are used in pooling layers, maximum, average, global average *etc*., while the max-pooling is the most common one.

3. **Fully Connected Layers:** Fully connected layers are the same layers used in classic ANNs. Every output neuron is connected to each of every input neuron. It has been proven that at least one fully connected layer is a universal function approximator [Cybenko, 1989]. This feature allows the final fully-connected layers to learn task-specific procedures from the received feature maps, for example, a classifier for the image recognition task or a regression function for the image detection.

Deep Learning literature contains many more layer types for different problems and tasks. For example, loss layers are the part where the error is calculated for backpropagation. They are task-specific functions for different kinds of problems. Similarly, many different layers are specific to their field of application. They are out of the scope of this brief introduction chapter. In the following chapters, any particular layer type will be analyzed and discussed as required.

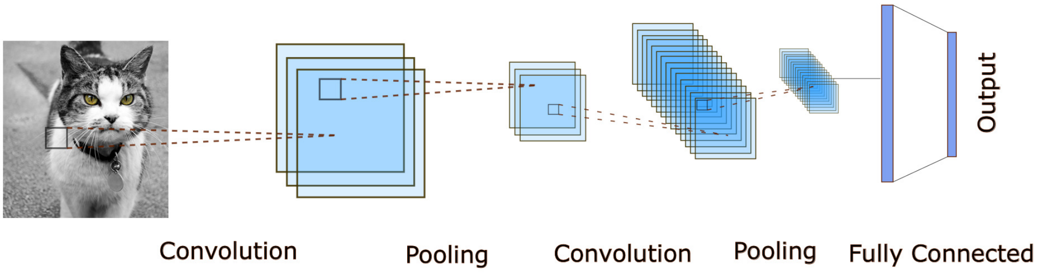


Figure 1.1: Simple illustration of a CNN, demonstrating general layers types used in the construction of the models.

Data used in deep learning comes in many shapes and forms. For example, simple RGB images, illustrations, depth images, thermal images, sentences, movie scripts, conversations, videos, body skeleton scans, voice, and many other types exist in the literature. Even for RGB images, there are many different ways to collect them, from professional cameras to drawings. Between two collections of images, there can be many different conditions such as lightning, blur, background, noise levels, environmental conditions *etc*. Classical ML and deep learning methods might suffer from these differences, which might be summarized as follows:

- Same collection method but different conditions such as lightning, day or night *etc*.
- Same conditions but collected through different tools, for example, webcams, professional cameras, web images.
- Same object or class is represented through uncorrelated methods, for example, paintings and real images or text descriptions of an image and the image itself.

Domain adaptation addresses these challenges where classical methods fail. It aims to bring two different data collections together and learn a shared space where ML can achieve better results. The following section will give an overview of Domain Adaptation, current methodology, and our proposed solutions to several problems in its current state.

1.2 Domain Adaptation

In this current age, millions of images are collected, produced, created, and stored every day through different methods and tools all around the world. For any chosen image category, either specific or broad such as animals or American Wirehair cats, its image representation can be found in many different types, including, but not limited to digital, painting, thermal imaging, x-ray imaging *etc*. There are even more sub-categories within any of these representations; for example, painting of cats can have different styles such as modern, surrealistic, abstract,

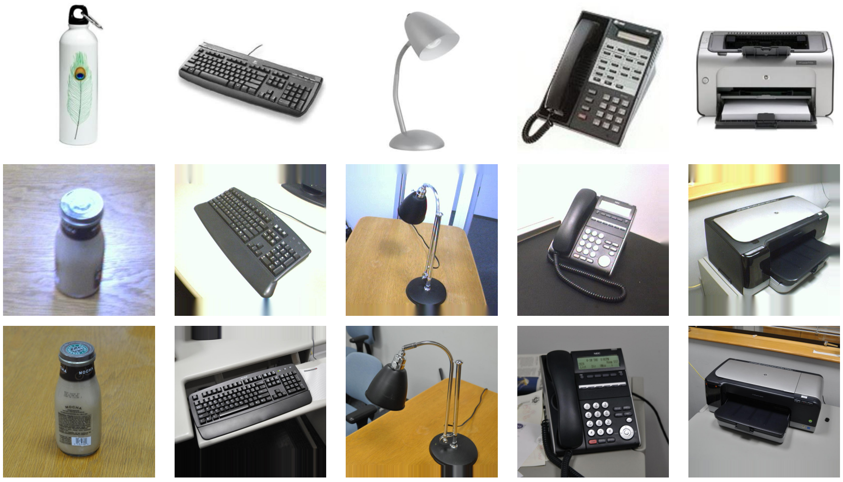


Figure 1.2: Top, middle and bottom rows show examples from the Amazon, DSLR, and Webcam domains from the Office dataset [Saenko et al., 2010]

or other source materials such as oil, acrylic, watercolor, or pencil. These mediums where the images of one category are represented through different means is what we call *"Domain"*. A domain defines a generic approach, tool, or methodology of creating or defining an image. For example, Digital Single Lens Reflex (DSLR) photography represents the domain where the imaging device is DSLR cameras. Similarly, webcam photography would be the domain of images where the imagery tool is a webcam.

Different domains create different representations of the same category images where each domain would contain different statistical information. For example, even if the result would look similar to the human eye for DSLR and webcam images, they would still have a different range of pixel values, noise levels, mean *etc*. Dissimilarity would increase further if we compare disconnected mediums such as DSLR photography against painting.

Figure 1.2 shows 5 different objects from 3 different domains. Images are from the Office dataset [Saenko et al., 2010] which provides images for daily office objects from 3 different domain sources: Amazon product images, DSLR images, and webcam images. In the figure, the first row images are from the Amazon domain, the middle ones are from the webcam domain, and the last row images are from the DSLR domain from the dataset. It is evident in the figure that one object category can be represented differently in different domains. For example, while Amazon products are depicted on a white background, the other two domains' images include real-world background representations in the image. Even when comparing webcam and DSLR images, the difference between these domains is apparent in images: light reflection in webcam images, quality difference *etc*.

In ML methods where the computer needs to learn from the given data, these different representations provide a diversified knowledge spectrum. Domain adaptation aims to improve the learning in a given target domain using the knowledge from source and target domains

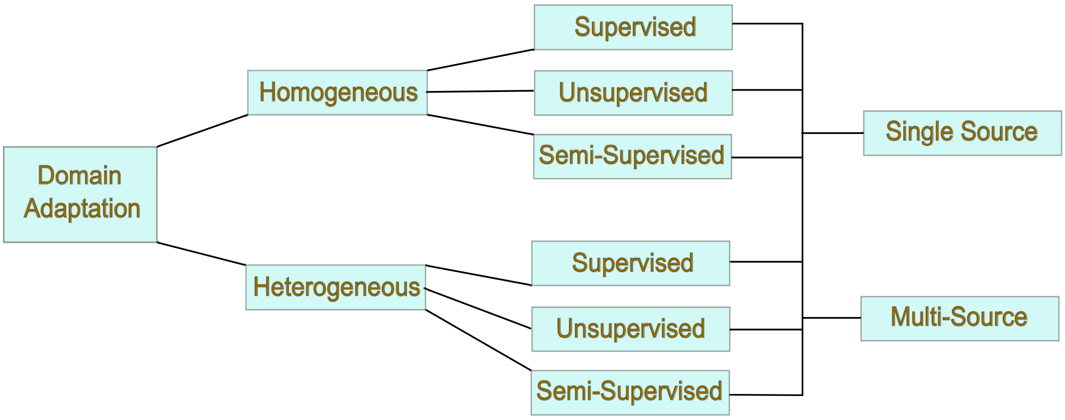


Figure 1.3: Simple illustration of Domain adaptation method categories.

together. It is a sub-task of ML where both supervised and unsupervised methods can be used. Learning here refers to the specific problem of ML, for example, the image recognition problem. For the image recognition problem, one domain could contain DSLR images of animals, and another domain could be their paintings. Although statistically and distribution-wise different, these two other domains provide information about the object of interest from different views. Domain Adaptation (DA) is the ability to utilize images from various domains to learn the task of interest jointly.

DA defines two terms to refer to different domains: source and target. The target domain is the target data where the trained models need to achieve better accuracy, ranking, or any other metric. The model aims to learn the target distribution. On the other hand, the source is the domain that provides a different distribution for training to help learning in the target domain. Source provides a different perspective on the given target data. DA aims to utilize source domain along with target to understand target distribution better, even though they are separate domains with different distributions.

Depending on the problem and the data available for that specific problem, DA can be done through different ML methodologies supervised, unsupervised, or semi-supervised learning. In Supervised Domain Adaptation (SDA), both source and target domains have labeled data, meaning that the samples' classes and categories are known during training time. In contrast, for Unsupervised Domain Adaptation (UDA), target samples' labels are unknown during training while the source domain still contains labeled data. Compared to SDA, UDA constitutes a more complex problem, as the target model's labels will not be known, and the knowledge to understand and align clusters of target data will be learned from source data only. Semi-supervised DA (SSDA), on the other hand, will include a small set of target samples with labels available and the rest of the target samples without labels. Our primary focus will be on SDA and, in particular Deep Learning based SDA.

Domain adaptation is available in many different types. Depending on the number of sources used, there are single or multiple-source DA methods available. While single-source DA uses one source domain with the target, multiple source DA will utilize several source domains together, for example, images in different lighting conditions, different backgrounds *etc*. Depending on the nature of the data and discrepancy between source and target domains, it can be named homogeneous or heterogeneous DA [Zhang et al., 2019]. In homogeneous DA, source and target share similar feature space and thus same feature dimensions; for example, source and target domain in the Office dataset [Saenko et al., 2010] form a homogeneous DA problem where both domains are images. For heterogeneous DA, target and source do not share a common feature space, such as the source contains images while the target domain is text. Summary of DA types can be seen in Figure 1.3.

1.2.1 Deep Domain Adaptation

In the literature, various solutions and approaches exist for DA problems. Depending on the proposed solution, we can categorize DA into shallow and deep categories. Shallow DA can be further divided into two sub-classes: instance-based and feature-based [Wang and Deng, 2018]. Instance-based DA trains models on weighted source domain samples to learn and adjust the domain shift between source and target domains [Xu et al., 2018], [Chu et al., 2013]. On the other hand, feature-based DA aims to learn a common feature space where the source and target domains are mapped into to address domain shift problem [Pan et al., 2010], [Gong et al., 2013], [Gheisari and Baghshah, 2015].

In recent years, deep DA solutions have been proposed to tackle domain adaptation problems, where deep learning-based networks are used. [Tzeng et al., 2015] presents a new CNN architecture to utilize well-labeled source data to transfer knowledge to sparsely labeled training data by forcing the target network to output activations similar to those in the source network. [Sun and Saenko, 2016] proposes a correlation alignment loss for unsupervised DA where the target is unlabelled. This loss aims to learn target distribution from the second-order statistics of the source domain. [Rozantsev et al., 2018] utilizes a two-stream network with weights not being shared between the layers while also adding regularization loss between each related layer to control their difference spread.

We regard the SDA problem as a type of alignment problem with multiple domains with multiple categories. SDA requires both learning class-wise alignment to understand the separation of each class and the alignment of domains to better adapt the data from the source domain into the target domain. Figure 1.4 shows a simple illustration of the domain alignment problem. Three classes are shown from two different domains, desired outcome shown on the right image. Domains should be aligned such that the same classes have similar distributions in both domains, but they should still be separable with the shown separation boundaries.

For tackling this domain alignment problem, we propose a novel mixture of alignment loss based on second and higher-order scatter tensors. Pairwise class statistics are used to bring the

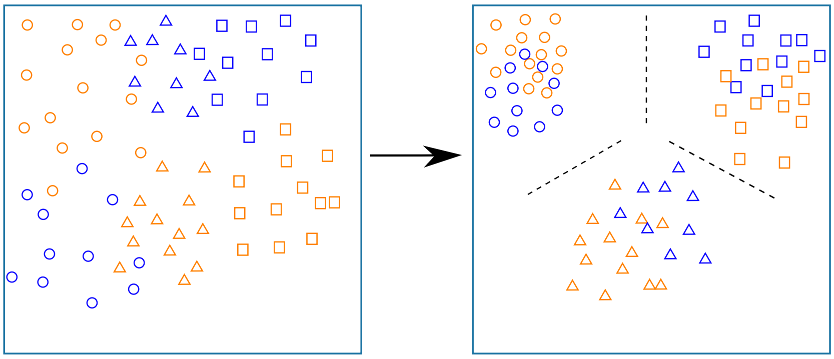


Figure 1.4: Simplified representation of domain alignment problem. Blue-colored shapes represent one domain, and orange is another domain. Circle, triangle, and rectangles represent different classes available in the shown domains. The image on the left shows unaligned domains, and the image on the right shows the desired alignment outcome. Dashed lines indicate the separation boundaries between the classes.

same classes together. Given that each class might benefit from different levels of alignment, we further control class-wise alignment with a trainable weight for each class. Further details of our proposed mixture of alignment loss and results can be found in Chapter 2.

There are several benchmarking datasets available in the literature. A commonly used example, Office dataset [Saenko et al., 2010], provides images of 31 different daily office objects in three other domains: Amazon, DSLR, and webcam. DSLR and webcam domains include photos taken with DSLR and webcam cameras, respectively, while images in the Amazon domain are on white backgrounds, object-only images. Office dataset contains around 4100 images in total for three given domains. Sample images from this dataset where the same class images are shown for all three domains can be seen in Figure 1.2. Although it is widely used as a benchmark dataset, it contains several downsides. In terms of complexity, it lacks challenging tasks, and the domain shift is visually tiny, for example, DSLR to webcam. Lack of complexity results in saturated accuracy results passing 90% line. Also, for deep DA, the total number of images per domain is low compared to general deep learning datasets. NYU Depth Dataset [Nathan Silberman and Fergus, 2012] provides video sequences of indoor scenes recorded with both RGB and depth cameras. Although it offers many unlabelled video frames, there are 1449 annotated RGB and depth image pairs.

To fill in these missing points for general SDA tasks, we propose Open Museum Identification Challenge (Open MIC). Dataset is created from 10 distinct exhibitions of paintings, timepieces, sculptures, glassware, relics, science exhibits, natural history pieces, ceramics, pottery, tools, and indigenous crafts. Open MIC contains images of 866 unique exhibit instances with annotated 8560 source domain images and 7596 target domain images along with 380k unlabelled video frames. While a large number of annotated target and source samples let the dataset be used in SDA, unlabelled video frames can still be used in UDA or SSDA methods.

Each art piece in the dataset is both captured with a mobile phone and a wearable cam-

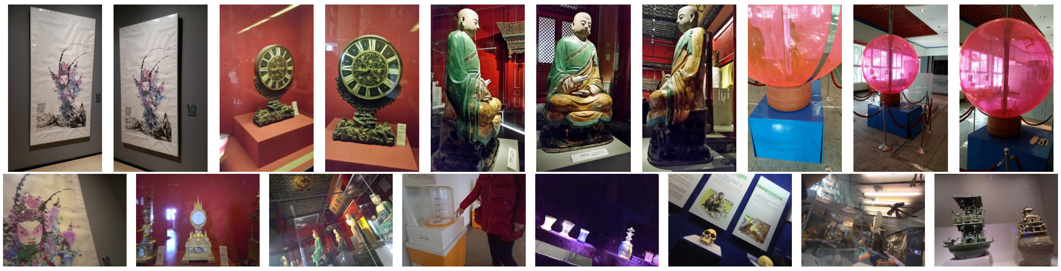


Figure 1.5: Examples of the source and target subsets of Open MIC. Top row includes Paintings (*Shn*), Clocks (*Shg*), Sculptures (*Scl*) and Science Exhibits (*Sci*). As the images per exhibit demonstrate, different viewpoints and scales are covered during capturing. Bottom row shows samples images from target subsets of Paintings (*Shn*), Clocks (*Shg*), Sculptures (*Scl*), Science Exhibits (*Sci*) and Glasswork (*Gls*), Cultural Relics (*Rel*), Natural History Exhibits (*Nat*), (*Shx*), Porcelain (*Clv*) exhibits. Due to capturing through wearable cameras, variety of photometric and geometric distortions can be seen in the images.

era. The difference in capturing method creates challenging problems for domain adaptation research such as quality of lightning, motion blur, occlusions and clutter, rotations, glare, transparency, non-planarity, clipping *etc*. Example images from source and target sets can be seen in Figure 1.5 demonstrating various challenges and distortions available in the dataset.

1.2.2 Action Recognition

Action recognition is the task of differentiating different action categories, for example, running, jumping, handshaking, pulling *etc*. For action recognition, due to its nature of sequential movements, generally, videos are used as the primary source. However, in recent years, RGB-D-based sources have become popular with the easy recording capabilities of Kinect sensors on 3D human skeleton body joints [Wang et al., 2018]. Since these joints' movement and their relation to each other in 3D space define an action, they are essential in understanding the activity.

Action recognition literature contains various datasets such as NTU RGB+D [Shahroudy et al., 2016], SBU-Kinect-Interaction [Yun et al., 2012], KTH [Schuldt et al., 2004], UTKinect-Action3D [Xia et al., 2012] and many more others where they offer recordings of actions in similar mediums, for example, videos or sequences of human body joints. A quick look at these datasets reveals that the range of action classes are rather small, starting from 10 and reaching 120 in NTU RGB+D [Shahroudy et al., 2016] which is one of the largest datasets for human action recognition. Given that the number of actions is rather small in those datasets, it is inevitable that these action classes would overlap between different datasets. Also, recordings of actions are generally very similar, either using an RGBD camera to record videos and depth or using Kinect-like hardware to record body skeletons with joint locations. This similarity between the datasets inspired us to utilize domain adaptation. We can combine multiple datasets to jointly train a supervised model and use overlapping classes between different datasets.

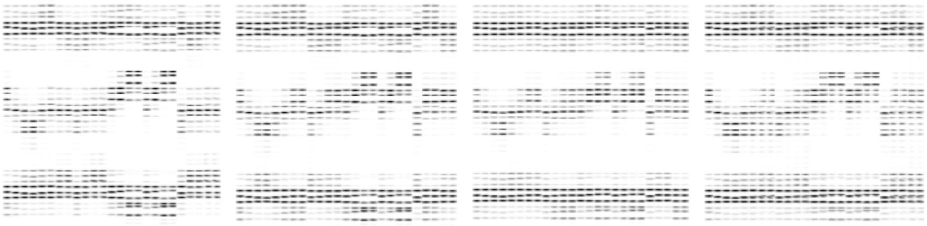


Figure 1.6: Four texture maps of 4 different actions. Note the subtle differences

Combining multiple datasets allows us to exploit the large-scale action recognition datasets to improve classifiers trained on small-scale datasets. The large-scale counterpart offers much more data for the overlapping classes than the small-scale dataset.

Early approaches to action recognition were more focused on handcrafted temporal features on time sequences such as temporal-templates tracking the motion of spatial locations [Bobick and Davis, 2001], histograms of oriented 3D spatio-temporal gradients [Klaser et al., 2008], spatial interest points in the spatio-temporal domain [Laptev, 2005]. However, with the advancements in DL, it was inevitable that deep learning based action recognition achieved much better results. [Simonyan and Zisserman, 2014a] uses RGB images together with the optical flow to feed a two-stream-based CNN. 3D convolutional networks are used to address the time domain in videos [Ji et al., 2012]. [Karpathy et al., 2014] utilizes CNN for large-scale video classification.

We propose a novel method for encoding sequences of these joints to make them usable in traditional CNN. Traditionally inputs to CNN consists of images in the spatial domain where there is no relation to time. 3D body joints sequence represents the body joints in 3D coordinates in the spatial domain, and its frame by frame sequence is the temporal domain. We combine spatial and temporal domains into one 2D CNN texture-map like inputs derived based on kernel methods [Scholkopf and Smola, 2001]. We describe our methodology and how we utilize kernel-based derivation in more detail in Chapter 4. Examples of the generated texture maps which will be fed to CNN can be seen in Figure 1.6

We further propose a domain adaptation based strategy to the action recognition problem. We leverage Kinect-based data available in other datasets as a source domain to improve performance on the target domain. Our texture maps allow us to easily integrate our mixture of alignments loss to action recognition problems. We extend our alignment loss to work with datasets that partially overlap in the available classes of source and target domains.

1.2.3 Multimodal Conversation

With the advancements and improvements in deep learning achieving outstanding results in various problems and tasks, attention has been shifting towards multimodal problems in deep learning literature in recent years. Multimodal problems are the tasks that require learning from multiple separate domains, such as image captioning [Xu et al., 2015], question answering from videos [Zeng et al., 2017]. This task involves domain adaptation types of approaches to learning a shared space between these separate domains.

The multimodal conversation system is another example of multimodality-based input. It consists of utterances where they contain multimodal data. For example, a dialogue between an online retail assistant and a shopper will include images, sentences, and maybe even a voice if the retail assistant can talk. This heterogeneous nature of multimodal conversations makes it a challenging task to learn for ML methods.

Multimodal Dialogue dataset (MMD) [Saha et al., 2018] provides a large set of multimodal conversations. It imitates the dialogue between a shopper and a retail agent with a semi-automated process. Utterances in dialogues can contain a sentence, clothing images, or both. In the dialogues, it can be observed that images are referred to in text through different indicators such as index numbers first, second *etc*. or conjunctions after, before, next *etc*. Capturing these fine details and understanding the relationship between the images and text is the challenge to address. The authors propose two benchmark tasks on the dataset, image ranking, and next sentence predictions. These tasks are named image task and text task in short. Each task requires a different approach and methodology as the target modality is different for each task.

In Chapter 5, we propose utilizing external data and adapting knowledge through assisted supervision to improve results on image and text tasks. We translate target sentences to French in the text task to capture another language's intrinsic structure to learn English targets better. We propose an embedding method to search and find nearest neighbor images of target images to better understand ranking through external knowledge for the image task.

1.3 Contributions

Our contributions to Deep Learning based Domain Adaptation are listed as follows:

- We formulate a novel mixture alignment loss based on second and higher-order class scatter tensors for SDA. We provide a fast kernelized version of this loss and its derivatives, making its use tractable in deep learning. SDA end-to-end training of non-euclidean Jensen-Bregman LogDet Divergence (JBLD) and Affine-Invariant Riemannian Metric (AIRM) distances used at this mixture of alignment loss. We demonstrate how we make this non-euclidean distance-based training tractable using Nystrom projections.

- We propose a novel method for encoding 3D body skeleton joint sequences into a texture-like feature map representations based on kernel methods. To the best of our knowledge, we are the first to adapt SDA to action recognition from sequences of 3D skeleton joints. We extend our mixture of alignment loss by making it work with partially overlapping target and source domains. This overlapping helps us utilize various action recognition datasets as source domains where they would only partially overlap with the target domain.

- For the Multimodal Dialogue (MMD) problem, we propose a novel method to incorporate external images through soft and hard assigned nearest neighbor embeddings (MMD-Neha and MMD-Nesa). This method helps the network in image ranking tasks by assisting with external knowledge. For the MMD problem, we propose utilizing French translation of sentences in assisted supervision to improve the network's understanding of sentences through the structure of another language to generate better sentence predictions.

- We collect and annotate a new challenging dataset, Open MIC, for domain adaptation research, consisting of museum exhibit images. By using Android phones and wearable cameras to capture the photos of exhibits, we create two domains for DA problems. The latter will create distortion-like challenges due to the nature of the capturing method. We provide extensive baseline results, evaluation protocols, statistics for the Open MIC dataset.

1.4 Thesis Outline

The rest of the thesis is structured into chapters with respect to listed contributions as follows.

In Chapter 2, we start our work on supervised domain adaptation. We look into previous SDA methods and discuss that if the domain alignment is executed class-wise, it can improve the performance. We propose a novel alignment loss based on the mixture of alignments of second- or higher-order scatter statistics between the source and target domains, which aims to bring the same classes' distributions in source and target domains close to each other while also maintaining separability of different classes. This loss is used as a connection bridge between source and target at the end of a two-stream CNN, trained in an end-to-end fashion. We demonstrate achieving higher than the state of the art results in several DA datasets.

In Chapter 3, we address the limitations of datasets in SDA literature where results have reached saturation, and the number of images is limited for deep learning based networks. We demonstrate a new dataset named Open Museum Identification Challenge to address these problems. We discuss the details of our new dataset, how it is collected, available domains, how domain adaptation can be applied to it, and which challenges it poses. We propose an extension to our alignment loss by using end-to-end Bregman divergences and the Riemannian metric and how we make this training with non-euclidean distances tractable in deep networks. We present extensive ablation studies to provide baseline results in our dataset for each DA split and a demonstration of how it can also be used for unsupervised domain adaptation.

In Chapter 4, we move to action recognition problem. We propose a new representation by encoding sequences of 3D body skeleton joints in texture-like representations based on kernel methods. We demonstrate how we utilize these representations in traditional CNN, enabling us to apply our SDA methods to the action recognition problem. To the best of our knowledge, we are the first to adapt SDA to the action recognition on time sequences of 3D body skeleton joints. We extend our mixture of alignments loss to work with datasets which class concepts match partially.

In Chapter 5, we change our focus to a more challenging problem, multimodal conversation systems. We work on the Multimodal Dialogue (MMD) dataset [Saha et al., 2018] which provides multimodal dialogues between a shopper and the retail agent. We propose a novel assisted supervision method by leveraging external datasets through nearest-neighbor embeddings for image tasks. We formulate our nearest neighbor embedding method and demonstrate higher results than state-of-the-art. For the text task, we propose a novel assisted supervision by utilizing French translations as additional target output of the network. This approach helps the network exploit a different language's structure in training. We demonstrate our results on the MMD dataset.

And finally, in Chapter 6, we finalize the thesis by discussing what we've achieved and what future work could be taken based on our findings in this thesis.

Chapter 2

Domain Adaptation by Mixture of Alignments of Second- or Higher-Order Scatter Tensors

2.1 Summary

In this chapter, we propose an approach to the domain adaptation, dubbed Second- or Higher-order Transfer of Knowledge (So-HoT), based on the mixture of alignments of second- or higher-order scatter statistics between the source and target domains. The human ability to learn from few labeled samples is a recurring motivation in the literature for domain adaptation. Towards this end, we investigate the supervised target scenario for which few labeled target training samples per category exist. Specifically, we utilize two CNN streams: the source and target networks fused at the classifier level. Features from the fully connected layers fc7 of each network are used to compute second- or even higher-order scatter tensors; one per network stream per class. As the source and target distributions are somewhat different despite being related, we align the scatters of the two network streams of the same class (within-class scatters) to a desired degree with our bespoke loss while maintaining good separation of the between-class scatters. We train the entire network in end-to-end fashion. We provide evaluations on the standard Office benchmark (visual domains), RGB-D combined with Caltech256 (depth-to-rgb transfer) and Pascal VOC2007 combined with the TU Berlin dataset (image-to-sketch transfer). We attain state-of-the-art results.

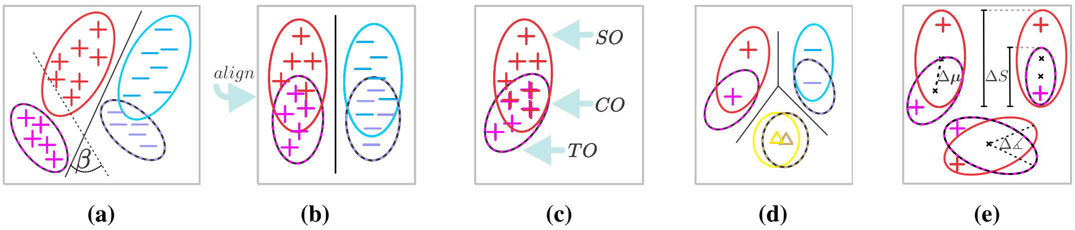
We start our research by tackling domain adaptation as an alignment problem. As discussed in the previous chapter, state of the art methods did not utilize supervised class-wise alignment in domain adaptation tasks. We address this by formulating an approach to align matching classes between two domains based on second- or higher-order class scatter tensors. This chapter has been published as a conference paper: "Piotr Koniusz*, Yusuf Tas*, Fatih Porikli: Domain Adaptation by Mixture of Alignments of Second- or Higher-Order Scatter Tensors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

2017.". * indicates shared credit, equal contributions.

2.2 Introduction

Domain adaptation and transfer learning are the problems widely studied in computer vision and machine learning communities [Baxter et al., 1995; Li et al., 2016]. They are directly inspired by the human cognitive abilities of generalizing to new concepts from very few data samples (cf. training from scratch on over a million of labeled images of the ImageNet dataset [Russakovsky et al., 2015]). From psychological point of view, transfer of learning is "*the dependency of human conduct, learning or performance on prior experience"*. This problem was introduced in 1901 under a notion of "*transfer of particle*" [Woodworth and Thorndike, 1901]. In machine learning, transfer learning (or inductive learning) concerns *"storing knowledge gained while solving one problem and applying it to a different but related problem"* [West et al., 2007]. In practical computer vision and machine learning systems, transfer learning refers to *"an ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonality"*. In general, given a new (target) task, the arising question is how to identify the commonality between this task and previous (source) tasks, and transfer knowledge from the previous tasks to the target one. Therefore, one has to address three questions: what to transfer, how to transfer, and when to transfer [Tommasi et al., 2010].

In what follows, we propose an approach to the domain adaptation, dubbed Second- or Higher-order Transfer of Knowledge (*So-HoT*), based on the mixture of alignments of second- and/or higher-order scatter statistics between the source and target domains. Specifically, we utilize second- and/or higher-order scatter tensors, one per each network stream per class, such that the first stream corresponds to the source domain while the second to the target. The scatters are built from the feature vectors produced by the *fc7* layer of AlexNet [Krizhevsky et al., 2012]. We propose that, as the source and target distributions are only partially related by their commonality, the scatters of the same class from both streams (*within-class scatters*) should be aligned to a desired degree to capture this commonality as an overlap between parts of the two distributions. At the same time, to achieve high classification accuracy, we maintain good separation between the scatters representing different classes (*between-class scatters*). We devise a simple loss that brings each pair of within-class scatters closer in terms of their covariances as well as their corresponding means. Therefore, the CNN parameters stored by convolutional filters and weights of the target network regularized by the source data in this end-to-end fashion must produce statistics consistent with the source network. We view such a regularization paradigm as being motivated by the theory of privileged learning [Vapnik and Vashist, 2009]. In our case, the statistics of the source network regularize the target (and vice-



(a) (b) (c) (d) (e)

Figure 2.1: Our alignment problem. In Figure 2.1a, a two-class toy problem with positive and negative samples $(+)$ and $(-)$ is given. The solid and dashed ellipses indicate the source and target domain distributions. The two hyperplane lines that separate $(+)$ from $(-)$ on the target data indicate large uncertainty (denoted as β) in the optimal orientation for the target problem. Figure 2.1b shows that the source and target distributions can be aligned enough to separate well two classes for both the source and target problems. Figure 2.1c shows that partially aligned distributions have the commonality (CO) as well as the source and target specific parts (SO) and (TO) that represent dissimilarity between the source and target. Figure 2.1d depicts a multi-class problem. Beside of partially aligned means, the orientations of the source and target distributions are allowed to partially differ – as a result, they *i.e.* fit better into the piece-wise linear decision boundary. Figure 2.1e shows that differences in means $\Delta\mu$, scale/shear ΔS and orientation $\Delta\measuredangle$ of within-class scatters are all part of the alignment process.

versa) whilst in the privileged learning, the side information regularizes the solution dictated by the empirical loss evaluated on the main data samples. See Figures 2.1 and 2.2 for illustrative examples.

Furthermore, as distributions of the source and target domains may require different level of alignment per class (the commonality depends on the class label), we investigate not only an unweighted alignment loss (class-independent level of alignment) but also its weighted counterpart which learns one weight per class (class-specific levels of alignment).

Additionally, as we work with second- and/or higher-order tensors, we propose a kernelized variant of our alignment loss which provides computational speed-ups for typical domain adaptation datasets.

To summarize, our main contributions are: i) a novel loss that we call *So-HoT*, which defines the commonality between the source and target domains as the mixture of alignments of second- and/or higher-order scatter tensors, ii) unweighted and weighted variants of alignments, and iii) a fast kernelized alternative of our alignment loss.

Next, we detail the notion of domain adaptation and transfer learning, review the related literature and explain how our work differs from the state-of-the-art approaches.

2.3 Related Work

Domain adaptation assumes that the transfer of knowledge takes place among two or more distinct domains *e.g.*, e-commerce reviews and biomedical data. In contrast, transfer learning utilizes the same domain *e.g.*, images of natural scenes with related but different distributions where the goal may be to learn objects of a new class while leveraging other already learned classes [Thrun, 1996; Tommasi et al., 2010]. Not surprisingly, these both notions are often interchangeable *e.g.*, natural images and sketches have related distributions but they come from distinct domains at the same time. Another example is a so-called domain shift *e.g.*, bicycle in natural images vs. on-line retailer galleries. Transfer of knowledge may vary from simply carrying over discriminative information from a source to target domain under the same set of classes to inferring a solution to a new distinct task from a set of former ones [Thrun, 1996; Intrator and Edelman, 1996]. Domain adaptation comes in many flavors. Single- or multiple-source [Crammer et al., 2008] setups are possible *e.g.*, single stream of natural images vs. multiple streams supplied with photos of objects: on cluttered backgrounds, on a clear background, in a daytime or night setting, or even in multi-spectral setting. Moreover, the problem in hand may be homogeneous or heterogeneous [Tommasi et al., 2010; Yeh et al., 2014] in nature *e.g.*, identical source and target representations using RGB images vs. a source represented by a CNN trained on images [Fukushima, 1980; Krizhevsky et al., 2012] and a target using an LSTM [Hopfield, 1982; Hochreiter and Schmidhuber, 1997] which is trained on text corpora or video data [Herath et al., 2017b]. The architecture in use may be shallow [Daumé III et al., 2010; Sun et al., 2016] or deep [Ganin et al., 2016] so that the commonality is established only at the classifier level or across entire source and target networks, respectively. Noteworthy is also recent trend in the CNN fine-tuning which by itself is a powerful domain adaptation and transfer learning tool [Girshick et al., 2014; Sermanet et al., 2013] which requires large training datasets. Moreover, domain adaptation and transfer learning address problems such as: learning new categories from few annotated samples (supervised domain adaptation [Chopra et al., 2013; Tzeng et al., 2015]), utilizing available unlabeled data (unsupervised [Sun et al., 2016; Ganin et al., 2016; Herath et al., 2017b] or semi-supervised domain adaptation [Daumé III et al., 2010; Tzeng et al., 2015]), recognizing new categories in embedded spaces *e.g.*, attribute-based, without any training samples (zero-shot learning [Fei-Fei et al., 2006]).

In this chapter, we investigate the case of a deep supervised single-source domain adaptation which can be easily extended to the multi-source and heterogeneous cases.

The Commonality. Deep learning [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014b; Harandi and Fernando, 2016] has been used in the context of domain adaptation in recent works *e.g.*, [Tzeng et al., 2015; Ganin et al., 2016; Chopra et al., 2013; Wang and Hebert, 2016; Kuzborskij et al., 2016; Tommasi et al., 2016; Long et al., 2015]. These works differ in how they establish the so-called commonality between domains. In [Tzeng et al., 2015], the authors propose to align both domains via the cross entropy which “maximally confuses” both domains for supervised and semi-supervised settings. In [Ganin et al., 2016], an unsupervised approach utilizes the assumption that predictions must be made based on features which cannot discriminate between the source and target domains. Specifically, they minimize a trade-off between the so-called source risk and the empirical divergence to find examples in the source domain indistinguishable from the target samples.

Our work differs from these methods in that we define the commonality as the desired degree of overlap between the second- and/or higher-order scatters of the source and target. After such an alignment, we let the non-overlapping tails of distributions also guide learning which results in a more general classifier (*i.e.* avoid the domain-specific bias).

Moreover, in [Chopra et al., 2013], the authors capture the "interpolating path" between the source and target domains using linear projections into a low-dimensional subspace which lies on the Grassman manifold. In [Wang and Hebert, 2016], the authors propose to learn the transformation between the source and target by the deep model regression network. These two approaches assume that the source representation can be interpolated or regressed into the target as, given the nature of CNNs, they can approximate highly non-linear functions.

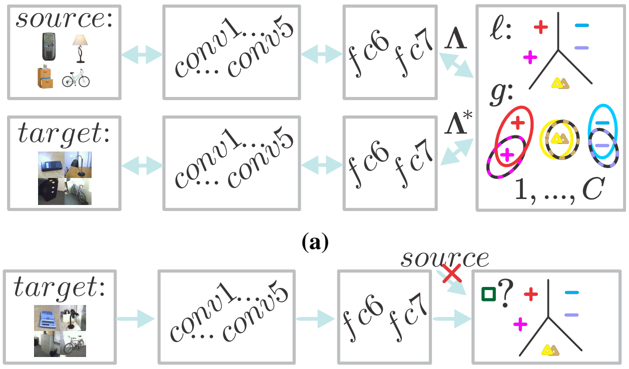
Our model differs in that our source and target network streams co-regularize each other to produce the commonality between the source and target distributions and accommodate the domain-specific parts that should not be aligned.

For visual domains, the commonality can be captured in the spatially-local sense. In [Tommasi et al., 2016], the authors utilize so-called "domainness maps" which capture locally the degree of domain specificity. Similarly, in [Kuzborskij et al., 2016], the authors extract local patches of varying sizes at process each of these patches via CNNs. Our work is orthogonal to these techniques. We represent the commonality globally, however, our ideas could also be applied in a spatially-local setting.

Correlation Methods. Some recent works use correlation between the source and target distributions. Inspired by the Canonical Correlation Analysis (CCA), the authors of [Yeh et al., 2014] utilize a correlation subspace as a joint representation for associating the data across different domains. They also use kernelized CCA. In [Sun et al., 2016], the authors propose an unsupervised domain adaptation by the correlation alignment.

Our work is somewhat related in that we utilize second-order statistics. However, we align partially the class-specific source and target distributions to define the commonality (partial intersection of scatters) in the supervised setting. We also align partially the distribution means while the above unsupervised approaches use zero-centered feature vectors and the full alignment of the generic (c.f. class-specific) source and target distributions. We detail how to learn the degree of alignment in an end-to-end fashion and introduce the kernelized loss between the second- and/or higher-order scatter tensors; all being novel propositions.

Tensor Methods. Correlation approaches outlined above use second-order scatter matrices which are tensors of order $r=2$. In this work, we also investigate the applicability of higher-order scatters $r \geq 3$ for alignment. Third-order tensors have been found useful for various



(a)

(b)

Figure 2.2: The pipeline. Figure 2.2a shows the source and target network streams which merge at the classifier level. The classification and alignment loss ℓ and g take the data $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^*$ from both streams and participate in end-to-end learning. At the test time, we use the target stream and the trained classifier as in Figure 2.2b.

vision tasks. For example, spatio-temporal third-order tensor on video data is proposed for action analysis in [Kim et al., 2007], non-negative tensor factorization is used for image denoising in [Shashua and Hazan, 2005], tensor textures are proposed for texture rendering in [Vasilescu and Terzopoulos, 2004], and higher order tensors are used for face recognition in [Vasilescu and Terzopoulos, 2002]. A survey of multi-linear algebraic methods for tensor subspace learning is available in [Lu et al., 2011]. The above applications use a single tensor, while our goal is to use tensors as the domain- and class-specific representations, similar to the sum-kernel approaches [Koniusz et al., 2016b; Koniusz and Cherian, 2016; Koniusz et al., 2016a], and apply them to alignment tasks.

2.4 Background

In this section, we review notations and the necessary background on scatter tensors, polynomial kernels and their linearizations, which are useful in deriving our mixture of alignments of second- and/or higher-order scatter tensors.

2.4.1 Notations

Let $\mathbf{x} \in \mathbb{R}^d$ be a d-dimensional feature vector. Then, we use $\boldsymbol{\mathcal{X}} = \uparrow\otimes_r \mathbf{x}$ to denote the r-mode super-symmetric rank-one tensor $\boldsymbol{\mathcal{X}}$ generated by the r-th order outer-product of $\mathbf{x}$, where the element of $\boldsymbol{\mathcal{X}} \in \mathfrak{S}^d_{\times^r}$ at the $(i_1, i_2, ..., i_r)$-th index is given by $\mathit{\Pi}_{j=1}^r x_{i_j}$. $\mathcal{I}_N$ stands for the index set $\{1, 2, ..., N\}$. We denote the space of super-symmetric tensors of dimension $d\times...\times d$ with r modes as $\mathfrak{S}^d_{\times^r} \subset \mathbb{R}^{\times_r d}$, where $\mathbb{R}^{\times_r d}$ is the space of tensors $\mathbb{R}^{d\times...\times d}$ with r modes. The Frobenius norm of tensor is given by $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\sum\limits_{i_1,i_2,...,i_r} \mathcal{X}^2_{i_1,i_2,...,i_r}}$, where $\mathcal{X}_{i_1,i_2,...,i_r}$ represents the

$(i_1, i_2, ..., i_r)$-th element of $\boldsymbol{\mathcal{X}}$. Similarly, the inner-product between two tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ is given by $\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{Y}}\rangle=\sum\limits_{i_1,i_2,...,i_r}\mathcal{X}_{i_1,i_2,...,i_r}\cdot\mathcal{Y}_{i_1,i_2,...,i_r}$. Using Matlab style notation, the $(i_3, ..., i_r)$-th slice of $\boldsymbol{\mathcal{X}}$ is given by $\boldsymbol{\mathcal{X}}_{:,:,i_3,...,i_r}$. The space of positive semi-definite matrices is $\mathcal{S}_+^d$. Lastly, $\mathbb{1}$ denotes a vector with all coefficients equal one.

2.4.2 Second- or Higher-order Scatter Tensors

We define a scatter tensor of order r as a mean-centered TOSST representation [Koniusz and Cherian, 2016]:

Definition 1. *Suppose $\boldsymbol{\phi}_n \in \mathbb{R}^d, \forall n \in \mathcal{I}_N$ represent some data vectors, then a scatter tensor $\boldsymbol{\mathcal{X}} \in \mathfrak{S}^d_{\times^r}$ of order r on these data vectors is given by:*

$$\boldsymbol{\mathcal{X}} = \frac{1}{N}\sum_{n=1}^{N} \uparrow\otimes_r (\boldsymbol{\phi}_n - \boldsymbol{\mu}) \quad \textit{and} \quad \boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}_n. \tag{2.1}$$

In our supervised domain adaptation setting, the scatter tensors are obtained via applying (2.1) on the class-specific data vectors such as outputs of the *fc7* layer of AlexNet. When we need to highlight order r of $\boldsymbol{\mathcal{X}}$, we write $\boldsymbol{\mathcal{X}}^{(r)}$.

The following properties of the scatter tensors are worth noting (see [Koniusz and Cherian, 2016] for proofs):

Proposition 1. *For a scatter tensor $\boldsymbol{\mathcal{X}} \in \mathfrak{S}^d_{\times^r}$, we have:*

1. *Super-Symmetry: $\boldsymbol{\mathcal{X}}_{i_1,i_2,...,i_r} = \boldsymbol{\mathcal{X}}_{\Pi(i_1,i_2,...,i_r)}$ for indexes $(i_1, i_2, ..., i_r)$ and their any permutation Π. The number of unique coefficients of $\boldsymbol{\mathcal{X}}$ is $\binom{d+r-1}{r}$.*

2. *Every slice is at least positive semi-definite for any even order $r \geq 2$ and $\boldsymbol{\mathcal{X}}_{:,:,i_3,...,i_r} \in \mathcal{S}_+^d, \forall(i_3, ..., i_r) \in \mathcal{I}_d$.For $r = 2$, tensor $\boldsymbol{\mathcal{X}}$ also is a covariance matrix.*

3. *Indefiniteness for any odd order $r \geq 1$,* i.e., *under a CP decomposition [De Lathauwer et al., 2000], it can have positive, negative, or zero entries in its core-tensor.*

Due to the indefiniteness of tensors of odd orders and potential rank deficiency, we restrict ourselves to work with the Euclidean distance between such scatter representations. Also, as the number of unique coefficients of $\boldsymbol{\mathcal{X}}$ is of order $\sim d^r$, which is prohibitive for $r \geq 3$, we propose a light-weight kernelized variant of the Euclidean distance which avoids explicit use of tensors. The following easily verifiable two results will come handy in the sequel:

Proposition 2. *Suppose we want to evaluate the Frobenius norm between tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}^* \in \mathfrak{S}^d_{\times^r}$, then it holds that:*

$$||\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^*||_F^2 = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle - 2 \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}}^* \rangle + \langle \boldsymbol{\mathcal{X}}^*, \boldsymbol{\mathcal{X}}^* \rangle . \tag{2.2}$$

Proof. $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}^*$ can be vectorized and the Frobenius norm replaced by the ℓ_2-norm for which the above expansion is known to hold. □

Proposition 3. *Suppose* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ *are two arbitrary vectors, then for an ordinal* $r > 0$*, we have:*

$$\langle \mathbf{x}, \mathbf{y} \rangle^r = \langle \uparrow\otimes_r \mathbf{x}, \uparrow\otimes_r \mathbf{y} \rangle . \tag{2.3}$$

Moreover, for sets of vectors $\mathbf{x}_n, \mathbf{y}_{n'} \in \mathbb{R}^d$*, we have:*

$$\sum_n \sum_{n'} \langle \mathbf{x}_n, \mathbf{y}_{n'} \rangle^r = \Big\langle \sum_n \uparrow\otimes_r \mathbf{x}_n, \sum_{n'} \uparrow\otimes_r \mathbf{y}_{n'} \Big\rangle . \tag{2.4}$$

Proof. The expansion in (2.3) is derived in [Koniusz et al., 2016b] while (2.4) can be verified due to bilinear properties of the dot-product. □

2.5 Proposed Approach

In this section, we first formulate the problem of mixture of alignments of second- and/or higher-order scatter tensors, which precedes an exposition to our next two contributions: a weighted mixture of alignments and a kernelized approach which avoids explicit evaluations of scatters.

2.5.1 Problem Formulation

Suppose $\mathcal{I}_N$ and $\mathcal{I}_{N^*}$ are the indexes of N source and N^* target training data points. $\mathcal{I}_{N_c}$ and $\mathcal{I}_{N_c^*}$ are the class-specific indexes for $c \in \mathcal{I}_C$, where C is the number of classes. Suppose we have feature vectors from *fc7* in the source network stream, one per image, and associated with them labels. Such pairs are given by $\boldsymbol{\Lambda} \equiv \{(\boldsymbol{\phi}_n, y_n)\}_{n \in \mathcal{I}_N}$, where $\boldsymbol{\phi}_n \in \mathbb{R}^d$ and $y_n \in \mathcal{I}_C$, $\forall n \in \mathcal{I}_N$, as shown in Figure 2.2a. For the target data, by analogy, we define pairs $\boldsymbol{\Lambda}^* \equiv \{(\boldsymbol{\phi}^*_n, y^*_n)\}_{n \in \mathcal{I}^*_N}$, where $\boldsymbol{\phi}^* \in \mathbb{R}^d$ and $y^*_n \in \mathcal{I}_C$, $\forall n \in \mathcal{I}^*_N$. Class-specific sets of feature vectors are given as $\boldsymbol{\Phi}_c \equiv \{\boldsymbol{\phi}^c_n\}_{n \in \mathcal{I}_{N_c}}$ and $\boldsymbol{\Phi}^*_c \equiv \{\boldsymbol{\phi}^{c*}_n\}_{n \in \mathcal{I}_{N^*_c}}$, $\forall c \in \mathcal{I}_C$. Then, $\boldsymbol{\Phi} \equiv (\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_C)$ and $\boldsymbol{\Phi}^* \equiv (\boldsymbol{\Phi}^*_1, ..., \boldsymbol{\Phi}^*_C)$. Note that we use the asterisk symbol written in superscript (*e.g.* $\boldsymbol{\phi}^*$) to denote variables associated with the target network whilst the source-related and generic variables have no such indicator. Below, we formulate our problem as a trade-off between the classifier loss ℓ and the alignment loss g which acts on the scatter tensors and is related to their means:

$$\begin{aligned} \underset{\substack{\mathbf{W},\mathbf{b},\boldsymbol{\Theta},\boldsymbol{\Theta}^* \\ \text{s. t. } \|\boldsymbol{\phi}_n\|_2^2\leq\tau, \\ \|\boldsymbol{\phi}^*_{n'}\|_2^2\leq\tau, \\ \forall n\in\mathcal{I}_N, n'\in\mathcal{I}_N^*}}{\arg\min} \quad & \ell(\mathbf{W},\mathbf{b},\boldsymbol{\Lambda}\cup\boldsymbol{\Lambda}^*)+\lambda\|\mathbf{W}\|_F^2 \\ & +\underbrace{\frac{\sigma_1}{C}\sum_{c\in\mathcal{I}_C}\|\boldsymbol{\mathcal{X}}_c-\boldsymbol{\mathcal{X}}^*_c\|_F^2+\frac{\sigma_2}{C}\sum_{c\in\mathcal{I}_C}\|\boldsymbol{\mu}_c-\boldsymbol{\mu}^*_c\|_2^2}_{g(\boldsymbol{\Phi},\boldsymbol{\Phi}^*)}. \end{aligned} \tag{2.5}$$

For ℓ, we use a generic loss used by CNNs, say Softmax. The matrix $\mathbf{W}\in\mathbb{R}^{d\times C}$ contains unnormalized probabilities (c.f. hyperplane of SVM), $\mathbf{b}\in\mathbb{R}^C$ is the bias term, and λ is the regularization constant. Moreover, the union $\boldsymbol{\Lambda}\cup\boldsymbol{\Lambda}^*$ of the source and target training data reveals that we train one universal classifier for both domains[1]. In Equation (2.5), separating the class-specific distributions is addressed by ℓ while bringing closer the within-class scatters of both network streams is handled by g (as Figure 2.2 shows). Specifically, our loss g depends on two sets of variables $(\boldsymbol{\mathcal{X}}_1(\boldsymbol{\Phi}_1),...,\boldsymbol{\mathcal{X}}_C(\boldsymbol{\Phi}_c)),(\boldsymbol{\mu}_1(\boldsymbol{\Phi}_1),...,\boldsymbol{\mu}_C(\boldsymbol{\Phi}_C))$ and $(\boldsymbol{\mathcal{X}}^*_1(\boldsymbol{\Phi}^*_1),...,\boldsymbol{\mathcal{X}}^*_C(\boldsymbol{\Phi}^*_C)),(\boldsymbol{\mu}^*_1(\boldsymbol{\Phi}^*_1),...,\boldsymbol{\mu}^*_C(\boldsymbol{\Phi}^*_C))$ – one set per network stream. Feature vectors $\boldsymbol{\Phi}(\boldsymbol{\Theta})$ and $\boldsymbol{\Phi}^*(\boldsymbol{\Theta}^*)$ depend on the parameters of the source and target network streams $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}^*$ that we optimize over *e.g.*, they represent coefficients of convolutional filters and weights of *fc* layers. $\boldsymbol{\mathcal{X}}_c$, $\boldsymbol{\mathcal{X}}^*_c$, $\boldsymbol{\mu}_c$ and $\boldsymbol{\mu}^*_c$ denote the scatter tensors and means, respectively, one tensor/mean pair per network stream per class, evaluated as in (2.1). Lastly, σ_1 and σ_2 control the overall degree of the scatter and mean alignment, τ constraints the ℓ_2-norm of feature vectors (needed if λ is low). Derivatives of loss g are given in Appendix A.

In this work, we assume that highly non-linear CNN streams are able to rotate the within-class scatters sufficiently as dictated by our loss to yield a desired overlap of two scatters. Such an assumption is common in *i.e.* [Chopra et al., 2013; Wang and Hebert, 2016].

2.5.2 Weighted Alignment Loss

Below we propose a weighted variant of alignment loss g that incorporates class-specific weights $\boldsymbol{\zeta},\bar{\boldsymbol{\zeta}}\in\mathbb{R}^C$ that adjust the degree of alignment per class between the within-class scatters as well as related to them means. As the statistical literature states that combination of moments $m=1,...,\infty$ can capture any distribution, we combine $r'=2,...,r$ orders

[1] For VGG streams, we use a couple of domain-specific classifiers *e.g.*,$\ell(\mathbf{W},\mathbf{b},\boldsymbol{\Lambda})+\ell(\mathbf{W}^*,\mathbf{b}^*,\boldsymbol{\Lambda}^*)+\lambda\|\mathbf{W}\|_F^2+\lambda^*\|\mathbf{W}^*\|_F^2+\beta'\|\mathbf{W}-\mathbf{W}^*\|_F^2$.

($\boldsymbol{\mathcal{X}}_c^{(1)}\!=\!\boldsymbol{\mathcal{X}}_c^{*(1)}\!=\!0$ due to data centering):

$$
\begin{aligned}
g^{(r)}(\boldsymbol{\Phi},\boldsymbol{\Phi}^*_,\{\zeta_{r'}\}_{r'\in\mathcal{I}_r},\bar{\zeta}) =& \frac{\sigma_1}{rC}\sum_{r'\in\mathcal{I}_r\setminus\{1\}}\sum_{c\in\mathcal{I}_C}\zeta_{cr'}||\boldsymbol{\mathcal{X}}_c^{(r')}-\boldsymbol{\mathcal{X}}_c^{*(r')}||_F^2\\
&+\frac{\sigma_2}{C}\sum_{c\in\mathcal{I}_C}\bar{\zeta}_c||\boldsymbol{\mu}_c-\boldsymbol{\mu}_c^*||_2^2+\frac{\alpha_1}{r}\sum_{r'\in\mathcal{I}_r\setminus\{1\}}||\boldsymbol{\zeta}_{r'}-\mathbb{1}||_2^2+\alpha_2||\bar{\boldsymbol{\zeta}}-\mathbb{1}||_2^2,
\end{aligned}
\tag{2.6}
$$

where α_1 and α_2 control the degree of weight deviation. To use the weighted alignment, we replace the corresponding loss in Eq. (2.5) by the alignment loss g defined in (2.6). Then, we additionally minimize (2.5) over $\bar{\boldsymbol{\zeta}}$ and a set $\{\boldsymbol{\zeta}_{r'}\}_{r'\in\mathcal{I}_r\setminus\{1\}}$ that determines contributions of tensors of order $r'\!=\!2,...,r$.

2.5.3 Kernelized Alignment Loss

Evaluating scatter tensors during the gradient descent is costly, even if using covariances ($r=2$), as the typical size feature vectors from *fc7* is $d = 4096$. Below we propose an efficient kernelization of the Frobenius norm on tensors of arbitrary order r.

Proposition 4. *The inner-product of scatter tensors* $\boldsymbol{\mathcal{X}}^{(r)}, \boldsymbol{\mathcal{Y}}^{(r)} \in \mathfrak{S}^d_{\times^r}$ *of order r from Eq.* (2.1)*, can be written implicitly as a sum of entries of a polynomial kernel* $\bar{\bar{\mathbf{K}}}^r \in \mathbb{R}^{N\times N^*}$*, where* $\bar{\bar{K}}^r_{nn'} = \langle \mathbf{x}_n-\boldsymbol{\mu}, \mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r$*, and* $\mathbf{x}_n \in \mathbb{R}^d, \forall n \in \mathcal{I}_N$ *and* $\mathbf{y}_{n'} \in \mathbb{R}^d, \forall n' \in \mathcal{I}_{N^*}$ *are some N and N* feature vectors (that form* $\boldsymbol{\mathcal{X}}^{(r)}$ *and* $\boldsymbol{\mathcal{Y}}^{(r)}$*),* $\boldsymbol{\mu}$ *and* $\boldsymbol{\mu}^*$ *are their means. Then:*

$$
\langle\boldsymbol{\mathcal{X}}^{(r)}, \boldsymbol{\mathcal{Y}}^{(r)}\rangle = \frac{1}{NN^*}\sum_{n}\sum_{n'}\langle\mathbf{x}_n-\boldsymbol{\mu}, \mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r = \frac{1}{NN^*}\mathbb{1}^T\bar{\bar{\mathbf{K}}}^r\mathbb{1}.
\tag{2.7}
$$

Proof. Substituting $\mathbf{x}_n-\boldsymbol{\mu}$ and $\mathbf{y}_{n'}-\boldsymbol{\mu}^*$ into Proposition 3, the proof follows. □

Proposition 5. *Suppose we have polynomial kernels* $\mathbf{K}^r \in \mathbb{R}^{N\times N}$*,* $\bar{\mathbf{K}}^r \in \mathbb{R}^{N^*\times N^*}$ *and* $\bar{\bar{\mathbf{K}}}^r \in \mathbb{R}^{N\times N^*}$ *defined as* $K^r_{nn'} = \langle\mathbf{x}_n-\boldsymbol{\mu}, \mathbf{x}_{n'}-\boldsymbol{\mu}\rangle^r$*,* $\bar{K}^r_{nn'} = \langle\mathbf{y}_n-\boldsymbol{\mu}^*, \mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r$ *and* $\bar{\bar{K}}^r_{nn'} = \langle\mathbf{x}_n-\boldsymbol{\mu}, \mathbf{y}_{n'}-\boldsymbol{\mu}^*\rangle^r$*, where* $\mathbf{x}_n, \mathbf{y}_{n'}, \boldsymbol{\mu}, \boldsymbol{\mu}^*, N, N^*$ *are defined as in Proposition 4. The Frobenius norm between two scatter tensors* $\boldsymbol{\mathcal{X}}^{(r)}, \boldsymbol{\mathcal{Y}}^{(r)} \in \mathfrak{S}^d_{\times^r}$ *of order r, which are defined in Eq.* (2.1)*, can be expressed implicitly as:*

$$
||\boldsymbol{\mathcal{X}}^{(r)}-\boldsymbol{\mathcal{X}}^{*(r)}||_F^2 = \frac{1}{N^2}\mathbb{1}^T\mathbf{K}^r\mathbb{1}+\frac{1}{N^{*2}}\mathbb{1}^T\bar{\mathbf{K}}^r\mathbb{1}-\frac{2}{NN^*}\mathbb{1}^T\bar{\bar{\mathbf{K}}}^r\mathbb{1}.
\tag{2.8}
$$

Proof. Combining Proposition 2 with 4, the proof follows. □

Derivatives of (2.8) are in Appendix B. Equation (2.8) can be evaluated on class-specific feature vectors and substituted directly into the loss functions in (2.5) and (2.6). This way,

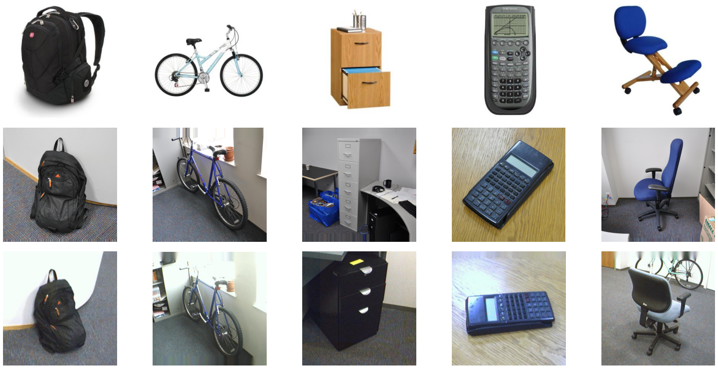


Figure 2.3: The Office dataset. Top, middle and bottom rows show examples from the Amazon, DSLR, and Webcam domains.

we obtain two different regimes for evaluating the Frobenius norm on the scatter tensors: one explicit and one kernelized; both exhibiting different strengths as detailed below.

Complexity. The Frobenius norm on the scatter tensors has complexity $\mathcal{O}((N+N^*+1)D)$, where $D = \binom{d+r-1}{r}$ as detailed in Proposition 1. The kernelized variant proposed above has complexity $\mathcal{O}((N^2 + NN^* + N^*N^*)(d+\rho)$, where $\rho \leq \log r$ estimates the complexity of "rising x to the power of r". As $\rho \ll d$, its cost is negligible and can be safely left out from the above analysis.

It is easy to verify that, for the standard domain adaptation problems with $N = 20$ source and $N^* = 3$ target training points per class, $d = 4096$ and $r = 2$, explicit evaluations of the Frobenius norm are $\sim$52× slower than the proposed by us kernelized substitute. For the same scenario but with the scatter tensor of order $r = 3$, explicit evaluations of the Frobenius norm are not tractable, as they take $\sim$ 143000× more time than the kernelized substitute, which demonstrates the clear benefit of our approach. The kernelization makes Eq. (2.6) tractable for $r > 2$.

2.6 Experiments

In this section, we present experiments demonstrating the usefulness of our framework. We start by describing datasets we use in evaluations.

2.6.1 Datasets

Office dataset. A popular dataset for evaluating algorithms against the effect of domain shift is the Office dataset [Saenko et al., 2010] which contains 31 object categories in three domains: Amazon, DSLR and Webcam. The 31 categories in the dataset consist of objects commonly encountered in office settings, such as keyboards, file cabinets, and laptops. The Amazon domain contains on average 90 images per class and 2817 images in total. As these images were captured from a website of online merchants, they are captured against clean background and at a unified scale. The DSLR domain contains 498 low-noise high resolution images (4288 × 2848). There are 5 objects per category. Each object was captured from different viewpoints on average 3 times. For Webcam, the 795 images of low resolution (640 × 480) exhibit significant noise and color as well as white balance artifacts. Otherwise, 5 objects per category were also used in the capturing process. Figure 2.3 illustrates the three domains. We distinguish the following six domain shifts: Amazon-Webcam ($\mathcal{A}\rightarrow\mathcal{W}$), Amazon-DSLR ($\mathcal{A}\rightarrow\mathcal{D}$), Webcam-Amazon ($\mathcal{W}\rightarrow\mathcal{A}$), Webcam-DSLR ($\mathcal{W}\rightarrow\mathcal{D}$), DSLR-Amazon ($\mathcal{D}\rightarrow\mathcal{A}$) and DSLR-Webcam ($\mathcal{D}\rightarrow\mathcal{W}$).

We evaluate across 10 randomly chosen data splits per domain shift. We follow the standard protocol for this dataset and, for each training source split, we sample 20 images per category for the Amazon domain and 8 examples per category for the DSLR and Webcam domains. From the training target splits, we sample 3 images per class per split per domain. We present results for the supervised setting and report accuracies on the remaining target images, as the the standard protocol for this dataset suggests.

RGB-D-Caltech256 dataset. The RGB-D [Lai et al., 2011] and Caltech256 [Griffin et al., 2007] datasets have been used as the source and target for evaluations of unsupervised domain adaptation problems [Chen et al., 2014; Motiian and Doretto, 2016]. We use the 10 classes that are common between the two datasets *e.g.*, calculator, cereal box, coffee mug, ball, tomato. We use 50 and 5/10 images per class in the source and target domains for the supervised setting. We test on the remaining target samples. We report the mean average accuracy over 5 data splits, that is, we select randomly the source and target data samples for each split.

Pascal VOC2007-TU Berlin dataset. Transfer from Pascal VOC2007 [Everingham et al.,

	$\mathcal{A}\rightarrow\mathcal{W}$	$\mathcal{A}\rightarrow\mathcal{D}$	$\mathcal{W}\rightarrow\mathcal{A}$	$\mathcal{W}\rightarrow\mathcal{D}$	$\mathcal{D}\rightarrow\mathcal{A}$	$\mathcal{D}\rightarrow\mathcal{W}$	acc.
DLID	51.9	-	-	89.9	-	78.2	73.33
DeCAF$_6$ S+T	80.7±2.3	-	-	-	-	94.8±1.2	87.75
DaNN	53.6±0.2	-	-	83.5±0.0	-	71.2±0.0	69.43
Source CNN	56.5±0.3	64.6±0.4	42.7±0.1	93.6±0.2	47.6±0.1	92.4±0.3	66.23
Target CNN	80.5±0.5	81.8±1.0	59.9±0.3	81.8±1.0	59.9±0.3	80.5±0.5	74.06
Source+Target CNN	82.5±0.9	85.2±1.1	65.2±0.7	96.3±0.5	65.8±0.5	93.9±0.5	81.48
Dom. Conf.+Soft Labs.	82.7±0.8	86.1±1.2	65.0±0.5	**97.6±0.2**	66.2±0.3	**95.7±0.5**	82.22
Source+Target CNN	82.4±2.0	85.5±0.9	65.1±1.4	95.8±0.8	66.0±1.2	94.3±0.6	81.53
Second-order (*So*)	**84.5±1.7**	**86.3±0.8**	**65.7±1.7**	97.5±0.7	**66.5±1.0**	95.5±0.6	**82.68**

Table 2.1: Comparison of our second-order alignment loss (*So*) to the state of the art on the Office dataset. Results of DLID [Chopra et al., 2013], DeCAF$_6$ S+T [Donahue et al., 2014], DaNN [Ghifary et al., 2014], Source CNN [Tzeng et al., 2015], Target CNN [Tzeng et al., 2015], Source+Target CNN [Tzeng et al., 2015], Dom. Conf.+Soft Labs.[Tzeng et al., 2015] on each split is shown.

		sp1	sp2	sp3	sp4	sp5	sp6	sp7	sp8	sp9	sp10	aver. acc.
$\mathcal{A}\to\mathcal{W}$	*S*+*T*	91.5	87.9	91.5	89.6	89.0	87.2	86.2	87.2	91.2	86.5	88.76±1.9
	So	91.9	89.3	91.7	90.6	89.0	88.2	87.6	87.6	91.9	87.2	89.50±1.8
	So+ζ	92.4	89.9	90.5	92.2	88.9	88.2	90.0	89.5	91.3	89.6	**90.24±1.3**
	To+ζ	92.5	90.3	91.8	91.9	89.0	89.9	89.3	89.6	91.6	89.6	**90.55±1.1**
	So+*To*+ζ	92.6	90.5	92.0	92.0	89.2	90.0	89.6	89.9	91.8	89.9	**90.75±1.2**
	So+*To*+*Fo*+ζ	93.0	90.7	92.1	92.9	89.2	89.9	89.6	89.9	92.0	89.7	**90.92±1.3**
$\mathcal{A}\to\mathcal{D}$	*S*+*T*	90.6	88.9	89.4	92.4	90.1	87.2	91.1	88.2	90.9	89.4	89.83±1.4
	So	92.4	92.4	91.1	92.4	92.9	89.6	93.4	91.9	94.1	92.6	**92.26±1.1**
	So+*To*+ζ	92.7	92.9	91.6	92.5	93.3	89.7	93.7	91.9	94.0	93.0	**92.52±1.2**
	So+*To*+*Fo*+ζ	93.1	93.1	92.0	92.7	93.3	89.9	94.1	91.9	94.0	93.4	**92.73±1.1**

Table 2.2: The Office dataset on VGG streams. (Top) $\mathcal{A}\to\mathcal{W}$ and (Bottom) $\mathcal{A}\to\mathcal{D}$ domain shifts are evaluated on second-order (*So*), second- (*So*+ζ) and third-order+weights (*To*+ζ), second- and third- (*So*+*To*+ζ) and fourth-order (*So*+*To*+*Fo*+ζ) alignment with weight learning. Our baseline fine-tuning on the combined source and target domains (*S*+*T*) is also evaluated for comparison.

2007] to TU Berlin [Eitz et al., 2012] (images-to-sketches transfer) has never been attempted yet in domain adaptation to our best knowledge. We utilize 50 and 3 source and target training samples per class, respectively, and the 14 classes that are common between the source and target datasets. We perform testing on the remaining target data. We report the mean average accuracy over 5 data splits.

2.6.2 Experimental Setup

In each stream, we employ the AlexNet architecture [Krizhevsky et al., 2012] which was pre-trained on the ImageNet dataset [Russakovsky et al., 2015] for the best results. At the training and testing time, we use the pipelines shown in Figures 2.2a and 2.2b, respectively. Where stated, we use the 16-layer VGG model [Simonyan and Zisserman, 2014b] per stream to quantify the impact of different CNN models on our algorithm. We set non-zero learning rates on the fully-connected and the last two convolutional layers of the two streams.

On the RGB-D-Caltech256 dataset, we use the RGB images from Caltech256 as the target domain. In contrast to [Chen et al., 2014; Motiian and Doretto, 2016] which use both the RGB data and depth maps as a source, we adapt our source stream based on AlexNet to use only the depth data from the RGB-D dataset – this helps us isolate performance of our algorithm in case of distinct heterogeneous domains. As these both domains are very different from each other, we apply two classifiers – one per network stream (see the footnote[1] in Section 2.5).

We evaluate Second- and/or Higher-order Transfer of Knowledge (*So*-*HoT*) approaches such as: unweighted and weighted second-order alignment losses (*So*) and (*So*+ζ), the third-order loss (*To*) and its weighted variant (*To*+ζ), and combined second- and third- (*So*+*To*+ζ) as well as fourth-order (*So*+*To*+*Fo*+ζ) weighted alignment losses. The model parameters were selected by cross-validation.

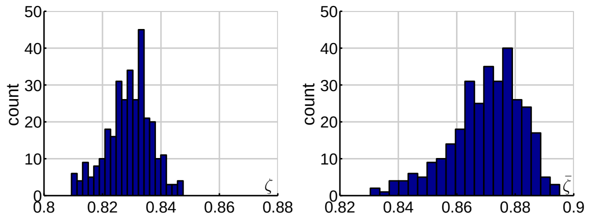
2.6.3 Comparison to the State of the Art

We apply our algorithm on the Office dataset. Table 2.1 presents results for the six domain shifts. Our second-order alignment loss (*So*) is compared against the baseline (*S*+*T*) for which the source and target training samples were used together to fine-tune a standard CNN network. As can be seen, our method outperforms such a baseline as well as recent approaches such as *Domain Confusion with Soft Labels* and fine-tuning on the source or target data, respectively.

Performance on the VGG architecture. To evaluate effectiveness of our algorithm on other powerful networks, we follow the same pipeline as in Figure 2.2, except that we employ the pre-trained VGG [Simonyan and Zisserman, 2014b] in place of AlexNet [Krizhevsky et al., 2012]. As VGG utilizes more parameters than AlexNet, we demonstrate in Table 2.2 that applying our second-order alignment loss (*So*) on $\mathcal{A}\rightarrow\mathcal{W}$ and $\mathcal{A}\rightarrow\mathcal{D}$ improves performance compared to the baselines (*S*+*T*) by 0.74% and 2.43%. Without resorting to data augmentations, we outperform *e.g.* a multi-scale multi-patch CNN approach [Kuzborskij et al., 2016] by 0.6% on $\mathcal{A}\rightarrow\mathcal{W}$.

Weighted vs. Unweighted Alignment. In this experiment, we demonstrate the benefit of using the weighted alignment of the scatter matrices and their means on the $\mathcal{A}\rightarrow\mathcal{W}$ and $\mathcal{A}\rightarrow\mathcal{D}$ domain shits. Table 2.2 shows that our weighted second- (*So*+ζ) and third-order (*To*+ζ) alignment losses, introduced in Eq. (2.6), improve over our unweighted second-order alignment loss (*So*) from Eq. (2.5) by 0.74% and 1.05% on $\mathcal{A}\rightarrow\mathcal{W}$, respectively. Learning ζ and $\bar{\zeta}$ can be implemented at no visible increase in computations.

In Figure 2.4, we show histograms of the ζ and $\bar{\zeta}$ weights from (*So*+ζ) over the 31 classes and the 10 splits. The histograms reveal that the levels of alignment of the scatter matrices and their means vary according to the Beta distributions. The means of these distributions are slightly below the desired mean value of one which indicates that, in this experiment, σ_1 and σ_2 from Eq. (2.6) were initialized with values larger than needed. Also, their optimal values



(a) **(b)**

Figure 2.4: Histograms of the ζ and $\bar{\zeta}$ weights in plots 2.4a and 2.4b, learned on $\mathcal{A}\rightarrow\mathcal{W}$, show the level of alignment of the scatter matrices and their means according to the loss function in (2.6).

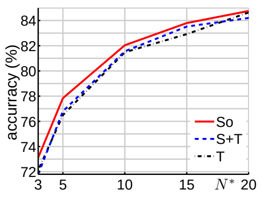


Figure 2.5: Our second-order algorithm (*So*) vs. the baseline fine-tuning on: i) the combined source and target domains (*S*+*T*) and ii) the target domain only (*T*). We use RGB-D-Caltech256 (heterogeneous setup). N^* is the number of target tr. samples per class.

might vary over time – learning weights compensates for this.

Figure 2.6 shows that, as $\sigma_1 \to 0$ and $\sigma_2 \to 0$, our algorithm converges to the baseline fine-tuning on the combined source and target domains (S+T) which yielded 85.9% accuracy for split *sp1*. Moreover, the overall performance is stable *i.e.*, within ±0.2% accuracy, for a large range of values *e.g.*, $5e{-}9 \leq \sigma_1 \leq 5e{-}8$ and $1e{-}6 \leq \sigma_2 \leq 5e{-}5$.

Alignment of combined Second-, Third- and Fourth-order Scatter Tensors. Our kernelized loss in Eq. (2.8) admits alignment between second- and/or higher-order scatter tensors which, beyond the scale/shear and orientation, capture higher-order statistical moments. In Table 2.2, we evaluate third-order weighted alignment loss (*To*+ζ), as well as combined second-, third- (*So*+*To*+ζ) and fourth-order (*So*+*To*+*Fo*+ζ) weighted alignments. As the order increases, the performance improves. For (*So*+*To*+*Fo*+ζ), we outperform (*So*) by 1.42% and 2.9% on $\mathcal{A}\!\to\!\mathcal{W}$ and $\mathcal{A}\!\to\!\mathcal{D}$.

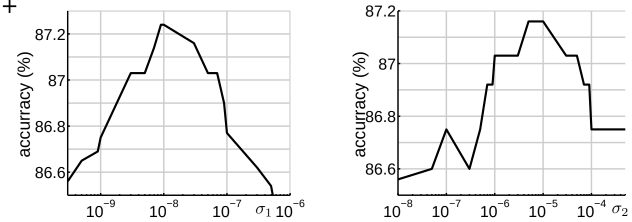
Heterogeneous setting on RGB-D-Caltech256. In this experiment, we verify the behavior of our second-order alignment loss (*So*) w.r.t. the varying number of target training samples N^*. Figure 2.5 shows that the largest improvement of 1.24% and 1.04% over the baseline (S+T) is obtained for a small number $N^* = 3$ and $N^* = 5$, respectively. As N^* increases, the improvement over baselines becomes smaller. Such a trend is consistent with other works on domain adaptation [Tommasi et al., 2010]. In some cases, the baseline (S+T) performs worse than the fine-tuning on target only (T) which is known as so-called *negative transfer* [Tommasi et al., 2010]. For all $3 \leq N^* \leq 20$, our (*So*) outperforms baselines (S+T) and (T) which demonstrates robustness of our approach.

Heterogeneous setting on Pascal VOC2007-TU Berlin. Table 2.3 shows results on transfer from Pascal VOC2007 [Everingham et al., 2007] to TU Berlin [Eitz et al., 2012] (images-to-sketches transfer). These dataset have never been used together in domain adaptation. We utilize 50 and 3 source and target training samples per class, respectively, and the 14 classes that are common between the source and target datasets. We use AlexNet streams in this experiment. As demonstrated in the table, our second-order approach (*So*) and the baselines (S+T) and (T) yield **63.4**, 62.66 and 62.46%, respectively.

Comparisons to CORAL on the Office dataset. To compare (*So*) to CORAL [Sun et al., 2016], we modified our code to align second-order marginal statistics (M) in the supervised setting. For AlexNet and $\mathcal{A}\!\to\!\mathcal{W}$, (M) scores 82.6% vs. baseline (S+T) of 82.4% but is below 84.5% from our (*So*). On $\mathcal{D}\!\to\!\mathcal{W}$, (M) gave 94.6% vs. 95.5% from our (*So*). On $\mathcal{W}\!\to\!\mathcal{D}$, (M) gave 95.9% vs. 97.5% from our (*So*).

	sp1	sp2	sp3	sp4	sp5	average acc.
S+T	58.86	63.43	63.14	59.14	68.71	62.66
T	59.86	63.43	64.14	57.86	67.0	62.46
So	60.57	63.28	64.28	59.14	69.71	**63.40**

Table 2.3: Pascal VOC2007-TU Berlin dataset. We use 5 splits and report accuracies on our method (*So*) vs. baselines (*S*+



(a) **(b)**

Figure 2.6: Performance of our second-order alignment loss (*So*) w.r.t. parameters σ_1 (2.6a) and σ_2 (2.6b) on the $\mathcal{A} \rightarrow \mathcal{W}$ domain shift (split *sp1* is used). Note the logarithmic scale.

2.7 Conclusions

We have presented an approach to domain adaptation by partial alignment of the within-class scatters to discover the commonality. The state-of-the-art results we obtain suggest that our simple strategy is effective despite challenges of domain adaptation. Moreover, the presented weighted approach and kernelized alignment loss improve the results and computational efficiency. Our method can be easily extended to multiple domains and other network architectures. In the next chapter, we will address the issues we have faced in the Office dataset and propose a new more challenging dataset for domain adaptation. Moreover, we will update our alignment loss to work with non-Euclidean metrics.

Chapter 3

Museum Exhibit Identification Challenge for Domain Adaptation and Beyond

3.1 Summary

This chapter approaches an open problem of artwork identification and proposes a new dataset dubbed Open Museum Identification Challenge (Open MIC). It contains photos of exhibits captured in 10 distinct exhibition spaces of several museums, which showcase paintings, timepieces, sculptures, glassware, relics, science exhibits, natural history pieces, ceramics, pottery, tools, and indigenous crafts. The goal of Open MIC is to stimulate research in domain adaptation, egocentric recognition, and few-shot learning by providing a testbed complementary to the famous Office dataset, which reaches $\sim$90% accuracy [Koniusz et al., 2017]. To form our dataset, we captured several images per art piece with a mobile phone and wearable cameras to create the source and target data splits, respectively. To achieve robust baselines, we build on a recent approach that aligns per-class scatter matrices of the source and target CNN streams [Koniusz et al., 2017]. Moreover, we exploit the positive definite nature of such representations by using end-to-end Bregman divergences and the Riemannian metric. We present baselines such as training/evaluation per exhibition and training/evaluation on the combined set covering 866 exhibit identities. Each exhibition poses distinct challenges e.g., quality of lighting, motion blur, occlusions, clutter, viewpoint and scale variations, rotations, glares, transparency, non-planarity, clipping, we break down results w.r.t. these factors.

As stated in the previous chapter, we will address the issues we have faced in the Office dataset [Saenko et al., 2010]. Office dataset is commonly used in supervised domain adaptation literature. But domain shift in the dataset is rather trivial, which we can see in the results reaching 90+% accuracy. Also, the total number of images is lower than datasets that are used in deep models. We create the large and more challenging Open Museum Identification Challenge dataset to address these problems, which provides annotated source and target samples with more complex domain shifts in between them. We further improve our alignment loss to make it tractable to train deep models with non-Euclidean metrics. This chapter has been published as a conference paper: "Piotr Koniusz*, Yusuf Tas*, Hongguang Zhang, Mehrtash Harandi, Fatih Porikli, and Rui Zhang. Museum exhibit identification challenge for the supervised domain adaptation and beyond. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 788-804. 2018" (Oral presentation, $\sim$2% acceptance rate). * indicates shared credit, equal contributions.

3.2 Introduction

Domain adaptation and transfer learning are the problems widely studied in computer vision and machine learning communities [Baxter et al., 1995; Li et al., 2016]. They are inspired by the human cognitive capacity to learn new concepts from very few data samples (cf. training classifier on millions of labeled images from the ImageNet dataset [Russakovsky et al., 2015]). Generally, given a new (target) task to learn, the arising question is how to identify the so-called *commonality* [Tommasi et al., 2010; Koniusz et al., 2017] between this task and previous (source) tasks, and transfer knowledge from the source tasks to the target one. Therefore, one has to address three questions: what to transfer, how, and when [Tommasi et al., 2010].

Domain adaptation and transfer learning utilize annotated and/or unlabeled data and perform tasks-in-hand on the target data *e.g.*, learning new categories from few annotated samples (supervised domain adaptation [Chopra et al., 2013; Tzeng et al., 2015]), utilizing available unlabeled data (unsupervised [Sun et al., 2016; Ganin et al., 2016] or semi-supervised domain adaptation [Daumé III et al., 2010; Tzeng et al., 2015]), recognizing new categories in embedded spaces (*e.g.*attribute-based) without any training samples (zero-shot learning [Fei-Fei et al., 2006]). Problems such as one- and few-shoot learning attempt to train robust class predictors from at most few data points [Fei-Fei et al., 2006].

Recently, algorithms for supervised domain adaptation such as *Simultaneous Deep Transfer Across Domains and Tasks* [Tzeng et al., 2015] and *Second- or Higher-order Transfer (So-HoT)* of knowledge [Koniusz et al., 2017] combined with Convolutional Neural Networks (CNN) [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014b] in end-to-end fashion have reached state-of-the-art results $\sim$90% accuracy on classic benchmarks such as the Office dataset [Saenko et al., 2010]. By and large, such an increase in performance is due to fine-tuning of CNNs on the large-scale datasets such as ImageNet [Russakovsky et al., 2015] and Places Database [Zhou et al., 2014]. Indeed, fine-tuning of CNN is a powerful domain adaptation and transfer learning tool by itself [Girshick et al., 2014; Sermanet et al., 2013]. Furthermore, recent semi-supervised and unsupervised approach to *Learning an Invariant Hilbert*

Space [Herath et al., 2017c] has also reached $\sim$90% accuracy by using generic CNN descriptors vs. $\sim$56% for SURF. The gap between CNN-based and simpler representations is also visible in the *CORAL* method [Sun et al., 2016], for which performance varies between 46% and 70% accuracy. Thereby, these works exhibit saturation for CNN features when evaluated on the Office [Saenko et al., 2010] dataset or its newer Office+Caltech 10 variant [Gong et al., 2012].

Therefore, we propose a new dataset for the task of exhibit identification in museum spaces that challenges domain adaptation and fine-tuning due to its significant domain shifts between the source and target subsets.

For the source domain, we captured the photos in a controlled fashion by Android phones *e.g.*, we ensured that each exhibit is centered and non-occluded in photos. We prevented adverse capturing conditions and did not mix multiple objects per photo unless they were all part of one exhibit. We captured 2–30 photos of each art piece from different viewpoints and distances in their natural settings.

For the target domain, we employed an egocentric setup to ensure *in-the-wild* capturing process. We equipped 2 volunteers per exhibition with cheap wearable cameras and let them stroll and interact with artworks at their discretion. Such a capturing setup is applicable to preference and recommendation systems *e.g.*, a curator takes training photos of exhibits with an Android phone while visitors stroll with wearable cameras to capture data from the egocentric perspective for a system to reason about the most popular exhibits. Open MIC contains 10 distinct source-target subsets of images from 10 different kinds of museum exhibition spaces, each exhibiting various photometric and geometric challenges, as detailed in Section 3.6.

To demonstrate the intrinsic difficulty of Open MIC, we chose useful baselines in supervised domain adaptation detailed in Section 3.6. They include fine-tuning CNNs on the source and/or target data and training a state-of-the-art So-HoT model [Koniusz et al., 2017] which we equip with non-Euclidean distances [Cherian et al., 2012; Pennec et al., 2006] for robust end-to-end learning.

We provide various evaluation protocols which include: (i) training/evaluation per exhibition subset, (ii) training/testing on the combined set that covers all 866 identity labels, (iii) testing w.r.t. various scene factors annotated by us such as quality of lighting, motion blur, occlusions, clutter, viewpoint and scale variations, rotations, glares, transparency, non-planarity, clipping, *etc*.

Moreover, we introduce a new evaluation metric inspired by a saliency problem detailed next. As numerous exhibits can be captured in a target image, we asked our volunteers to enumerate in descending order the labels of most salient/central exhibits they had interest in at a given time followed by less salient/distant exhibits. As we ideally want to understand the volunteers' preferences, the classifier has to decide which detected exhibit is the most salient. We note that the annotation- and classification-related processes are not free of noise. Therefore, we propose to not only look at the top-k accuracy known from ImageNet [Russakovsky et al., 2015] but to also check if any of top-k predictions are contained within the top-n fraction of all ground-truth labels enumerated for a target image. We refer to this as a top-k-n measure.

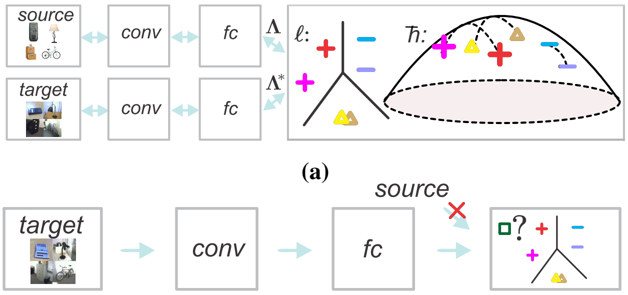
To obtain convincing baselines, we balance the use of an existing approach [Koniusz et al., 2017] with our mathematical contributions and evaluations. The So-HoT model [Koniusz et al., 2017] uses the Frobenius metric for partial alignment of within-class statistics obtained from CNNs. The hypothesis behind such modeling is that the partially aligned statistics capture so-called *commonality* [Tommasi et al., 2010; Koniusz et al., 2017] between the source and target domains; thus facilitating knowledge transfer. For the pipeline in Figure 3.1, we use two CNN streams of the VGG16 network [Simonyan and Zisserman, 2014b] which correspond to the source and target domains. We build scatter matrices, one per stream per class, from feature vectors of the *fc* layers. To exploit benefits of geometry of positive definite matrices, we regularize and align scatters by the Jensen-Bregman LogDet Divergence (*JBLD*) [Cherian et al., 2012] in end-to-end manner and compare to the Affine-Invariant Riemannian Metric (*AIRM*) [Pennec et al., 2006; Bhatia, 2009]. However, evaluations of gradients of non-Euclidean distances are slow for typical 4096×4096 dimensional matrices. We show by the use of Nyström projections that, with typical numbers of data samples per source/target per class being $\sim$50 in domain adaptation, evaluating such distances can be fast and exact.

To summarize, our contributions are as follows: (i) we collect and annotate a new challenging Open MIC dataset with domains consisting of the pictures taken by Android phones and wearable cameras; the latter exhibiting a series of realistic distortions due to the egocentric capturing process, (ii) we compute useful baselines, provide various evaluation protocols, statistics and top-k-n results, as well as include breakdown of results w.r.t. annotated by us scene factors, (iii) we use non-Euclidean JBLD and AIRM distances for end-to-end training of the supervised domain adaptation approach and we exploit the Nyström projections to make this training tractable. To our best knowledge, these distances have not been used before in the supervised domain adaptation due to their high computational complexity.

3.3 Related Work

We start by describing the most popular datasets for the problem at hand and explain how the Open MIC dataset differs from them. Subsequently, we describe various domain adaptation approaches which are related to our work.

Datasets. A popular dataset for evaluating against the effect of domain shift is the Office dataset [Saenko et al., 2010] which contains 31 object categories and three domains: Amazon, DSLR and Webcam. The 31 categories in the dataset consist of objects commonly encountered



(a)

(b)

Figure 3.1: The pipeline. Figure 3.1a shows the source and target network streams which merge at the classifier level. The classification and alignment losses ℓ and $\hbar$ take the data $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^*$ from both streams and participate in end-to-end learning. Loss $\hbar$ aligns covariances on the manifold of $\mathcal{S}_{++}$ matrices. At the test time, we use the target stream and the trained classifier as in Figure 3.1b.

in the office setting, such as keyboards, file cabinets, and laptops. The Amazon domain contains images which were collected from a website of on-line merchants. Its objects appear on clean backgrounds and at a fixed scale. The DSLR domain contains low-noise high resolution images of object captured from different viewpoints while Webcam contains low resolution images. The Office dataset has been used in numerous publications [Sun et al., 2016; Tzeng et al., 2015; Ganin et al., 2016; Chopra et al., 2013; Wang and Hebert, 2016; Kuzborskij et al., 2016; Tommasi et al., 2016; Herath et al., 2017c] that address domain adaptation, to name but a few of approaches. Its recent extension includes a new Caltech 10 domain [Gong et al., 2012].

The Office dataset is primarily used for the transfer of knowledge about object categories between domains. In contrast, our dataset addresses the transfer of instances between domains. Each domain of the Open MIC dataset contains 37–166 specific instances to distinguish from (866 in total) compared to relatively low number of 31 classes in the Office dataset. Moreover, our target subsets are captured in an egocentric manner *e.g.*, we did not align objects to the center of images or control the shutter *etc*.

A recent large collection of datasets for domain adaptation was proposed in technical report [Tommasi and Tuytelaars, 2014] to study cross-dataset domain shifts in object recognition with use of the ImageNet, Caltech-256, SUN, and Bing datasets. Even larger is the latest Visual Domain Decathlon challenge [Rebuffi et al., 2017] which combines datasets such as ImageNet, CIFAR–100, Aircraft, Daimler pedestrian classification, Describable textures, German traffic signs, Omniglot, SVHN, UCF101 Dynamic Images, VGG–Flowers. In contrast, our dataset contains highly varied target appearances which are challenging in few-shot learning scenarios. We target the identity recognition across exhibits captured in egocentric setting which vary from paintings to sculptures to glass to pottery to figurines. Moreover, some artworks in our dataset exhibit fine-grained traits as they are hard to distinguish from without the expert knowledge.

The PIE Multiview dataset [Gross et al., 2010] includes face images of 67 subjects and

Dist./Ref.	$d^2(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}^*)$	Invar.	Tr. Ineq.	Geo.	d if $\mathcal{S}_+$	$\nabla_{\boldsymbol{\Sigma}}$ if $\mathcal{S}_+$	$\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \boldsymbol{\Sigma}}$
Frobenius	$\lVert\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*\rVert_F^2$	rot.	yes	no	fin.	fin.	$2(\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*)$
AIRM	$\lVert\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Sigma}^*\boldsymbol{\Sigma}^{-\frac{1}{2}}\rVert_F^2$	aff./inv.	yes	yes	∞	∞	$-2\boldsymbol{\Sigma}^{-\frac{1}{2}}\log(\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Sigma}^*\boldsymbol{\Sigma}^{-\frac{1}{2}})\boldsymbol{\Sigma}^{-\frac{1}{2}}$
JBLD	$\log\left\lvert\frac{\boldsymbol{\Sigma}+\boldsymbol{\Sigma}^*}{2}\right\rvert-\frac{1}{2}\log\lvert\boldsymbol{\Sigma}\boldsymbol{\Sigma}^*\rvert$	aff./inv.	no	no	∞	∞	$(\boldsymbol{\Sigma}+\boldsymbol{\Sigma}^*)^{-1}-\frac{1}{2}\boldsymbol{\Sigma}^{-1}$

Table 3.1: Frobenius, JBLD [Cherian et al., 2012] and AIRM [Pennec et al., 2006] distances and their properties from the literature. These distances operate between a pair of arbitrary matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^*$ which are points in $\mathcal{S}_{++}$ (and/or $\mathcal{S}_+$ for Frobenius).

exhibits different viewpoints, varies in illumination and expressions. It has been used in the instance-based domain adaptation [Herath et al., 2017c]. Our Open MIC however is not limited to instances of faces or controlled capture setting. Open MIC contains diverse 10 subsets with paintings, timepieces, sculptures, science exhibits, glasswork, relics, ancient animals, plants, figurines, ceramics, native arts *etc*.

Domain adaptation algorithms. Deep learning has been used in the context of domain adaptation in numerous recent works *e.g.*, [Tzeng et al., 2015; Ganin et al., 2016; Chopra et al., 2013; Wang and Hebert, 2016; Kuzborskij et al., 2016; Tommasi et al., 2016; Koniusz et al., 2017]. These works establish the so-called commonality between domains. In [Tzeng et al., 2015], the authors propose to align both domains via the cross entropy which 'maximally confuses' both domains for supervised and semi-supervised settings. In [Chopra et al., 2013], the authors capture the 'interpolating path' between the source and target domains using linear projections into a low-dimensional subspace on the Grassman manifold. In [Wang and Hebert, 2016], the authors propose to learn the transformation between the source and target by the deep regression network. Our model differs in that our source and target network streams co-regularize each other via the JBLD or AIRM distance that respects the non-Euclidean geometry of the source and target matrices. We perform an alignment of scatter matrices advocated in [Koniusz et al., 2017].

For visual domains, the domain adaptation can be applied in the spatially-local sense to target so-called *roots* of domain shift. In [Tommasi et al., 2016], the authors utilize so-called 'domainness maps' which capture locally the degree of domain specificity. Our work is orthogonal to this method. We perform domain adaptation globally in the spatial sense, however, our ideas can be extended to a spatially-local setting.

Some recent works enforce correlation between the source and target distributions *e.g.*, the authors of [Yeh et al., 2014] utilize a correlation subspace as a joint representation for associating the data across different domains. They also use kernelized CCA. In [Sun et al., 2016], the authors propose an unsupervised domain adaptation by the correlation alignment. In [Koniusz et al., 2017], the authors perform class-specific alignment of source and target distributions with use of tensors and the Frobenius norm. Our work is similar in spirit as it utilizes a similar general setup. However, we first project class-specific vector representations from the *fc* layers of the source and target CNN streams to the common space via Nyström projections for tractability and then we combine them with the JBLD or AIRM distance to exploit the (semi)definite positive nature of scatter matrices. We perform end-to-end learning which requires non-trivial derivatives of JBLD/AIRM distance and Nyström projections for computational efficiency.

3.4 Background

In this section, we review our notations and the necessary background on scatter matrices, Nyström projections, the Jensen-Bregman LogDet (*JBLD*) divergence [Cherian et al., 2012] and the Affine-Invariant Riemannian Metric (*AIRM*) [Pennec et al., 2006; Bhatia, 2009].

3.4.1 Notations

Let $\mathbf{x} \in \mathbb{R}^d$ be a d-dimensional feature vector. $\mathcal{I}_N$ stands for the index set $\{1, 2, ..., N\}$. The Frobenius norm of a matrix is given by $\|\mathbf{X}\|_F = \sqrt{\sum\limits_{m,n} X_{mn}^2}$, where X_{mn} represents the (m, n)-th element of $\mathbf{X}$. The spaces of symmetric positive semidefinite and definite matrices are $\mathcal{S}_+^d$ and $\mathcal{S}_{++}^d$. A vector with all coefficients equal one is denoted by $\mathbb{1}$ and $\mathbf{J}_{mn}$ is a matrix of all zeros with one at position (m, n).

3.4.2 Nyström Approximation

In our domain adaptation model, we rely on Nyström projections, thus, we review their general mechanism first.

Proposition 6. *Suppose* $\mathbf{X} \in \mathbb{R}^{d \times N}$ *and* $\mathbf{Z} \in \mathbb{R}^{d \times N'}$ *store* N *feature vectors and* N' *pivots (vectors used in approximation) of dimension* d *in their columns, respectively. Let* $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ *be a positive definite kernel. We form two kernel matrices* $\mathbf{K_{ZZ}} \in \mathcal{S}_{++}^{N'}$ *and* $\mathbf{K_{ZX}} \in \mathbb{R}^{N' \times N}$ *with their* (i, j)*-th elements being* $k(\mathbf{z}_i, \mathbf{z}_j)$ *and* $k(\mathbf{z}_i, \mathbf{x}_j)$*, respectively. Then, the Nyström feature map* $\tilde{\boldsymbol{\Phi}} \in \mathbb{R}^{N' \times N}$*, whose columns correspond to the input vectors in* $\mathbf{X}$*, and the Nyström approximation of kernel* $\mathbf{K_{XX}}$ *for which* $k(\mathbf{x}_i, \mathbf{x}_j)$ *is its* (i, j)*-th entry, are given by:*

$$\tilde{\boldsymbol{\Phi}} = \mathbf{K_{ZZ}}^{-0.5} \mathbf{K_{ZX}} \quad \text{and} \quad \mathbf{K_{XX}} \approx \tilde{\boldsymbol{\Phi}}^T \tilde{\boldsymbol{\Phi}}. \tag{3.1}$$

Proof. See [Bo and Sminchisescu, 2009] for details. □

Remark 1. *The quality of approximation of* (3.1) *depends on the kernel* k*, data points* $\mathbf{X}$*, pivots* $\mathbf{Z}$ *and their number* N'*. In the sequel, we exploit a specific setting under which* $\mathbf{K_{XX}} = \tilde{\boldsymbol{\Phi}}^T \tilde{\boldsymbol{\Phi}}$

which indicates no approximation loss.

3.4.3 Scatter Matrices

We make a frequent use of distances $d^2(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}^*)$ that operate between covariances $\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}(\boldsymbol{\Phi})$ and $\boldsymbol{\Sigma}^* \equiv \boldsymbol{\Sigma}(\boldsymbol{\Phi}^*)$ on feature vectors. Therefore, we provide a useful derivative of $d^2(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}^*)$ w.r.t. feature vectors $\boldsymbol{\Phi}$.

Proposition 7. *Suppose $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N]$ and $\boldsymbol{\Phi}^* = [\boldsymbol{\phi}^*_{1}, ..., \boldsymbol{\phi}^*_{N^*}]$ are some feature vectors of quantity N and N^*,* e.g., *formed by Eq.* (3.1) *and used to evaluate $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^*$ with $\boldsymbol{\mu}$ and $\boldsymbol{\mu}^*$ being the mean of $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$, respectively. Then, derivatives of $d^2 \equiv d^2(\boldsymbol{\Sigma}, \boldsymbol{\Sigma}^*)$ w.r.t. $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ are:*

$$\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \boldsymbol{\Phi}} = \frac{2}{N}\frac{\partial d^2}{\partial \boldsymbol{\Sigma}}\big(\boldsymbol{\Phi} - \boldsymbol{\mu}\mathbb{1}^T\big), \frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \boldsymbol{\Phi}^*} = \frac{2}{N^*}\frac{\partial d^2}{\partial \boldsymbol{\Sigma}^*}\big(\boldsymbol{\Phi}^* - \boldsymbol{\mu}^*\mathbb{1}^T\big). \tag{3.2}$$

Moreover, assume some projection matrix $\boldsymbol{Z}$. Then for $\boldsymbol{\Phi}' = \boldsymbol{Z}[\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N]$ and $\boldsymbol{\Phi}'^* = \boldsymbol{Z}[\boldsymbol{\phi}^*_{1}, ..., \boldsymbol{\phi}^*_{N^*}]$ with covariances $\boldsymbol{\Sigma}'$, $\boldsymbol{\Sigma}'^*$, means $\boldsymbol{\mu}'$, $\boldsymbol{\mu}'^*$ and $d'^2 \equiv d^2(\boldsymbol{\Sigma}', \boldsymbol{\Sigma}'^*)$, we obtain:

$$\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \boldsymbol{\Phi}} = \frac{2\boldsymbol{Z}^T}{N}\frac{\partial d'^2}{\partial \boldsymbol{\Sigma}'}\big(\boldsymbol{\Phi}' - \boldsymbol{\mu}'\mathbb{1}^T\big), \frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \boldsymbol{\Phi}^*} = -\frac{2\boldsymbol{Z}^T}{N^*}\frac{\partial d'^2}{\partial \boldsymbol{\Sigma}'^*}\big(\boldsymbol{\Phi}'^* - \boldsymbol{\mu}'^*\mathbb{1}^T\big). \tag{3.3}$$

Proof. See Appendix C. □

3.4.4 Non-Euclidean Distances

In Table 3.1, we list the distances d with derivatives w.r.t. $\boldsymbol{\Sigma}$ used in the sequel. We indicate properties such as invariance to rotation (*rot.*), affine mainpulations (*aff.*) and inversion (*inv.*). Moreover, we indicate which distances meet the triangle inequality (*Tr. Ineq.*) and which are geodesic distances (*Geo.*). Lastly, we indicate if the distance d and its gradient $\nabla_{\boldsymbol{\Sigma}}$ are finite (*fin.*) or infinite (∞) for $\mathcal{S}_+$ matrices. This last property indicates that JBLD and AIRM distances require some regularization as our covariances are $\mathcal{S}_+$.

3.5 Problem Formulation

In this section, we equip the supervised domain adaptation approach So-HoT [Koniusz et al., 2017] with the JBLD and AIRM distances. Moreover, we show how to use the Nyström projections to make our computations fast.

3.5.1 Supervised Domain Adaptation

Suppose $\mathcal{I}_N$ and $\mathcal{I}_{N^*}$ are the indexes of N source and N^* target training data points. $\mathcal{I}_{N_c}$ and $\mathcal{I}_{N_c^*}$ are the class-specific indexes for $c\in\mathcal{I}_C$, where C is the number of classes (exhibit identities). Furthermore, suppose we have feature vectors from an *fc* layer of the source network stream, one per image, and their associated labels. Such pairs are given by $\boldsymbol{\Lambda}\equiv\{(\boldsymbol{\phi}_n,y_n)\}_{n\in\mathcal{I}_N}$, where $\boldsymbol{\phi}_n\in\mathbb{R}^d$ and $y_n\in\mathcal{I}_C$, $\forall n\in\mathcal{I}_N$. For the target data, by analogy, we define pairs $\boldsymbol{\Lambda}^*\equiv\{(\boldsymbol{\phi}^*_n,y^*_n)\}_{n\in\mathcal{I}^*_N}$, where $\boldsymbol{\phi}^*\in\mathbb{R}^d$ and $y^*_n\in\mathcal{I}_C$, $\forall n\in\mathcal{I}^*_N$. Class-specific sets of feature vectors are given as $\boldsymbol{\Phi}_c\equiv\{\boldsymbol{\phi}^c_n\}_{n\in\mathcal{I}_{N_c}}$ and $\boldsymbol{\Phi}^*_c\equiv\{\boldsymbol{\phi}^{*c}_n\}_{n\in\mathcal{I}_{N^*_c}}$, $\forall c\in\mathcal{I}_C$. Then, $\boldsymbol{\Phi}\equiv(\boldsymbol{\Phi}_1,...,\boldsymbol{\Phi}_C)$ and $\boldsymbol{\Phi}^*\equiv(\boldsymbol{\Phi}^*_1,...,\boldsymbol{\Phi}^*_C)$. Note that we write the asterisk symbol in superscript (*e.g.* $\boldsymbol{\phi}^*$) to denote variables related to the target network while the source-related and generic variables have no such indicator. Figure 3.1 shows our setup. We formulate our problem as a trade-off between the classifier and alignment losses ℓ and $\hbar$:

$$\underset{\substack{\mathbf{W},\mathbf{W}^*,\boldsymbol{\Theta},\boldsymbol{\Theta}^*\\ \text{s. t. } ||\boldsymbol{\phi}_n||_2^2\leq\tau,\\ ||\boldsymbol{\phi}^*_{n'}||_2^2\leq\tau,\\ \forall n\in\mathcal{I}_N,n'\in\mathcal{I}^*_N}}{\arg\min}\quad \ell(\mathbf{W},\boldsymbol{\Lambda})+\ell(\mathbf{W}^*,\boldsymbol{\Lambda}^*)+\eta||\mathbf{W}-\mathbf{W}^*||_F^2+ \underbrace{\frac{\sigma_1}{C}\sum_{c\in\mathcal{I}_C}d_g^2\left(\boldsymbol{\Sigma}_c,\boldsymbol{\Sigma}^*_c\right)+\frac{\sigma_2}{C}\sum_{c\in\mathcal{I}_C}||\boldsymbol{\mu}_c-\boldsymbol{\mu}^*_c||_2^2.}_{\hbar(\boldsymbol{\Phi},\boldsymbol{\Phi}^*)} \tag{3.4}$$

Note that Figure 3.1a indicates by the elliptical/curved shape that $\hbar$ performs the alignment on the $\mathcal{S}_+$ manifold along exact (or approximate) geodesics. For ℓ, we employ a generic loss used by CNNs *e.g.*, Softmax. For the source and target streams, the matrices $\mathbf{W},\mathbf{W}^*\in\mathbb{R}^{d\times C}$ contain unnormalized probabilities (c.f. hyperplanes of two SVMs). In Equation (3.4), separating the class-specific distributions is addressed by ℓ while attracting the within-class scatters of both network streams is handled by $\hbar$. Variable η controls the proximity between $\mathbf{W}$ and $\mathbf{W}^*$ which encourages the similarity between decision boundaries of classifiers.

Our loss $\hbar$ depends on two sets of variables $(\boldsymbol{\Phi}_1,...,\boldsymbol{\Phi}_C)$ and $(\boldsymbol{\Phi}^*_1,...,\boldsymbol{\Phi}^*_C)$ – one set per network stream. Feature vectors $\boldsymbol{\Phi}(\boldsymbol{\Theta})$ and $\boldsymbol{\Phi}^*(\boldsymbol{\Theta}^*)$ depend on the parameters of the source and target network streams $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}^*$ that we optimize over. $\boldsymbol{\Sigma}_c\equiv\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}_c))$, $\boldsymbol{\Sigma}^*_c\equiv\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}^*_c))$, $\boldsymbol{\mu}_c(\boldsymbol{\Phi})$ and $\boldsymbol{\mu}^*_c(\boldsymbol{\Phi}^*)$ denote the covariances and means, respectively, one covariance/mean pair per network stream per class. Coeffs. σ_1, σ_2 control the degree of the scatter and mean alignment, τ controls the ℓ_2-norm of feature vectors.

The Nyström projections are denoted by $\boldsymbol{\Pi}$. Table 3.1 indicates that back-propagation on the JBLD and AIRM distances involves inversions of $\boldsymbol{\Sigma}_c$ and $\boldsymbol{\Sigma}^*$ to be performed for each $c\in\mathcal{I}_C$ according to (3.4). As these covariances are formed from 4096 dimensional feature vectors of the *fc* layer, such inversions are too costly to run fine-tuning *e.g.*, 4*s* per iteration is prohibitive. Thus, we demonstrate next how the Nyström projections can be combined with d_g.

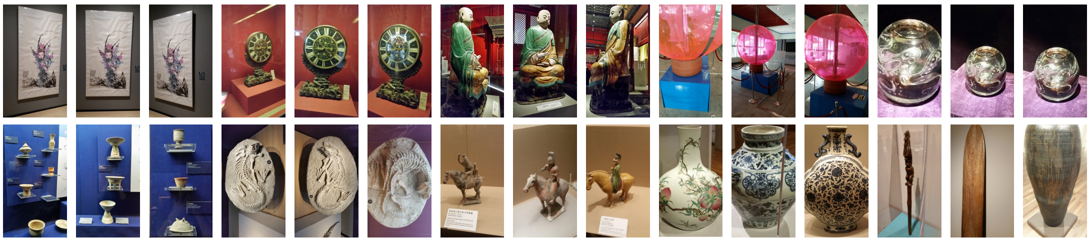


Figure 3.2: Examples of the source subsets of Open MIC. Top row includes Paintings (*Shn*), Clocks (*Shg*), Sculptures (*Scl*), Science Exhibits (*Sci*) and Glasswork (*Gls*). As 3 images per exhibit demonstrate, we covered different viewpoints and scales during capturing. Bottom row includes 3 different art pieces per exhibition such as Cultural Relics (*Rel*), Natural History Exhibits (*Nat*), Historical/Cultural Exhibits (*Shx*), Porcelain (*Clv*) and Indigenous Arts (*Hon*). Note the composite scenes of Relics, fine-grained nature of Natural History and Cultural Exhibits and non-planarity of exhibits.

Proposition 8. *Let us choose* $\mathbf{Z}=\mathbf{X}=\ [\boldsymbol{\Phi},\boldsymbol{\Phi}^*]$ *for pivots and source/target feature vectors, and kernel* k *to be linear. Substitute these assumptions into Eq.* (3.1)*. As a result, we obtain* $\boldsymbol{\Pi}(\mathbf{X})=(\mathbf{Z}^T\mathbf{Z})^{-0.5}\mathbf{Z}^T\mathbf{X}=\mathbf{Ƶ}\mathbf{X}=(\mathbf{Z}^T\mathbf{Z})^{0.5}=(\mathbf{X}^T\mathbf{X})^{0.5}$ *where* $\boldsymbol{\Pi}(\mathbf{X})$ *is a projection of* $\mathbf{X}$ *on itself that is isometric* e.g.*, distances between column vectors of* $(\mathbf{X}^T\mathbf{X})^{0.5}$ *correspond to distances of column vectors in* $\mathbf{X}$*. Thus,* $\boldsymbol{\Pi}(\mathbf{X})$ *is an isometric transformation w.r.t. distances in Table 3.1, that is* $d_g^2(\boldsymbol{\Sigma}(\boldsymbol{\Phi}),\boldsymbol{\Sigma}(\boldsymbol{\Phi}^*))=d_g^2(\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi})),\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}^*)))$*.*

Proof. Firstly, we note that the following holds:

$$\mathbf{K}_{\mathbf{XX}}=\boldsymbol{\Pi}(\mathbf{X})^T\boldsymbol{\Pi}(\mathbf{X})=(\mathbf{X}^T\mathbf{X})^{0.5}(\mathbf{X}^T\mathbf{X})^{0.5}=\mathbf{X}^T\mathbf{X}. \tag{3.5}$$

Note that $\boldsymbol{\Pi}(\mathbf{X})=\mathbf{Ƶ}\mathbf{X}$ projects $\mathbf{X}$ into a more compact subspace of size $d'=N+N^*$ if $d'\ll d$ which includes the spanning space for $\mathbf{X}$ by construction as $\mathbf{Z}=\ \mathbf{X}$. Eq. (3.5) implies that $\boldsymbol{\Pi}(\mathbf{X})$ performs at most rotation on $\mathbf{X}$ as the dot-product (used to obtain entries of $\mathbf{K}_{\mathbf{XX}}$) just like the Euclidean distance is rotation-invariant only *e.g.*, has no affine invariance. As spectra of $(\mathbf{X}^T\mathbf{X})^{0.5}$ and $\mathbf{X}$ are equal, this implies $\boldsymbol{\Pi}(\mathbf{X})$ performs no scaling, shear or inverse. Distances in Table 3.1 are all rotation-invariant, thus $d_g^2(\boldsymbol{\Sigma}(\boldsymbol{\Phi}),\boldsymbol{\Sigma}(\boldsymbol{\Phi}^*))=d_g^2(\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi})),\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}^*)))$.

A stricter proof is to show that $\mathbf{Ƶ}$ performs a composite rotation $\boldsymbol{V}\boldsymbol{U}^T$. Let us use SVD of $\mathbf{Z}$ equal $\boldsymbol{U}\boldsymbol{\lambda}\boldsymbol{V}^T$. Then:

$$\begin{aligned}\mathbf{Ƶ}=(\mathbf{Z}^T\mathbf{Z})^{-0.5}\mathbf{Z}^T&=\left(\boldsymbol{V}\boldsymbol{\lambda}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\lambda}\boldsymbol{V}^T\right)^{-0.5}\boldsymbol{V}\boldsymbol{\lambda}\boldsymbol{U}^T\\&=\boldsymbol{V}\boldsymbol{\lambda}^{-1}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\lambda}\boldsymbol{U}^T=\boldsymbol{V}\boldsymbol{U}^T\end{aligned} \tag{3.6}$$

□

In practice, for each class $c\in\mathcal{I}_C$, we choose $\mathbf{X}=\mathbf{Z}=\ [\boldsymbol{\Phi}_c,\boldsymbol{\Phi}_c^*]$. Then, as $\mathbf{Ƶ}[\boldsymbol{\Phi},\boldsymbol{\Phi}^*]=(\mathbf{X}^T\mathbf{X})^{0.5}$, we have $\boldsymbol{\Pi}(\boldsymbol{\Phi})=[\mathbf{y}_1,...,\mathbf{y}_N]$ and $\boldsymbol{\Pi}(\boldsymbol{\Phi}^*)=[\mathbf{y}_{N+1},...,\mathbf{y}_{N+N*}]$ where $\mathbf{Y}=[\mathbf{y}_1,...,\mathbf{y}_{N+N*}]=(\mathbf{X}^T\mathbf{X})^{0.5}$. With typical $N\approx 30$ and $N^*\approx 3$, we obtain covariances of side size $d'\approx 33$ rather than $d=4096$.

Proposition 9. *Typically, the inverse square root* $(\mathbf{X}^T\mathbf{X})^{-0.5}$ *of* $\boldsymbol{\mathcal{Z}}(\mathbf{X})$ *can be only differentiated via the costly eigenvalue decomposition. However, if* $\mathbf{X}=[\boldsymbol{\Phi},\boldsymbol{\Phi}^*]$, $\boldsymbol{\mathcal{Z}}(\mathbf{X})=(\mathbf{X}^T\mathbf{X})^{-0.5}\mathbf{X}^T$ *and* $\boldsymbol{\Pi}(\mathbf{X})=\boldsymbol{\mathcal{Z}}(\mathbf{X})\mathbf{X}$ *as in Prop. 8, and if we consider the chain rule we require:*

$$\frac{\partial d_g^2(\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi})),\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}^*)))}{\partial\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}))}\odot\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}))}{\partial\boldsymbol{\Pi}(\boldsymbol{\Phi})}\odot\frac{\partial\boldsymbol{\Pi}(\boldsymbol{\Phi})}{\partial\boldsymbol{\Phi}},\text{[1]} \tag{3.7}$$

then $\boldsymbol{\mathcal{Z}}(\mathbf{X})$ *can be treated as a constant in differentiation:*

$$\frac{\partial\boldsymbol{\Pi}(\mathbf{X})}{\partial X_{mn}}=\frac{\partial\boldsymbol{\mathcal{Z}}(\mathbf{X})\mathbf{X}}{\partial X_{mn}}=\boldsymbol{\mathcal{Z}}(\mathbf{X})\frac{\partial\mathbf{X}}{\partial X_{mn}}=\boldsymbol{\mathcal{Z}}(\mathbf{X})\mathbf{J}_{mn}. \tag{3.8}$$

Proof. It follows from the rotation-invariance of the Euclidean, JBLD and AIRM distances. Let us write $\boldsymbol{\mathcal{Z}}(\mathbf{X})=\boldsymbol{R}(\mathbf{X})=\boldsymbol{R}$, where $\boldsymbol{R}$ is a rotation matrix. Thus, we have: $d_g^2(\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi})),\boldsymbol{\Sigma}(\boldsymbol{\Pi}(\boldsymbol{\Phi}^*)))=d_g^2(\boldsymbol{\Sigma}(\boldsymbol{R\Phi}),\boldsymbol{\Sigma}(\boldsymbol{R\Phi}^*))=d_g^2(\boldsymbol{R\Sigma}(\boldsymbol{\Phi})\boldsymbol{R}^T,\boldsymbol{R\Sigma}(\boldsymbol{\Phi}^*)\boldsymbol{R}^T)$. Therefore, even if $\boldsymbol{R}$ depends on $\mathbf{X}$, the distance d_g^2 is unchanged by any choice of valid $\boldsymbol{R}$ *i.e.*, for the Frobenius norm we have: $||\boldsymbol{R\Sigma R}^T-\boldsymbol{R\Sigma}^*\boldsymbol{R}^T||_F^2=\operatorname{Tr}\left(\boldsymbol{RA}^T\boldsymbol{R}^T\boldsymbol{RAR}^T\right)=\operatorname{Tr}\left(\boldsymbol{R}^T\boldsymbol{RA}^T\boldsymbol{A}\right)=\operatorname{Tr}\left(\boldsymbol{A}^T\boldsymbol{A}\right)=||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2$, where $\boldsymbol{A}=\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*$. Therefore, we obtain: $\frac{\partial||\boldsymbol{R\Sigma}(\boldsymbol{\Phi})\boldsymbol{R}^T-\boldsymbol{R\Sigma}(\boldsymbol{\Phi}^*)\boldsymbol{R}^T||_F^2}{\partial\boldsymbol{R\Sigma}(\boldsymbol{\Phi})\boldsymbol{R}^T}\odot\frac{\partial\boldsymbol{R\Sigma}(\boldsymbol{\Phi})\boldsymbol{R}^T}{\partial\boldsymbol{\Sigma}(\boldsymbol{\Phi})}\odot\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\Phi})}{\partial\boldsymbol{\Phi}}=\frac{\partial||\boldsymbol{\Sigma}(\boldsymbol{\Phi})-\boldsymbol{\Sigma}(\boldsymbol{\Phi}^*)||_F^2}{\partial\boldsymbol{\Sigma}(\boldsymbol{\Phi})}\odot\frac{\partial\boldsymbol{\Sigma}(\boldsymbol{\Phi})}{\partial\boldsymbol{\Phi}}$ [1]which completes the proof. □

Complexity. The Frobenius norm between covariances plus their computation have combined complexity $\mathcal{O}((d'+1)d^2)$, where $d'=N+N^*$. For non-Euclidean distances, we take into account the dominant cost of evaluating the square root of matrix and/or inversions by the eigenvalue decomposition, as well as the cost of building scatter matrices. Thus, we have $\mathcal{O}((d'+1)d^2+d^\omega)$, where constant $2<\omega<2.376$ concerns complexity of eigenvalue decomposition. Lastly, evaluating the Nyström projections combined with building covariances and running a non-Euclidean distance enjoys $\mathcal{O}(d'^2d+(d'+1)d'^2+d'^\omega)=\mathcal{O}(d'^2d)$ complexity for $d\gg d'$.

For typical $d'=33$ and $d=4096$, the non-Euclidean distances are 1.7× slower[2] than the Frobenius norm. However, non-Eucldiean distances combined with our projections are 210× and 124× faster than naively evaluated non-Eucldiean distances and the Frobenius norm, resp. This cuts the time of each training from few days to 6–8 hours and makes the cost of our loss negligible compared to CNN fine-tuning.

[1] For simplicity of notation, operator $\odot$ denotes the typical summation over multiplications in chain rules.

[2] We assume that the eigenvalue decomposition of large matrices ($d=4096$) in CUDA BLAS is fast and efficient–which is not the case.

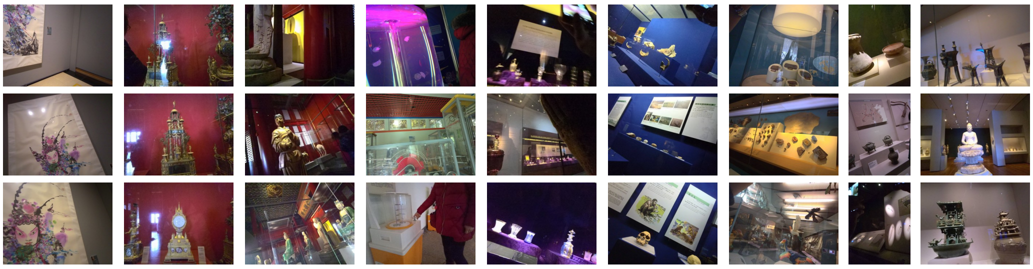


Figure 3.3: Examples of the target subsets of Open MIC. From left to right, each column illustrates Paintings (*Shn*), Clocks (*Shg*), Sculptures (*Scl*), Science Exhibits (*Sci*) and Glasswork (*Gls*), Cultural Relics (*Rel*), Natural History Exhibits (*Nat*), Historical/Cultural Exhibits (*Shx*) and Porcelain (*Clv*). Note the variety of photometric and geometric distortions due to the use of wearable cameras.

3.6 Experiments

In this section, we explain our CNN setup and give more details about our Open MIC and present our evaluations.

Setting. At the training and testing time, we use the setting shown in Figures 3.1a and 3.1b, respectively. The images in our dataset are portrait or landscape oriented. Therefore, we extract 3 square patches per image that cover its entire region. For training, these patches serve as training data points. For testing, we average over 3 predictions from a group of patches to label image. We briefly compare the VGG16 [Simonyan and Zisserman, 2014b] and GoogLeNet networks [Szegedy et al., 2015] as well as the Eucldiean, JBLD and AIRM distances on subsets of the Office and Open MIC dataset. As demonstrated in Table 3.3, VGG16 and GoogLeNet yield similar scores while JBLD and AIRM beat the Euclidean distance. Thus, we employ the VGG16 model and the JBLD distance in what follows.

Parameters. The networks are pre-trained on the ImageNet dataset [Russakovsky et al., 2015] for the best results. We set non-zero learning rates on the fully-connected and the last two convolutional layers of the two streams. Subsequently, fine-tuning on the source and target data takes between 30–100K iterations. We set τ to the average value of the ℓ_2 norm of fc feature vectors sampled on ImageNet and the hyperplane proximity $\eta = 1$. Inverse in $\mathbf{Ƶ}(\mathbf{X}) = (\mathbf{X}^T\mathbf{X})^{-0.5}\mathbf{X}^T$ and matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^*$ are regularized with a small constant 1e-6 on diagonals. Lastly, we set σ_1 and σ_2 between 0.005–1 to perform cross-validation.

Office. This dataset contains three domains: Amazon, DSLR and Webcam. The Amazon and Webcam domains contain 2817 and 795 images. For brevity, we first test our pipeline on the Amazon-Webcam domain shift ($\mathcal{A} \rightarrow \mathcal{W}$) to ensure that we match results in the literature.

Open MIC. The proposed dataset contains 10 distinct source-target subsets of images from 10 different kinds of museum exhibition spaces which are illustrated in Figures 3.2 and 3.3, respectively. They include Paintings from Shenzhen Museum (*Shn*), the Clock and Watch Gallery (*Clk*) and the Indian and Chinese Sculptures (*Scl*) from the Palace Museum, the Xi-

	Shn	*Clk*	*Scl*	*Sci*	*Gls*	*Rel*	*Nat*	*Shx*	*Clv*	*Hon*	Total
Inst.	79	113	41	37	98	100	111	166	81	40	866
Src+	566	413	225	637	601	775	763	2928	531	1121	8560
Src.	417	650	160	391	575	587	695	2697	503	970	7645
Tgt+	515	323	130	1692	964	1229	868	776	682	417	7596
Tgt.	404	305	112	1342	863	863	668	546	625	364	6092

Table 3.2: Unique exhibit instances (*Inst.*) and numbers of images of Open MIC in the source (*Src.*) and target (*Tgt.*) subsets including their backgrounds (*Src*+) and (*Tgt*+).

angyang Science Museum (*Sci*), the European Glass Art (*Gls*) and the Collection of Cultural Relics (*Rel*) from the Hubei Provincial Museum, the Nature, Animals and Plants in Ancient Times (*Nat*) from Shanghai Natural History Museum, the Comprehensive Historical and Cultural Exhibits from Shaanxi History Museum (*Shx*), the Sculptures, Pottery and Bronze Figurines from the Cleveland Museum of Arts (*Clv*), and Indigenous Arts from Honolulu Museum Of Arts (*Hon*).

For the target data, we annotated each image with labels of art pieces visible in it. The wearable cameras were set to capture an image every 10s and they operated *in-the-wild*, *e.g.*, volunteers had no control over shutter, focus, centering, *etc*. Therefore, the collected target subsets exhibit many realistic challenges, *e.g.*, sensor noises, motion blur, occlusions, background clutter, varying viewpoints, scale changes, rotations, glares, transparency, non-planar surfaces, clipping, multiple exhibits, active light, color inconstancy, very large or small exhibits, to name but a few phenomena visible in Figure 3.3. The numbers and statistics regarding the Open MIC dataset are given in Table 3.2. Every subset contains 37–166 exhibits to identify and 5 train, val., and test splits. In total, our dataset contains 866 unique exhibit labels, 8560 source (7645 exhibits and 915 backgrounds) and 7596 target (6092 exhibits and 1504 backgrounds including a few of unidentified exhibits) images.

Baselines. To demonstrate the intrinsic difficulty of the Open MIC dataset, we provide the community with baseline accuracies obtained from (i) fine-tuning CNNs on the source subsets (*S*) and testing on the randomly chosen target splits, (ii) fine tuning on target only (*T*) and evaluating on remaining disjoint target splits, (iii) fine-tuning on the source+target (*S*+*T*) and evaluating on remaining disjoint target splits, (iv) training state-of-the-art domain adaptation So-HoT algorithm [Koniusz et al., 2017] equipped by us with non-Euclidean distances [Cherian et al., 2012; Pennec et al., 2006; Bhatia, 2009] to enable robust end-to-end learning.

We include evaluation protocols: (i) training/eval. per exhibition subset, (ii) training/testing on the combined set with all 866 identity labels, (iii) testing w.r.t. scene factors annotated by us and detailed in Section 3.6.2 (Challenge III).

	VGG16	GoogLe Net
S+T	88.66	88.92
So	89.45	89.70
JBLD	**90.80**	**91.33**
AIRM	90.72	91.20

DLID	[Chopra et al., 2013]	51.9
DeCAF$_6$ S+T	[Donahue et al., 2014]	80.7
DaNN	[Ghifary et al., 2014]	53.6
Source CNN	[Tzeng et al., 2015]	56.5
Target CNN	[Tzeng et al., 2015]	80.5
Source+Target CNN	[Tzeng et al., 2015]	82.5
Dom. Conf.+Soft Labs.	[Tzeng et al., 2015]	82.7

Table 3.3: The Office dataset ($\mathcal{A}\rightarrow\mathcal{D}$ domain shift). (Left) Results on the VGG16 and GoogLeNet streams for the baseline fine-tuning on the combined source+target domains (*S*+*T*) and second-order (*So*) Euclidean-based method [Koniusz et al., 2017] are compared to our JBLD/AIRM dist. (Right) Comparisons to the state of the art.

	sp1	sp2	sp3	sp4	sp5	top-1	top-1-5	top-5	top-5-5	Avg$_k$ top-k-k
S	33.9	34.2	34.8	34.2	33.8	34.2	36.0	49.2	53.7	46.0
T	56.9	55.9	58.7	56.0	55.2	56.5	64.1	76.5	80.6	72.5
S+T	56.4	55.2	57.1	56.3	54.4	55.9	62.5	75.8	79.2	71.6
So	64.2	62.4	65.0	62.7	60.0	62.8	70.4	84.0	88.5	79.5
JBLD	**65.7**	**63.8**	**65.7**	**63.7**	**62.0**	**64.2**	**72.0**	**85.7**	**88.6**	**80.8**

Table 3.4: Challenge II. Open MIC performance on the combined set for data 5 splits. Baselines (*S*), (*T*) and (*S*+*T*) are given as well as second-order (*So*) method [Koniusz et al., 2017] and our JBLD approach.

3.6.1 Comparison to the State of the Art

Firstly, we validate that our reference method performs on the par or better than the state-of-the-art approaches. Table 3.3 shows that the JBLD and AIRM distances outperform the Euclidean-based So-HoT method (*So*) [Koniusz et al., 2017] by $\sim$1.6% and other recent approaches *e.g.*, [Tzeng et al., 2015] by $\sim$ 8.6% accuracy. We also observe that GoogLeNet outperforms the VGG16-based model by $\sim$0.5%. Having validated our model, we opt to evaluate our proposed Open MIC dataset on VGG16 streams for consistency with the So-HoT model [Koniusz et al., 2017].

3.6.2 Open MIC Challenge

In what follows, we detail our challenges on the Open MIC dataset and present our experimental results.

Challenge I. For this challenge, we run our supervised domain adaptation algorithm combined with the JBLD distance per subset. We prepare 5 training, validation and testing splits. For the source data, we use all available samples per class. For the target data, we use 3 samples per class for training and validation, respectively, and the rest for testing.

We report top-1 and top-5 accuracies. Moreover, as our target images often contain multiple exhibits, we ask a question whether any of top-k predictions match any of top-n image labels ordered by our expert volunteers according to the perceived saliency. If so, we count it as a correctly recognized image. We count these valid predictions and normalize by the total number of testing images. We denote this measure as top-k-n where $k,n \in \mathcal{I}_5$. Lastly, we indicate an *area-under-curve* type of measure Avg$_k$ top-k-k which rewards correct recognition of the most dominant object in the scene and offers some leniency if the order of top predictions is confused and/or if they match less dominant objects–a simple alternative to precision/recall plots.

We divided Open MIC into *Shn*, *Clk*, *Scl*, *Sci*, *Gls*, *Rel*, *Nat*, *Shx*, *Clv* and *Hon* subsets to allow short 6–8 hours long runs per experiment. We ran 150 jobs on (*S*), (*T*) and (*S*+*T*) baselines and 300 jobs on JBLD: 5 splits ×10 subsets ×6 hyperp. choices. Table 3.5 shows that the exhibits in the Comprehensive Historical and Cultural Exhibits (*Shx*) and the Sculptures (*Scl*) were the hardest to identify given scores of 48.5 and 54.4% top-1 accuracy. This is consistent with volunteers' reports that both exhibitions were crowded, the lighting was dim, exhibits were occluded, fine-grained and non-planar. The easiest to identify were the Sculptures, Pottery and Bronze Figurines (*Clv*) and the Indigenous Arts (*Hon*) as both exhibitions were spacious with good lighting. The average top-1 accuracy across all subsets on JBLD is 63.9%. Averages over baselines (*S*), (*T*) and (*S*+*T*) are 52.5, 57.4, and 58.5% top-1 acc. To account for uncertainty of saliency-based labeling and classifier confusing which exhibit to label, we report our proposed average top-1-5 acc. to be 70.6%. Our average combined score Avg$_k$ top-k-k is 79.3%. These results show that Open MIC challenges CNNs due to *in-the-wild* capture with wearable cameras.

Challenge II. Having provided the above results per subset, we evaluate the combined set covering 866 exhibit identities. In this setting, a single experiment runs 80–120 hours. We ran 15 jobs on (*S*), (*T*) and (*S*+*T*) baselines and 60 jobs on (*So*) and JBLD: 2 distances ×5 splits ×6

		S	*T*	*S*+*T*	*JBLD*		*S*	*T*	*S*+*T*	*So*	*JBLD*	*AIRM*		*S*	*T*	*S*+*T*	*JBLD*		*S*	*T*	*S*+*T*	*JBLD*		*S*	*T*	*S*+*T*	*JBLD*
sp1	*Shn*	45.3	45.3	59.0	**60.0**	*Clk*	55.8	51.9	55.8	55.8	**57.7**	57.2	*Scl*	56.5	60.9	65.2	**65.2**	*Sci*	59.3	58.9	65.6	**65.8**	*Gls*	64.1	67.1	62.8	**70.3**
sp2		48.4	52.6	53.7	**62.1**		55.4	44.6	50.0	58.9	**58.9**	58.9		44.4	50.0	44.4	**50.0**		56.9	57.2	67.1	**69.1**		59.9	61.9	59.2	**63.9**
sp3		46.1	52.7	60.4	**64.8**		58.9	58.9	67.9	69.6	**71.4**	71.4		55.6	38.9	44.4	**44.4**		69.9	62.0	65.7	**68.2**		65.9	69.3	64.9	**69.6**
sp4		49.5	50.5	54.8	**64.5**		51.9	48.1	46.1	53.8	**57.7**	57.7		55.0	55.0	55.0	**50.0**		58.1	59.2	64.2	**66.3**		62.3	67.0	61.6	**68.7**
sp5		49.5	57.0	63.4	**69.9**		62.5	41.7	60.4	58.3	**60.4**	60.4		56.2	56.2	62.5	**62.5**		57.3	53.3	61.5	**64.5**		60.1	64.5	59.0	**65.2**
top-1		47.7	51.6	58.3	**64.3**		56.9	49.1	56.0	59.3	**61.2**	61.1		53.5	52.2	54.3	**54.4**		58.5	58.1	64.9	**66.8**		62.5	65.9	61.6	**67.5**
top-1-5		48.2	54.2	60.2	**66.4**		58.9	56.3	60.3	66.2	**68.9**	68.9		54.7	55.4	57.3	**58.4**		60.2	61.7	67.8	**70.2**		77.3	84.4	76.7	**84.9**
top-5		64.5	68.8	76.9	**81.6**		76.7	63.8	78.2	**87.5**	86.9	87.2		67.4	66.6	70.0	**70.0**		83.3	82.7	86.0	**88.6**		85.2	89.4	83.1	**89.3**
top-5-5		66.0	73.3	79.5	**84.2**		77.8	75.0	82.7	**91.6**	91.0	91.4		69.4	69.8	71.1	**72.0**		85.6	86.3	89.4	**91.3**		87.3	95.0	89.7	**93.4**
Avg$_k$ top-k-k		59.0	63.4	71.0	**76.6**		69.4	65.6	73.6	**81.5**	81.2	81.4		63.7	62.5	65.1	**65.1**		75.3	76.0	80.7	**82.5**		78.4	86.2	80.7	**86.2**

		S	*T*	*S*+*T*	*JBLD*		*S*	*T*	*S*+*T*	*So*	*JBLD*	*AIRM*		*S*	*T*	*S*+*T*	*JBLD*		*S*	*T*	*S*+*T*	*JBLD*		*S*	*T*	*S*+*T*	*JBLD*
sp1	*Rel*	62.0	65.0	63.3	**66.3**	*Nat*	38.0	56.2	52.6	58.8	**58.8**	58.5	*Shx*	33.3	43.2	31.5	**58.6**	*Clv*	47.4	65.8	66.2	**71.4**	*Hon*	65.6	71.1	70.3	**75.8**
sp2		60.9	65.7	63.0	**68.0**		39.9	52.5	52.5	59.6	**59.6**	59.6		31.8	39.8	27.4	**47.8**		47.0	70.2	65.1	**72.2**		63.9	67.2	70.5	**74.6**
sp3		64.1	70.4	67.4	**70.7**		43.7	56.2	59.4	59.9	**59.9**	59.9		25.7	47.7	31.2	**47.7**		49.7	64.1	61.5	**67.7**		68.5	70.2	71.8	**79.0**
sp4		61.0	68.5	62.8	**67.1**		41.8	59.8	62.0	66.3	**67.9**	67.4		33.0	38.8	26.2	**44.7**		48.3	63.0	64.0	**68.5**		67.8	63.6	79.3	**76.9**
sp5		55.4	61.0	59.3	**62.6**		44.6	62.0	63.0	66.8	**67.4**	66.8		25.7	35.8	28.4	**44.0**		42.3	62.8	54.1	**65.8**		67.5	65.8	75.0	**80.0**
top-1		60.7	66.1	63.2	**67.0**		41.6	57.3	57.9	62.2	**62.7**	62.5		29.9	41.1	29.0	**48.5**		47.0	65.2	62.2	**69.1**		66.7	67.6	73.4	**77.3**
top-1-5		70.1	76.8	73.2	**79.5**		43.5	62.8	61.9	67.3	**67.7**	67.5		31.5	47.7	31.9	**56.3**		50.8	69.5	66.6	**73.9**		70.2	70.3	76.3	**79.7**
top-5		82.0	87.1	85.8	**90.3**		60.6	79.3	75.5	**84.6**	84.3	84.3		51.6	62.5	51.2	**75.0**		65.3	84.3	79.9	**87.7**		82.1	85.2	88.3	**90.0**
top-5-5		86.3	90.0	89.4	**93.7**		65.3	82.8	80.1	**87.5**	87.0	86.9		54.9	67.3	54.8	**77.6**		70.5	89.2	84.4	**91.0**		88.1	88.8	91.7	**92.7**
Avg$_k$ top-k-k		77.4	82.8	80.5	**85.2**		55.7	74.0	72.4	79.5	**79.6**	79.4		45.1	57.1	44.5	**66.8**		61.5	80.6	76.5	**83.5**		79.7	81.0	84.5	**86.7**

Table 3.5: Challenge I. Open MIC performance on the 10 subsets for data 5 splits. Baselines (*S*), (*T*) and (*S*+*T*) are given as well as our JBLD approach. We report top-1, top-1-5, top-5-1, top-5-5 accuracies and the combined scores Avg$_k$ top-k-k. See Section3.6.2 for details.

	clp	*lgt*	*blr*	*glr*	*bgr*	*ocl*	*rot*	*zom*	*vpc*	*sml*	*shd*	*rfl*	*ok*
S	41.4	17.0	23.8	27.3	40.3	34.5	29.7	52.7	33.4	14.2	10.4	32.3	65.5
T	56.2	38.2	42.6	56.1	57.9	49.6	58.3	60.4	50.3	29.6	59.2	60.7	64.3
S+*T*	56.6	34.6	39.8	54.9	56.2	48.3	56.7	65.9	48.7	27.3	56.5	59.0	72.6
JBLD	**65.3**	**48.6**	**51.6**	**64.0**	**65.9**	**56.4**	**65.0**	**70.0**	**58.6**	**34.1**	**70.4**	**67.5**	**81.0**

Table 3.6: Challenge III. Open MIC performance on the combined set w.r.t. 12 factors detailed in Section 3.6.2. Top-1 accuracies for baselines (*S*), (*T*), (*S*+*T*), and for our JBLD approach are listed.

hyperp. choices. Table 3.4 shows that our JBLD approach scores 64.2% top-1 accuracy and outperforms baselines (*S*), (*T*) and (*S*+*T*) by 30, 7.7 and 8.3%. Fine-tuning CNNs on the source and testing on target (*S*) is especially a very poor performer due to the significant domain shift in Open MIC.

Challenge III. For this challenge, we break down performance on the combined set covering 866 exhibit identities w.r.t. the following 12 factors: object clipping (*clp*), low lighting (*lgt*), blur (*blr*), light glares (*glr*), background clutter (*bgr*), occlusions (*ocl*), in-plane rotations (*rot*), zoom (*zom*), tilted viewpoint (*vpc*), small size/far away (*sml*), object shadows (*shd*), reflections (*rfl*) and the clean view (*ok*). Table 3.6 shows results averaged over 5 data splits. We note that JBLD outperforms baselines. The factors most affecting the supervised domain adaptation are the small size (*sml*) of exhibits/distant view, low light (*lgt*) and blur (*blr*). The corresponding top-1 accuracies of 34.1, 48.6 and 51.6% are below our average top-1 accuracy of 64.2% listed in Table 3.4. In contrast, images with shadows (*shd*), zoom (*zom*) and reflections (*rfl*) score 70.4, 70.0 and 67.5% top-1 accuracy (above avg. 64.2%). Our wearable cameras captured also a few of clean shots scoring 81.0% top-1 accuracy. This lets us form a claim that domain adaptation methods should evolve to deal with each of these adverse factors.

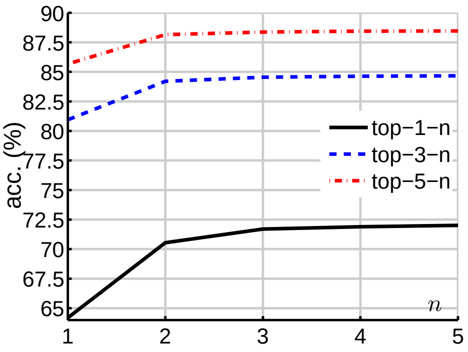


Figure 3.4: Challenge II. Open MIC performance on the combined set. In the plot, we list the mean top-*k*-*n* accuracy (averaged over 5 data splits) w.r.t. k and n for our JBLD approach. We vary $k \in \{1, 3, 5\}$ and $n \in \mathcal{I}_5$.

3.6.3 Additional Results on the OpenMIC dataset

Below, we give more details about our Open MIC dataset and present more evaluations. Table 3.7 contains a more detailed description of the 12 factors which we use to analyze performance

abbr.	details
clp	object clipping *e.g.*, side, base or top including small or large fragments of an exhibit
lgt	poor lighting *e.g.*, dark exhibition space, dark exhibit casing, strong light sources to which camera adapted leaving exhibit underexposed
blr	blur due to motion and/or poor lighting/long shutter exposure; full blur or part of the exhibit affected
glr	point-wise glares of light reflected from objects
bgr	background clutter: a non-uniform background behind an exhibit that changes with the camera viewpoint *e.g.*, people, other exhibits, furniture *etc*
ocl	side, frontal, large or partial exhibit occlusions due to humans, other objects or non-transparent protective casing
rot	in-plane rotations by more than 5 degrees due to a tilted camera or volunteers leaning towards exhibits
zom	large close-ups of an exhibit or a zoom of a part of exhibit
vpc	camera viewpoint that mismatches the normal to the surface of face of an exhibit–some exhibits have no frontal face, some have several faces due to their distinct axes of symmetry
sml	small object: an exhibit captured at a large distance *e.g.*, across a hall; also small scale exhibits which cannot be closely approached
shd	a shadow cast over part of an exhibit
rfl	reflections affecting surfaces such as a protective glass casing of exhibits which acts like a mirror
ok	no visible distortions listed above

Table 3.7: Challenge III. The 12 factors w.r.t. which we evaluate our dataset.

abbr.	details
lcl	light object clipping *e.g.*, side, base or top including small fragments below 20% of the exhibit area
hcl	heavy object clipping of large fragments *e.g.*, more than 20% of the exhibit area
bcl	clipping of the base of sculptures/exhibits *etc*
scl	side occlusions of exhibits by humans or other objects
fcl	frontal/central occlusions of exhibit by humans or other objects
ooc	unclassified kind of occlusion
lzo	close-ups of an exhibit
hzo	large close-ups or a heavy zoom on a part of exhibit
lro	small in-plane rotations by no more than 15 degrees due to a tilted camera *etc.*
hro	large in-plane rotations by more than 15 degrees due to a tilted camera *etc.*
lvp	mismatches by less than 15 degrees between the camera viewpoint and the normal to the surface of face of an exhibit
hvp	mismatches by more than 15 degrees between the camera viewpoint and the normal to the surface of face of an exhibit
spc	light specularities and other reflections from surface

Table 3.8: Challenge III. Additional factors w.r.t. which we evaluate our dataset.

on our Open MIC dataset. Additionally to the Table 6 in the main submission, which breaks down the performance w.r.t. these 12 factors, we performed an analysis w.r.t. pairs of factors.

Tables 3.9 and 3.10 present the image counts and results w.r.t. pairs of factors co-occurring together. The combination of (*sml*) with (*glr*), (*blr*), (*bgr*), (*lgt*), (*rot*) and (*vpc*) results in 13.5, 21.0, 29.9, 31.2, 32.6 and 33.2% mean top-1 accuracy, respectively. Therefore, these pairs of factors affect the quality of recognition the most.

Tables 3.11 and 3.12 present the image counts and results w.r.t. triplets of factors co-occurring together. To obtain these results, we first selected 12 pairs of most challenging co-occurring factors in Table 3.10 and then we further combined them with the 12 main factors from Table 3.7 to obtain triplets. As can be seen, (*sml*+*glr*+*lgt*) and (*sml*+*glr*+*blr*) combinations of factors were the most difficult to recognize and resulted in 0% accuracy. Moreover, (*sml*+*bgr*+*glr*), (*sml*+*ocl*+*lgt*), (*sml*+*rot*+*glr*) and (*sml*+*blr*+*ocl*) resulted in 7.7, 10.0, 10.5, and 11.1% accuracy which also highlights the difficult nature of these combinations of factors in domain adaptation and recognition.

Tables 3.8 presents additional factors that we use in our analysis. We split (*clp*), (*rot*), (*vpc*) and (*zoo*) into their light and heavy variants. We also split (*occ*) into the side and frontal

∩	*clp*	*lgt*	*blr*	*glr*	*bgr*	*ocl*	*rot*	*zom*	*vpc*	*sml*	*shd*	*rfl*	*ok*
all	5136	335	1728	1346	2290	1529	7344	2278	4571	557	125	2000	84
clp	5136	216	770	572	1415	873	3401	1803	2549	167	66	1009	0
lgt	216	335	105	55	92	69	232	9	234	16	38	21	0
blr	770	105	1728	240	323	235	1348	240	820	152	23	330	0
glr	572	55	240	1346	183	143	1054	204	640	52	12	155	0
bgr	1415	92	323	183	2290	565	1604	464	1409	227	49	395	0
ocl	873	69	235	143	565	1529	1090	183	978	253	33	219	0
rot	3401	232	1348	1054	1604	1090	7344	1380	3292	405	113	1522	0
zom	1803	9	240	204	464	183	1380	2278	611	0	18	535	0
vpc	2549	234	820	640	1409	978	3292	611	4571	370	39	856	0
sml	167	16	152	52	227	253	405	0	370	557	0	69	0
shd	66	38	23	12	49	33	113	18	39	0	125	15	0
rfl	1009	21	330	155	395	219	1522	535	856	69	15	2000	0

Table 3.9: Challenge III. Target image counts for pairs of factors. The top row shows the counts for the 12 factors detailed in Table 3.7. The colors of each column are normalized w.r.t. the top cell in that column.

∩	*clp*	*lgt*	*blr*	*glr*	*bgr*	*ocl*	*rot*	*zom*	*vpc*	*sml*	*shd*	*rfl*	*ok*
all	65.3	48.6	51.6	64.0	65.9	56.4	65.0	70.0	58.6	34.1	70.4	67.5	81.0
clp	65.3	55.1	51.8	67.5	66.8	61.5	67.2	68.1	62.3	45.5	72.7	67.0	n/a
lgt	55.1	48.6	41.0	43.6	59.8	43.5	48.3	44.4	46.1	31.2	57.9	80.9	n/a
blr	51.8	41.0	51.6	48.7	48.6	37.0	52.3	64.2	43.3	21.0	39.1	59.4	n/a
glr	67.5	43.6	48.7	64.0	62.3	47.9	65.1	67.1	60.4	13.5	50.0	64.5	n/a
bgr	66.8	59.8	48.6	62.3	65.9	59.6	66.6	76.1	61.2	29.9	79.6	73.2	n/a
ocl	61.5	43.5	37.0	47.9	59.6	56.4	55.6	75.4	55.9	40.7	78.8	64.8	n/a
rot	67.2	48.3	52.3	65.1	66.6	55.6	65.0	75.5	57.6	32.6	73.4	70.4	n/a
zom	68.1	44.4	64.2	67.1	76.1	75.4	75.5	70.0	66.3	n/a	83.3	69.7	n/a
vpc	62.3	46.1	43.3	60.4	61.2	55.9	57.6	66.3	58.6	33.2	64.1	61.6	n/a
sml	45.5	31.2	21.0	13.5	29.9	40.7	32.6	n/a	33.2	34.1	n/a	46.4	n/a
shd	72.7	57.9	39.1	50.0	79.6	78.8	73.4	83.3	64.1	n/a	70.4	80.0	n/a
rfl	67.0	80.9	59.4	64.5	73.2	64.8	70.4	69.7	61.6	46.4	80.0	67.5	n/a

Table 3.10: Challenge III. Open MIC performance on the combined set w.r.t. the pairs of 12 factors detailed in Table 3.7. Top-1 accuracies for our JBLD approach are listed. The top row shows results w.r.t. the original 12 factors. Color-coded cells are normalized w.r.t. entries of this row. For each column, intense/pale red indicates better/worse results compared to the top cell, respectively.

occlusions. We further combine (*glr*) and (*rfl*) into specularities (*spc*). Table 3.13 shows that the large/heavy variants of truncation, rotation, viewpoint, zoom and occlusions affect performance more than the small/light variants. This highlights the need to further investigate the aspects of invariance to photometric and geometric transformations in domain adaptation algorithms and CNN representations.

Additionally, we revisit Challenge II and present the curves for our proposed top-k-n measure on the combined set. Figure 3.4 shows how the performance of our JBLD approach varies w.r.t. k and n variables detailed in Section 5.2 of our main submission. By increasing n, we can see the gradual increase in accuracy which means that the classifier sometimes confuses the most salient exhibits in images with less salient objects. Nonetheless, even if $n=5$, the results on our new dataset are far from saturation leaving the scope for the future works to improve upon our baselines.

∩	*sml* *glr*	*sml* *blr*	*sml* *bgr*	*sml* *lgt*	*sml* *rot*	*sml* *vpc*	*blr* *ocl*	*blr* *shd*	*sml* *ocl*	*lgt* *blr*	*lgt* *ocl*	*lgt* *glr*
all	52	152	227	16	405	370	235	23	253	105	69	55
clp	7	36	75	3	98	124	133	13	90	57	51	35
lgt	2	10	5	16	8	6	23	13	7	105	69	55
blr	19	152	44	10	122	101	235	23	45	105	23	19
glr	52	19	13	2	38	20	36	6	36	19	16	55
bgr	13	44	227	5	166	175	78	10	100	26	35	19
ocl	20	45	100	7	166	161	235	6	253	23	69	16
rot	38	122	166	8	405	258	171	18	166	72	40	31
zom	0	0	0	0	0	0	20	1	0	2	0	0
vpc	20	101	175	6	258	370	150	12	161	68	52	50
sml	52	152	227	16	405	370	45	0	253	10	7	2
shd	0	0	0	0	0	0	6	23	0	13	12	4
rfl	4	14	28	0	54	42	23	2	22	5	6	4

Table 3.11: Challenge III. Target image counts for the selected triplets of 12 factors detailed in Table 3.7. The top row shows the counts for the pairs of factors we chose to form triplets. The colors of each column are normalized w.r.t. the top cell in that column.

∩	*sml* *glr*	*sml* *blr*	*sml* *bgr*	*sml* *lgt*	*sml* *rot*	*sml* *vpc*	*blr* *ocl*	*blr* *shd*	*sml* *ocl*	*lgt* *blr*	*lgt* *ocl*	*lgt* *glr*
all	13.5	21.0	29.9	31.2	32.6	33.2	37.0	39.1	40.7	40.9	43.5	43.6
clp	42.8	27.8	38.7	66.7	42.8	46.0	44.4	53.8	45.5	49.1	45.1	45.7
lgt	0.0	30.0	40.0	31.2	37.5	50.0	52.3	38.5	10.0	40.9	43.5	43.6
blr	0.0	21.0	18.2	30.0	24.6	17.8	37.0	39.1	11.1	40.9	52.2	21.0
glr	13.5	0.0	7.7	0.0	10.5	15.0	27.8	33.3	27.8	21.0	31.2	43.6
bgr	7.7	18.2	29.9	40.0	27.7	31.4	37.2	60.0	33.0	46.1	51.4	42.1
ocl	15.0	11.1	33.0	14.3	39.7	41.0	37.0	83.3	40.7	52.2	43.5	31.2
rot	10.2	24.6	27.7	37.5	32.6	31.8	38.0	50.0	39.7	43.0	60.0	32.2
zom	n/a	n/a	n/a	n/a	n/a	n/a	75.0	100	n/a	100	n/a	n/a
vpc	15.0	17.8	31.4	50.0	31.8	33.2	35.3	58.3	41.0	35.3	40.4	46.0
sml	13.5	21.0	29.9	31.2	32.6	33.2	11.1	n/a	40.7	30.0	14.3	0.0
shd	n/a	n/a	n/a	n/a	n/a	n/a	83.3	39.1	n/a	38.5	75.0	50.0
rfl	75.0	50.0	39.3	n/a	46.3	45.2	69.6	100	68.2	100	50.0	100

Table 3.12: Challenge III. Open MIC performance on the combined set w.r.t. the selected triplets of 12 factors detailed in Table 3.7. Top-1 accuracies for baselines for our JBLD approach are listed. The top row shows results w.r.t. the most difficult pairs of factors we chose to form triplets. The colors of each column are normalized w.r.t. the top cell in that column.

Lastly, we investigate the use of Mean Average Precision (MAP) in place of the accuracy as MAP can quantify the quality of recognition for datasets with multiple labels per image. For (*Shn*), (*Clk*) and (*Shx*) subsets, we obtain 71.5, 68.1 and 64.8% MAP in contrast to 64.3, 61.2 and 48.5% mean top-1 accuracy, respectively. Such results support our claim that the Open MIC dataset is challenging and the results are far from being saturated; making it a good choice for studying domain adaptation and few-shot learning.

While the protocol for supervised domain utilizes the labeled source and the labeled target training data (a few of datapoints per class), the unsupervised domain adaptation assumes larger unannotated target dataset. Below, we evaluate methods such as the Unsupervised Domain Adaptation with Residual Transfer Networks (*RTN*) [Long et al., 2016], Deep Transfer Learning with Joint Adaptation Networks (*JAN*) [Long et al., 2017] and Deep Hashing Network for Unsupervised Domain Adaptation (*DHN*) [Venkateswara et al., 2017] on the (*Shn*), (*Clk*),

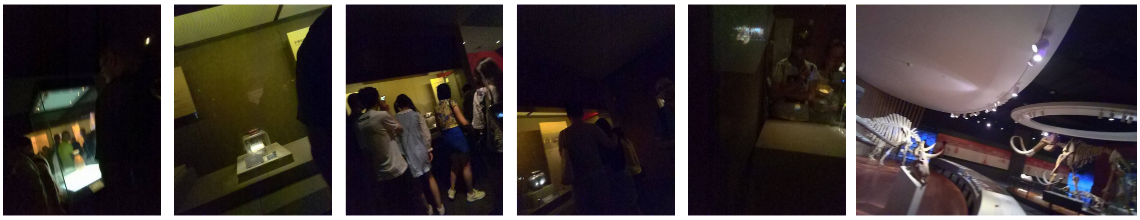


Figure 3.5: Some of the most difficult to identify exhibits from the target domain in the Open MIC dataset.

	acc.	files		acc.	files	*zoo=*	acc.	files
clp=lcl+ hcl+bcl	65.3	5316	*occ=scl+ fcl+ooc*	56.4	1529	*zoo= lzo+hzo*	70.0	2278
lcl	70.6	2827	*scl*	56.0	1086	*lzo*	74.7	1173
hcl	59.0	2344	*fcl*	44.8	268	*hzo*	65.0	1106
bcl	65.4	739	*ooc*	56.9	851			
rot= lro+hro	65.0	7344	*vpc= lvp+hvp*	58.6	4571	*spc= glr+rfl*	66.2	3191
lro	65.4	6724	*lvp*	60.8	3241	*glr*	64.0	1346
hro	60.3	622	*hvp*	53.0	1345	*rfl*	67.5	2000

Table 3.13: Challenge III. Open MIC performance on the combined set w.r.t. additional factors detailed in Table 3.8. Top-1 accuracies for our JBLD approach are listed.

and (*Hon*) subsets of Open MIC. Table 3.14 shows that the unsupervised approaches score lower than JBLD despite we used ResNet-50 for all methods, increased numbers of target datapoints and tweaked all hyper-parameters. However, lower results compared to the supervised domain adaptation are expected as the supervised and unsupervised approaches follow very different training protocols.

A complementary to ours is a dataset for fine-grained domain adaptation [Gebru et al., 2017] which contains 'easily acquired' ∼1M cars of 2657 classes from websites for 'fine-grained' domain adaptation on 170 classes and ∼100 samples per class. In contrast, it took us 6 months and 10 visits to several museums with volunteers to collect a specialist data which cannot be simply found on flicker. We used wearable cameras to capture the target images *e.g.*, skeletons, pottery, tools, jewelery, which all are made of varied materials. Some pieces of art are non-rigid, some emit light, some contain moving parts, some looks extremely similar *etc*. The target data exhibits big scale and viewpoint changes as well as occlusions, motion blur and light glares *etc*.

RTN+Shn	51.0	*JAN*+Shn	49.5	*DHN*+Shn	49.0	*JBLD*+Shn	**64.3**
RTN+Clk	54.7	*JAN*+Clk	51.0	*DHN*+Clk	52.2	*JBLD*+Clk	**61.2**
RTN+Hon	66.0	*JAN*+Rel	65.2	*DHN*+Rel	64.6	*JBLD*+Rel	**77.3**

Table 3.14: Evaluation of the unsupervised domain adaptation on the Open MIC dataset.

For evaluation on the Office-Home dataset [Venkateswara et al., 2017], we chose $Cl \rightarrow$

$Ar/Pr \rightarrow Ar$ domain pairs, 38 source and 12 target train images per class. The baseline (*So*) approach scored 59.5/60.0% accuracy. For JBLD, we obtained **61.6/62.2%**. For 20 source and 3 target training images per class, we obtained 48.1/49.3 (*So*) and **49.2/50.5%** (JBLD) accuracy. Unsupervised approach (*DHN*) scored only 34.69/29.91% in this setting.

3.7 Conclusions

We have collected, annotated and evaluated a new challenging Open MIC dataset with the source and target domains formed by images from Android and wearable cameras, respectively. We covered 10 distinct exhibition spaces in 10 different museums to collect a realistic *in-the-wild* target data in contrast to typical photos for which the users control the shutter. We have provided a number of useful baselines *e.g.*, breakdowns of results per exhibition, combined scores and analysis of factors detrimental to domain adaptation and recognition. Unsupervised domain adaptation and few-shot learning methods can also be compared to our baselines. Moreover, we proposed orthogonal improvements to the supervised domain adaptation *e.g.*, we integrated non-trivial non-Euclidean distances and Nyström projections for better results and tractability. We will make our data and evaluation scripts available to the researchers. One drawback of our approach is that it requires source and target domains to have same categories, classes. This limitation will be addressed in the next chapter, while also focusing on a new recognition task.

Chapter 4

CNN-based Action Recognition and Supervised Domain Adaptation on 3D Body Skeletons via Kernel Feature Maps

4.1 Summary

Deep learning is ubiquitous across many areas of computer vision. It often requires large scale datasets for training before being fine-tuned on small-to-medium scale problems. Activity, or, in other words, action recognition, is one of many application areas of deep learning. While there exist many Convolutional Neural Network architectures that work with the RGB and optical flow frames, training on the time sequences of 3D body skeleton joints is often performed via recurrent networks such as LSTM. In this chapter, we propose a new representation which encodes sequences of 3D body skeleton joints in texture-like representations derived from mathematically rigorous kernel methods. Such a representation becomes the first layer in a standard CNN network *e.g.*, ResNet-50, which is then used in the supervised domain adaptation pipeline to transfer information from the source to target dataset. This lets us leverage the available Kinect-based data beyond training on a single dataset and outperform simple fine-tuning on any two datasets combined in a naive manner. More specifically, in this chapter we utilize the overlapping classes between datasets. We associate datapoints of the same class via so-called commonality, known from the supervised domain adaptation. We demonstrate state-of-the-art results on three publicly available benchmarks.

Following from the previous chapter, we move to more complicated action recognition task. Compared to image recognition and domain adaption from images, action recognition requires understanding of videos which are essentially sequences of images. Also, action recognition task involves learning from 3D body skeleton joints which is inherently a very different representation compared to RGB images, thus making it harder to use them with traditional CNNs. To address these challenges, we propose a novel encoding method to embed these sequences of 3D body skeleton joints into 2D texture like representations. We further improve our alignment loss to work with domains where they contain different number of matching classes. This chapter has been published as a conference paper: "Yusuf Tas, Piotr Koniusz. CNN-based Action Recognition and Supervised Domain Adaptation on 3D Body Skeletons via Kernel Feature Maps. In British Machine Vision Conference (BMVC) 2018".

4.2 Introduction

In recent years, we have witnessed a great increase in the usage and development of deep learning frameworks such as Convolutional Neural Networks (CNN). Starting from an outstanding paper on the AlexNet architecture [Krizhevsky et al., 2012], application areas such as text processing, speech recognition, feature learning and extraction, semantic segmentation, object detection and recognition have adopted deep learning since [Girshick et al., 2014; Collobert and Weston, 2008; Hinton et al., 2012; Ren et al., 2015; Donahue et al., 2014].

Action recognition aims to distinguish between different action classes such as walking, pushing, hand shaking, kicking, punching, to name but a few of action concepts. The ability to recognize human actions enables progress in many application areas verging from the video surveillance to human-computer interaction [Herath et al., 2017a]. Videos have been the main source of the data for action recognition, however, data sources such as RGB-D have become popular since the introduction of the Kinect sensor as they facilitate tracking 3D coordinates of human skeleton body joints which form time sequences. Similar to the object classification, the past action recognition systems relied on handcrafted spatio-temporal feature descriptors such as [Bobick and Davis, 2001; Laptev, 2005; Klaser et al., 2008], with a notable shift to deep learning frameworks [Ji et al., 2012; Karpathy et al., 2014; Simonyan and Zisserman, 2014a; Feichtenhofer et al., 2016] which combine RGB and optical flow CNN streams. However, little has been done to investigate the use of sequences of 3D body skeleton joints in CNNs, with an exception of [Ke et al., 2017].

In this chapter, we focus on the action recognition of sequences of 3D body skeleton joints and propose an input layer which we combine with off-the-shelf CNNs. This enables us to further pursue our goal of the supervised domain adaptation to leverage Kinect-based datasets as the known supervised domain adaptation approaches [Tzeng et al., 2015; Koniusz et al., 2017] are based on CNNs rather than the recurrent networks such as RNN and LSTM [Du et al., 2015; Zhu et al., 2016; Liu et al., 2016a]. Even though recurrent networks are generally the first choice for time series based data, they are still limited by the number of steps they can process [Gu et al., 2017]. On the other hand, action recognition data consists of images frames, videos where they have many steps across time.

It has been shown that in deep networks, early layers recognize edges, corners, basic shapes and structures; prompting similarity to handcrafted features. However, in the consecutive layers, learned filters respond to more complex stimuli [Zeiler and Fergus, 2014]. This attractive property of deep learning together with the shift-invariance of pooling result in a superior performance compared to handcrafted features. Even more powerful are the residual CNN representations [He et al., 2016; Feichtenhofer et al., 2016] which have the ability to bypass the local minima resulting from the non-convex nature of CNN networks. Therefore, our work is based on the ResNet-50 model.

Papers on human action recognition use several datasets such as KTH [Schuldt et al., 2004], HMDB-51 [Kuehne et al., 2011], SBU-Kinect-Interaction [Yun et al., 2012], UTKinect-Action3D [Xia et al., 2012], NTU RGB+D [Shahroudy et al., 2016], most of which have a significant overlap of the class concepts describing actions. Thus, we adopt a domain adaptation approach based on the class-wise mixture of alignments of second-order scatter matrices [Koniusz et al., 2017]. We apply it to time sequences of 3D body skeleton joints to transfer the knowledge between the overlapping classes of two datasets. Our contributions are:

(i) We propose a novel method that encodes sequences of 3D body skeleton joints into a kernel feature map representation suitable for the use with off-the-shelf CNNs. Our representation enjoys a sound mathematical derivation based on kernel methods [Scholkopf and Smola, 2001].

(ii) We are the first to adapt the supervised domain adaptation [Koniusz et al., 2017] for the action recognition on time sequences of 3D body skeleton joints. We extend the so-called mixture alignment of classes [Koniusz et al., 2017] to work with datasets which class concepts match partially.

4.3 Related Work

First, we describe the most popular CNN action recognition models followed by the 3D body joint representations. Subsequently, we focus on the most related to our approach techniques.

CNNs for Action Recornition. Ji et al. [Ji et al., 2012] propose a CNN model to utilize 3D structure in videos by multiple convolution operations. Karpathy et al. [Karpathy et al., 2014] propose a method called 'slow fusion' which learns temporal information by feeding sequentially parts from the video to the algorithm. Simonyan and Zisserman [Simonyan and Zisserman, 2014a] propose a two-stream network which benefits from both spatial domain with RGB images and temporal domain with optical flow.

3D Body Joint Sequences. Systems such as Microsoft Kinect can locate body parts and produce a set of articulated connected body joints that evolve in time and form time sequences of 3D coordinates [Zatsiorsky and Zaciorskij, 2002]. Action recognition via sequences of 3D body skeleton joints has received a wider attention in the community, as witnessed by a survey paper [Presti and La Cascia, 2016].

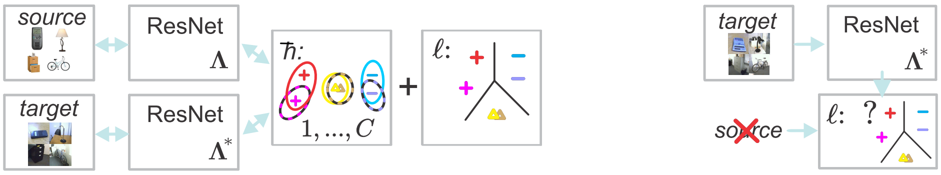
While the RGB-based video sequences contain background, clutter and other sources of noise, the advantage of skeleton-based representations is that they can accurately describe human motion. This was first demonstrated by Johansson [Johansson, 1973] in his seminal experiment involving the moving lights display. By observing moving body joints that represent *e.g.*, elbow, wrist, knee, ankle, one can tell the action taking place. Moreover, sensors such as Kinect fuse depth and RGB frames, and combine the body joint detector, tracker [Shotton et al., 2011], and segmentation to robustly separate the background clutter from the subject's motion. For any given subject/action, the 3D positions of body joints evolve spatio-temporally.

Various descriptors of body joints have been proposed *e.g.*, the motion of 3D points is used in [Hussein et al., 2013; Lv and Nevatia, 2006], orientations w.r.t. a reference axis are used by [Parameswaran and Chellappa, 2006] and relative body-joint positions are used in [Wang et al., 2012; Yang and Tian, 2014]. Connections between body segments are used in [Yacoob and Black, 1999; Ohn-Bar and Trivedi, 2013; Ofli et al., 2014; Vemulapalli et al., 2014]. In contrast, we represent sequences of 3D body-joints by a kernel whose linearization yields texture-like feature maps which capture complex statistics of joints for CNN.

Map generation from 3D Body Joint Sequences. A recent paper [Ke et al., 2017] forms texture arrays from 3D coordinates of body joints. Firstly, 4 key body joints are chosen as reference to form a center of coordinate system by which the 3D positions of remaining body joints are shifted before conversion into cylindircal coordinates. Coordinate of each body joint is stacked along rows while temporal changes happen along columns. This results in 12 maps resized to 224×224 and passed to 12 CNN streams combined at the FC layer.

Our method is somewhat related in that our feature maps resemble textures. However, our maps are obtained by a linearization of the proposed kernel function which measures similarity between any pair of two sequences. The parameters of these kernels introduce a desired degree of shift-invariance in both spatial and temporal domains. Our approach is also somewhat related to kernel descriptors for image recognition [Bo et al., 2011], Convolutional Kernel Networks [Mairal et al., 2014] and kernelized covariances [Cavazza et al., 2016] for action recognition, a time series kernel on scatter matrices [Gaidon et al., 2011] and a spatial compatibility kernel [Koniusz et al., 2016a] that yields a tensor descriptors. In contrast, our layer captures third-order co-occurrences between 3D skeleton body joints and temporal domain to produce texture-like feature maps that are passed to CNN.

Supervised Domain Adaptation. In this chapter, we employ the supervised domain adaptation whose role is to transfer knowledge from the labeled source to labeled target dataset and



(a) **(b)**

Figure 4.1: Supervised Domain Adaptation [Koniusz et al., 2017]. Figure 4.1a: The source and target network streams are combined by the classification and alignment losses ℓ and $\hbar$ (end-to-end learning) which operate on the feature vectors from the final *FC* layers of ResNet-50 streams $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^*$. Loss $\hbar$ aligns covariances for C classes to facilitate transfer learning. Figure 4.1b: At the test time, the target stream only and the classifier are used.

outperform naive fine-tuning on combined datasets. We adapt an approach [Koniusz et al., 2017] based on the mixture of alignments of second-order statistics. One alignment per class per source and target streams is performed to discover the so-called commonality [Koniusz et al., 2017] between the data streams. Thus, both CNN streams learn a transformation of the data into this shared commonality. Figures 4.1a and 4.1b show the training and testing procedures. Training requires a trade-off between alignment and training losses $\hbar$ and ℓ operating on source and target streams $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^*$. Testing uses only the target stream $\boldsymbol{\Lambda}^*$ and the pre-trained classifier.

The approach in [Koniusz et al., 2017] assumes that the source and target data have to share the same set of labels. We relax this assumption to perform transfer between the classes shared between both datasets. Thus, we employ separate source and target classifiers and perform the alignment.

4.4 Preliminaries

In what follows, we explain our notations and the necessary background on shift-invariant RBF kernels and their linearization, which are needed for deriving a kernel on sequences on 3D body skeleton joints together with its linerization into feature maps.

Notations. The Kronecker product is denoted by $\otimes$. $\mathcal{I}_N$ denotes the index set $\{1, 2, ..., N\}$. We use the MATLAB notation $\mathbf{v} = [\text{begin} : \text{step} : \text{end}]$ to generate a vector $\mathbf{v}$ with elements starting as *begin*, ending as *end*, with stepping equal *step*. Operator '$;$' in $[\mathbf{x}; \mathbf{y}]$ concatenates vectors $\mathbf{x}$ and $\mathbf{y}$ (or scalars) while $[\boldsymbol{\Phi}_i]_{i\in\mathcal{I}_J}$ concatenates $\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_J$ along rows.

Kernel Linearization. In the sequel, we use Gaussian kernel feature maps detailed below to embed 3D coordinates and their corresponding temporal time stamp into a non-linear Hilbert space and perform linearization which will result in our texture-like feature maps.

Proposition 10. *Let* $G_\sigma(\mathbf{x}-\mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|_2^2 / 2\sigma^2)$ *denote a Gaussian RBF kernel centered at* $\mathbf{y}$ *and having a bandwidth* σ*. Kernel linearization refers to rewriting this* G_σ *as an*

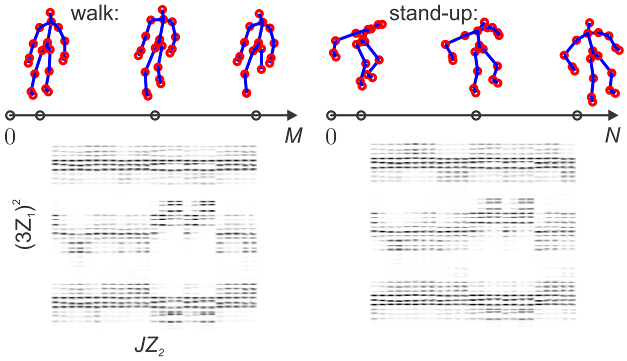


Figure 4.2: Visualization of the feature maps of sequences of 3D body skeleton joints. Note that irrespectively of the sequence length, we always obtain $\boldsymbol{\Phi} \in \mathbb{R}^{225\times 225}$ feature maps.

inner-product of two infinite-dimensional feature maps. To obtain these maps, we use a fast approximation method based on probability product kernels [Jebara et al., 2004]. Specifically, we employ the inner product of d'-dimensional isotropic Gaussians given $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{d'}$. Consider equation:

$$G_\sigma(\mathbf{x}-\mathbf{y}) = \left(\frac{2}{\pi\sigma^2}\right)^{\frac{d'}{2}} \int\limits_{\boldsymbol{\zeta}\in\mathbb{R}^{d'}} G_{\sigma/\sqrt{2}}(\mathbf{x}-\boldsymbol{\zeta})\, G_{\sigma/\sqrt{2}}(\mathbf{y}-\boldsymbol{\zeta})\, \mathrm{d}\boldsymbol{\zeta}. \tag{4.1}$$

Eq. (4.1) *can be approximated by replacing the integral with the sum over Z pivots $\boldsymbol{\zeta}_1, ..., \boldsymbol{\zeta}_Z$:*

$$G_\sigma(\mathbf{x}-\mathbf{y}) \approx \left\langle \sqrt{c}\boldsymbol{\varphi}(\mathbf{x}), \sqrt{c}\boldsymbol{\varphi}(\mathbf{y}) \right\rangle, \text{ where } \boldsymbol{\varphi}(\mathbf{x}) = \left[G_{\sigma/\sqrt{2}}(\mathbf{x}-\boldsymbol{\zeta}_1), ..., G_{\sigma/\sqrt{2}}(\mathbf{x}-\boldsymbol{\zeta}_Z)\right]^T, \tag{4.2}$$

and c represents a constant (it impacts the overall magnitude only so we set $c=1$). We refer to (4.2) *(left) as the linearization of the RBF kernel and* (4.2) *(right) as an RBF feature map*[1].

Proof. Rewrite the Gaussian kernel as the probability product kernel [Jebara et al., 2004] (Sec. 3.1). □

4.5 Proposed Method

Below, we formulate the problem of action recognition from sequences of 3D body skeleton joints, followed by our kernel formulation capturing actions, and its linearization into feature maps which we further feed to off-the-shelf CNN for classification.

[1] Note that (kernel) feature maps are not conv. CNN maps. They are two separate notions that share the name.

4.5.1 Generation of Feature Maps via Kernel Linearization

Let dataset consist of sequences of J 3D body skeleton joints describing human pose skeleton evolving in time. For brevity, we assume each sequence consists of M frames. However, our formulation is applicable to sequences of variable lengths *e.g.*, M and N. Our pose sequence Π is defined as:

$$\Pi = \left\{\mathbf{x}_{is} \in \mathbb{R}^3, i \in \mathcal{I}_J, s \in \mathcal{I}_M\right\}. \tag{4.3}$$

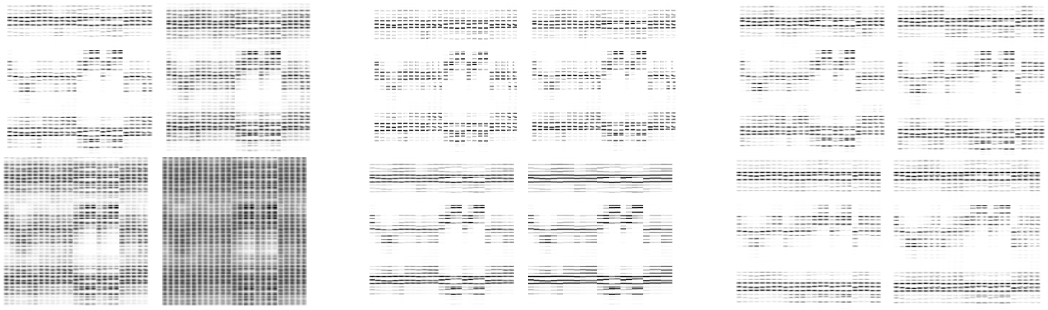
Each sequence Π is described by one of C action labels. We use the sequence Π to generate a feature map which can be considered a descriptor of action associated with Π. Then, such feature maps are generated from given datasets and then fed to the source and target CNN streams with the goal of performing the supervised domain adaptation. Figure 4.2 illustrates the sequences and feature maps obtained as a result of the process detailed next.

In what follows, we want to measure the similarity between any two action sequences in terms of their 3D body skeleton joints as well as their evolution in time. We normalize each skeleton w.r.t. the chest joint (chosen to be the center). Moreover, we normalize such relative coordinates by their total variance computed over the training data. Let Π_A and Π_B be two sequences, each with J joints, and M and N frames, respectively. Further, let $\mathbf{x}_{is} \in \mathbb{R}^3$ and $\mathbf{y}_{jt} \in \mathbb{R}^3$ correspond to coordinates of joints of body skeletons of Π_A and Π_B, respectively. We define our *sequence kernel* (SCK) between Π_A and Π_B as:

$$K(\Pi_A, \Pi_B) = \frac{1}{MN}\sum_{i\in\mathcal{I}_J}\sum_{s\in\mathcal{I}_M}\sum_{t\in\mathcal{I}_N} K_{\sigma_1}(\mathbf{x}_{is} - \mathbf{y}_{it})^2\, G_{\sigma_2}(\frac{s}{M} - \frac{t}{N}), \tag{4.4}$$

where $1/(MN)$ is a normalization constant, and G_{σ_1} and G_{σ_2} are subkernels that capture the similarity between the 3D body skeleton joints and temporal alignment, respectively. Therefore, we have two parameters σ_1 and σ_2 which control the level of tolerated invariance w.r.t. misalignment of 3D body joints and their temporal positions in two sequences, respectively. Moreover, the square of K_{σ_1} in Eq. (4.4) captures co-occurrences of x, y, and z Cartesian coordinates of each 3D body joint–it is shown below that the square operation corresponds to the Kronecker product which is known to capture co-occurrences.

First, we define $K_{\sigma_1}(\mathbf{x} - \mathbf{y}) = \sum_{i\in\mathcal{I}_3} G_{\sigma_1}(x^i - y^i)$ where superscript i chooses x-, y-, or z-axis of a 3D coordinate vector. Next, we linearize the above kernel using the theory from Section 4.4 so that $K_{\sigma_1}(\mathbf{x} - \mathbf{y}) \approx \sum_{i\in\mathcal{I}_3} \phi(x^i)^T\phi(y^i)$, which gives the dot-product of concatenations $K_{\sigma_1}(\mathbf{x} - \mathbf{y}) \approx [\phi(x^1); \phi(x^2); \phi(x^3)]^T[\phi(y^1); \phi(y^2); \phi(y^3)]$. In what follows, we write for simplicity that $K_{\sigma_1}(\mathbf{x} - \mathbf{y}) \approx \phi(\mathbf{x})^T\phi(\mathbf{y})$. Moreover, temporal kernel $G_{\sigma_2}(\frac{s}{M} - \frac{t}{N}) \approx \mathbf{z}(s/M)^T\mathbf{z}(t/N)$. The above linearizations combined with Eq. (4.4) lead



(a) (b) (c)

Figure 4.3: Illustration of the impact of $\sigma_1 = 0.4, 0.6, 0.8, 1.5$ and $\sigma_2 = 0.02, 0.1, 1.0, 5.0$ (in the scanline order) on feature maps are given in Figures 4.3a and 4.3b, respectively. Figure 4.3c shows four different maps for four different sequences. Note the subtle differences.

to:

$$K(\mathit{\Pi}_A, \mathit{\Pi}_B) \approx \frac{1}{MN} \sum_{i\in\mathcal{I}_J} \sum_{s\in\mathcal{I}_M} \sum_{t\in\mathcal{I}_N} \big(\boldsymbol{\phi}(\mathbf{x}_{is})^T \boldsymbol{\phi}(\mathbf{y}_{it})\big)^2 \mathbf{z}(s/M)^T \mathbf{z}(t/N), \tag{4.5}$$

which can be further rewritten into Eq. (4.6) and simplified by Eq. (4.7):

$$K(\mathit{\Pi}_A, \mathit{\Pi}_B) \approx \frac{1}{MN} \sum_{i\in\mathcal{I}_J} \sum_{s\in\mathcal{I}_M} \sum_{t\in\mathcal{I}_N} \Big\langle (\boldsymbol{\phi}(\mathbf{x}_{is}) \otimes \boldsymbol{\phi}(\mathbf{x}_{is}))\mathbf{z}(s/M)^T, (\boldsymbol{\phi}(\mathbf{y}_{it}) \otimes \boldsymbol{\phi}(\mathbf{y}_{it}))\mathbf{z}(t/N)^T \Big\rangle \tag{4.6}$$

$$= \sum_{i\in\mathcal{I}_J} \Big\langle \frac{1}{M} \sum_{s\in\mathcal{I}_M} (\boldsymbol{\phi}(\mathbf{x}_{is}) \otimes \boldsymbol{\phi}(\mathbf{x}_{is}))\mathbf{z}(s/M)^T, \frac{1}{N} \sum_{t\in\mathcal{I}_N} (\boldsymbol{\phi}(\mathbf{y}_{it}) \otimes \boldsymbol{\phi}(\mathbf{y}_{it}))\mathbf{z}(t/N)^T \Big\rangle \;\Rightarrow$$

$$K(\mathit{\Pi}_A, \mathit{\Pi}_B) \approx \langle \boldsymbol{\Phi}(\mathit{\Pi}_A), \boldsymbol{\Phi}(\mathit{\Pi}_B) \rangle, \text{ where} \tag{4.7}$$

$$\boldsymbol{\Phi}(\mathit{\Pi}_A) = \Big[\frac{1}{M} \sum_{s\in\mathcal{I}_M} (\boldsymbol{\phi}(\mathbf{x}_{is}) \otimes \boldsymbol{\phi}(\mathbf{x}_{is}))\mathbf{z}(s/M)^T\Big]_{i\in\mathcal{I}_J}, \boldsymbol{\Phi}(\mathit{\Pi}_B) = \Big[\frac{1}{N} \sum_{t\in\mathcal{I}_N} (\boldsymbol{\phi}(\mathbf{y}_{it}) \otimes \boldsymbol{\phi}(\mathbf{y}_{it}))\mathbf{z}(t/N)^T\Big]_{i\in\mathcal{I}_J},$$

and $\boldsymbol{\Phi}(\mathit{\Pi})$ is our texture-like feat. map for a chosen sequence $\mathit{\Pi}$.

We choose $Z_1 = 5$ pivots $\boldsymbol{\zeta} = [\zeta_1, ..., \zeta_{Z_1}]^T$ for G_{σ_1} which are sampled on interval $[-1; 1]$ with equal steps *e.g.*, $\boldsymbol{\zeta} = [-1 : 2/(Z_1 - 1) : 1]$. This results in a $3Z_1$ dimensional map that approximates K_{σ_1}. For G_{σ_2}, we choose such an integer number of pivots Z_2 that $Z_2 J = 225$. We sample these pivots on interval $[0; 1]$. This way, we obtain $\boldsymbol{\Phi} \in \mathbb{R}^{Z_1^2 \times Z_2 J}$ which can be readily fed to an off-the-shelf CNN stream. Figure 4.3 demonstrates the impact of σ_1 and σ_2 radii on the feature maps $\boldsymbol{\Phi}$. Our feature map is similar in spirit to Convolutional Kernel Networks [Mairal et al., 2014] for image classification which demonstrated that the linearization of a carefully designed kernel adheres to standard CNN operations such as convolution, non-linearity and pooling. This motivates our belief that our feature maps are more suited/compatible for interfacing with CNNs than ad-hoc texture-like representations [Ke et al., 2017].

4.5.2 Alignment of Second-order Statistics

For the full details of the *So-HoT* algorithm, please refer to Chapter 2. Below, we review the core part of our algorithm for the reader's convenience. Suppose $\mathcal{I}_N$ and $\mathcal{I}_{N^*}$ are the indexes of N source and N^* target training data points. $\mathcal{I}_{N_c}$ and $\mathcal{I}_{N_c^*}$ are the class-specific indexes for $c \in \mathcal{I}_C$, where C is the number of classes. Furthermore, suppose we have feature vectors from an *FC* layer of the source network stream, one per an action sequence or image, and their associated labels. Such pairs are given by $\boldsymbol{\Lambda} \equiv \{(\boldsymbol{\phi}_n, y_n)\}_{n \in \mathcal{I}_N}$, where $\boldsymbol{\phi}_n \in \mathbb{R}^d$ and $y_n \in \mathcal{I}_C$, $\forall n \in \mathcal{I}_N$. For the target data, by analogy, we define pairs $\boldsymbol{\Lambda}^* \equiv \{(\boldsymbol{\phi}^*_n, y^*_n)\}_{n \in \mathcal{I}^*_N}$, where $\boldsymbol{\phi}^* \in \mathbb{R}^d$ and $y^*_n \in \mathcal{I}_C$, $\forall n \in \mathcal{I}^*_N$. Class-specific sets of feature vectors are given as $\boldsymbol{\Phi}_c \equiv \{\boldsymbol{\phi}^c_n\}_{n \in \mathcal{I}_{N_c}}$ and $\boldsymbol{\Phi}^*_c \equiv \{\boldsymbol{\phi}^{*c}_n\}_{n \in \mathcal{I}_{N^*_c}}$, $\forall c \in \mathcal{I}_C$. Then $\boldsymbol{\Phi} \equiv (\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_C)$ and $\boldsymbol{\Phi}^* \equiv (\boldsymbol{\Phi}^*_1, ..., \boldsymbol{\Phi}^*_C)$. The asterisk in superscript (*e.g.* $\boldsymbol{\phi}^*$) denotes variables related to the target network while the source-related variables have no asterisk. Figure 4.4 shows the setup we use. The *So-HoT* problem is posed as a trade-off between the classifier and alignment losses ℓ and $\hbar$:

$$\mathop{\arg\min}_{\substack{\mathbf{W},\mathbf{W}^*,\boldsymbol{\Theta},\boldsymbol{\Theta}^* \\ \text{s. t. } ||\boldsymbol{\phi}_n||_2^2 \leq \tau, \\ ||\boldsymbol{\phi}^*_{n'}||_2^2 \leq \tau, \\ \forall n \in \mathcal{I}_N, n' \in \mathcal{I}^*_N}} \ell(\mathbf{W}, \boldsymbol{\Lambda}) + \ell(\mathbf{W}^*, \boldsymbol{\Lambda}^*) + \eta ||\mathbf{W} - \mathbf{W}^*||_F^2 + \underbrace{\frac{\alpha_1}{C} \sum_{c \in \mathcal{I}_C} ||\boldsymbol{\Sigma}_c - \boldsymbol{\Sigma}^*_c||_F^2 + \frac{\alpha_2}{C} \sum_{c \in \mathcal{I}_C} ||\boldsymbol{\mu}_c - \boldsymbol{\mu}^*_c||_2^2}_{\hbar(\boldsymbol{\Phi}, \boldsymbol{\Phi}^*)}. \tag{4.8}$$

For ℓ, a generic Softmax loss is employed. For the source and target streams, the matrices $\mathbf{W}, \mathbf{W}^* \in \mathbb{R}^{d \times C}$ contain unnormalized probabilities. In Equation (4.8), separating the class-specific distributions is addressed by ℓ while attracting the within-class scatters of both network streams is handled by $\hbar$. Variable η controls the proximity between $\mathbf{W}$ and $\mathbf{W}^*$ which encourages the similarity between decision boundaries of classifiers.

The loss $\hbar$ depends on two sets of variables $(\boldsymbol{\Phi}_1, ..., \boldsymbol{\Phi}_C)$ and $(\boldsymbol{\Phi}^*_1, ..., \boldsymbol{\Phi}^*_C)$ – one set per network stream. Feature vectors $\boldsymbol{\Phi}(\boldsymbol{\Theta})$ and $\boldsymbol{\Phi}^*(\boldsymbol{\Theta}^*)$ depend on the parameters of the source and target network streams $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}^*$ that we optimize over. $\boldsymbol{\Sigma}_c \equiv \boldsymbol{\Sigma}(\boldsymbol{\Phi}_c)$, $\boldsymbol{\Sigma}^*_c \equiv \boldsymbol{\Sigma}(\boldsymbol{\Phi}^*_c)$, $\boldsymbol{\mu}_c(\boldsymbol{\Phi})$ and $\boldsymbol{\mu}^*_c(\boldsymbol{\Phi}^*)$ denote the covariances and means, respectively, one covariance/mean pair per network stream per class. Coefficients α_1, α_2 control the degree of the scatter and mean

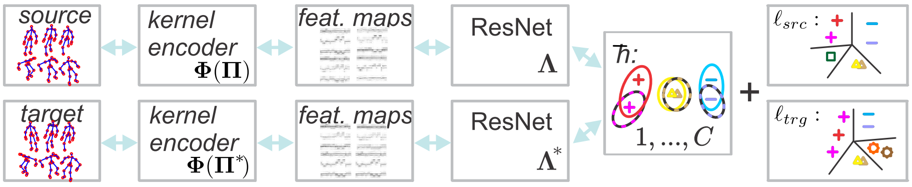


Figure 4.4: Our pipeline: combining the 3D body skeleton encoding and the supervised domain adaptation. Unlike [Koniusz et al., 2017], we utilize two classifiers (one per network stream) and perform alignment between the classes that are shared between the source and target datasets.

alignment, τ controls the ℓ_2-norm of feature vectors.

Algorithm 1 details how we perform domain adaptation. We enable the alignment loss $\hbar$ only if the source and target batches correspond to the same class. Otherwise, the alignment loss is disabled and the total loss uses only the classification log-losses ℓ_{src} and ℓ_{trg}. To generate the source and target batches that match w.r.t. the class label, we re-order source and target datasets class-by-class and thus each source/target batch contains only one class label at a time. Once all source and target datapoints with matching class labels are processed, remaining datapoints are processed next.

Our final pipeline is illustrated in Figure 4.4. As our ultimate goal is to transfer knowledge between Kinect-based datasets, we combine the described in Section 4.5.1 encoder of sequences of 3D body skeleton joints together with the supervised domain adaptation algorithm *So-HoT* [Koniusz et al., 2017] with details and modifications as discussed above. *So-HoT* yields state-of-the-art results on the Office dataset [Saenko et al., 2010], however, it works with datasets which are described by the same class concepts. Thus, we adapt their algorithm to our particular needs *e.g.*, we only perform the alignment of second-order statistics between the classes that are shared between the source and target datasets. Moreover, we employ separate classifier losses ℓ_{src} and ℓ_{trg} for the source and target stream, respectively. The separate target classifier allows the target network to work with class labels absent from the source dataset. At the test time, we cut off the source stream (and the source classifier), as illustrated in Figure 4.1b.

4.6 Experiments

Below, we detail our network setting, datasets and we show experiments on our feature maps for sequences of 3D body skeleton joints in the context of the supervised domain adaptation.

Network Model. We use the two streams network architecture from [Koniusz et al., 2017]. For each CNN stream, we chose the Residual CNN model ResNet-50 [He et al., 2016] pre-trained on ImageNet dataset [Krizhevsky et al., 2012] for both source and target streams. The *Pool-5* layers of the source and target streams are forwarded to a fully connected layer *FC* with 512 hidden units and this is forwarded to both the classification weight layer and the so-called alignment loss [Koniusz et al., 2017]. Two classifiers are used for the source and target streams. Moreover, the alignment loss is activated when the generated source and target mini-batches contain datapoints with the same class labels. See Algorithm 1 for more details and Figure 4.4 for the network setting.

The training is performed by the Stochastic Gradient Descent (SGD) with the momentum set to 0.9. Mini-batch sizes differ depending on both the source and target dataset.

Algorithm 1 Batch generation + a single epoch of the training procedure on the source and target datasets.

```
$src\_data$ := sort_by_class_label($src\_data$)
$target\_data$ := sort_by_class_label($target\_data$)
$C_s$                                                  ▷ Number of the source classes
$C_t$                                                  ▷ Number of the target classes
$C_{s \cap t}$                                         ▷ Number of classes in common
procedure EPOCH($src\_data, target\_data, batch\_size$)   ▷ Training (one epoch)
    for $i \leftarrow 1 : max(C_s, C_t)$ do
        if $i \leq C_s$ then
            $batch_s \leftarrow$ Choose($src\_data, i, batch\_size$)   ▷ 'Choose' pre-fetches data of class i
        else
            $batch_s \leftarrow$ Choose($src\_data, rnd(), batch\_size$)   ▷ 'Choose' pre-fetches data of random class
        if $i \leq C_t$ then
            $batch_t \leftarrow$ Choose($target\_data, i, batch\_size$)
        else
            $batch_t \leftarrow$ Choose($target\_data, rnd(), batch\_size$)
        if $i \leq C_{s \cap t}$ then
            $Loss \leftarrow \ell_{src} + \ell_{trg} + \hbar$
        else
            $Loss \leftarrow \ell_{src} + \ell_{trg}$
        Forward($net\_data$, $batch\_s$, $batch\_t$)
        Backward($net\_data$, $batch\_s$, $batch\_t$)
        Update($net\_data$, $batch\_s$, $batch\_t$)
```

Datasets.

We use the NTU RGB-D, SBUKinect Interaction and UTKinect-Action3D datasets.

NTU RGB-D [Shahroudy et al., 2016], the largest action recognition dataset to date, contains ~56000 sequences of 60 distinct action classes and sequences of actions/interactions performed by 40 different subjects. 3D coordinates of 25 body joints are provided. We use the cross-subject evaluation protocol [Shahroudy et al., 2016] and used only the train split as our source data. For pre-processing, we translated 3D body joints by the joint-2 (middle of the spine) and we chose the body with the largest 3D motion as the main actor for the multi-actor sequences.

SBUKinect [Yun et al., 2012] contains videos of 8 interaction categories between two people, and 282 skeleton sequences with 15 3D body joints. Although the locations of body joints are noisy [Yun et al., 2012] and pre-processing is common [Zhu et al., 2016], we do not perform any pre-processing or data augmentation in contrast to [Ke et al., 2017]. In domain adaptation setting, we use the NTU training set as the source and SBU as the target data. For evaluation,

Methods	SBU	UTK
Cylindrical textures, 1×CNN [Ke et al., 2017]	89.37%	95.0%
Cylindrical textures, 3×CNN [Ke et al., 2017]	90.24%	95.9%
Kernel feature maps, 1×CNN (ours)	**91.13%**	**96.5%**

Table 4.1: Comparisons of texture representations.

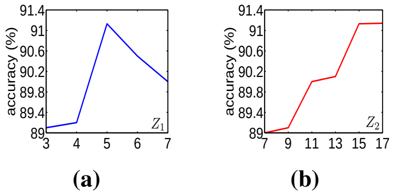


Figure 4.5: Accuracy w.r.t. Z_1 ($Z_2 = 15$) and Z_2 ($Z_1 = 5$) on SBU in Figures 4.2a and 4.2b.

we follow [Yun et al., 2012] and use 5-fold cross-validation on the given splits. As each sequence contains 2 persons, we used each skeleton as a separate training datapoint. For testing, we averaged predictions over such pairs.

UTKinect-Action3D [Xia et al., 2012] contains 10 action captured by Kinect, 199 sequences, and 20 3D body skeleton joints. We avoid data augmentation or pre-processing. Protocol [Zhu et al., 2013] has 2 splits: half of the subjects for training and half for testing. NTU training set is our source.

Experiments. Below, we focus on the following types of experiments, each utilizing our encoder which transforms sequences of 3D body skeleton joints into feature maps:

(i) Target-only: only target dataset is used for training and testing (no domain adaptation).

(ii) Source+target: the source and target datasets (both training and validation splits) are combined into one larger dataset. Testing is performed on the target testing set only. No domain adaptation is used but the network is trained on both domains.

(iii) Second-order alignment: our extended So-HoT model applies the domain adaptation between the source and target training datapoints. We perform the alignment of second-order statistics whenever the source and target class names match.

No Domain Adaptation. Firstly, we compare our encoding to texture-based representation [Ke et al., 2017]. Approach [Ke et al., 2017] forms 4 arrays of cylindrical coordinates of 3D skeleton body joints, each translated w.r.t. each 4 pre-defined key-joints. Such arrays are later resized, cropped *etc.* and fed to network via multiple CNN inputs. They require a dedicated

Methods	Accuracy
Raw Skeleton [Yun et al., 2012]	49.7%
Hierarchical RNN [Du et al., 2015]	80.35%
Deep LSTM [Zhu et al., 2016]	86.03%
Deep LSTM + Co-occurrence [Zhu et al., 2016]	90.41%
ST-LSTM [Liu et al., 2016a]	88.6%
ST-LSTM + Trust Gate [Liu et al., 2016a]	93.3%
Frames + CNN [Ke et al., 2017]	90.8%
Clips + CNN + MTLN [Ke et al., 2017]	93.57%
SBU only (*target*)	91.13%
NTU+SBU combined (*source+target*)	91.52%
Second-order alignment	**94.36%**

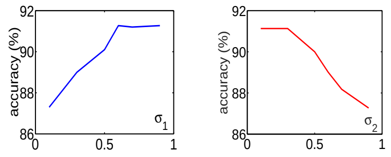
Table 4.3: Results on the SBUKinect dataset.

Methods	Accuracy
3D Histogram (leave one out) [Xia et al., 2012]	90.92%
Lie Group [Vemulapalli et al., 2014]	97.08%
SCK + DCK [Koniusz et al., 2016a]	98.39%
Skeleton Joint Features [Zhu et al., 2013]	90.9%
ST-LSTM + Trust Gate [Liu et al., 2016a]	95.0%
Elastic Functional Coding [Anirudh et al., 2015]	94.9%
UTK only (*target*)	96.5%
NTU+UTK combined (*source+target*)	97.5%
Second-order alignment	**98.9%**

Table 4.4: Results on the UTKinect dataset.

Methods	Cross subject
Hierarchical RNN [Du et al., 2015]	59.1%
Deep RNN [Shahroudy et al., 2016]	59.3%
Deep LSTM [Shahroudy et al., 2016]	60.7%
ST-LSTM + Trust Gate [Liu et al., 2016a]	69.2%
Frames + CNN [Ke et al., 2017]	**75.73%**
NTU only (*target*)	74.52%
NTU+UTK+SBU combined (*source+target*)	74.65%
Second-order alignment (UTK→NTU)	74.91%
Second-order alignment (SBU→NTU)	74.83%
Second-order alignment (UTK+SBU→NTU)	**75.35%**

Table 4.5: Results on the NTU dataset.



(a) **(b)**

Figure 4.6: Sensitivity w.r.t. param. σ_1 and σ_2 on SBU. Figures 4.6a and 4.6b show the accuracy w.r.t. σ_1 ($\sigma_2 = 0.3$) and σ_2 ($\sigma_1 = 0.6$), resp.

CNN pipeline which combines all these arrays. To make a fairer comparison to our encoding and use an off-the-shelf CNN setting, we simplified representation [Ke et al., 2017] to use only a single body key-joint center for translation. We use the same setting for our encoding and [Ke et al., 2017] based on ResNet-50. We do not use a domain adaptation for results in Table 4.1. We include however a variant of method [Ke et al., 2017] which generates 3 texture images (one per each cylindrical coordinate). Thus, these 3 texture images are passed via 3 CNN streams and their *FC* vectors are concatenated.

Table 4.1 shows the comparison of our texture-like feature map encoding against method [Ke et al., 2017]. With 3× more texture images taking 3× more time to process via 3 CNN streams, method (*Cylindrical textures,* 3×*CNN*) [Ke et al., 2017] performs ∼0.7–0.9% worse than ours. Moreover, for fairness, we next combine their 3 texture images (one per each cylindrical coordinate) into an RGB-like texture and passed via 1 CNN stream (*Cylindrical textures,* 1×*CNN*). Table 4.1 shows that given the same ResNet-50 pipeline, our method outperforms theirs by ∼1.8% and 1.4% on SBU and UTK. Figures 4.5 and 4.6 show that our encoder is not too sensitive w.r.t. the choice of Z_1, Z_2, σ_1 and σ_2 on the SBU dataset (no domain adaptation). Figure 4.7 shows a similar analysis on the UTK dataset.

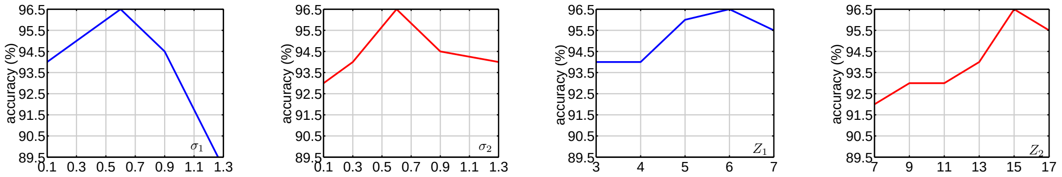
Although idea [Ke et al., 2017] appears somewhat related to ours, the inner workings of both methods differ *e.g.*, our method is mathematically inspired to attain desired shift-invariance w.r.t. 3D positions of coordinates and the temporal domain. In contrast, approach [Ke et al., 2017] is hand-crafted.

Domain Adaptation Setting. Having shown that our encoder outperforms [Ke et al., 2017] given the same pipeline, we discuss below results on the supervised domain adaptation pipeline.

In Table 4.3, we compare our method against state-of-the-art results on the SBU dataset. After enabling the domain adaptation algorithm (*second-order alignment*), the accuracy increases by **3.23%** over training on the target data only (*target*). Our method also outperforms naive training on the combined source and target data (*source+target*) by **1.84%**. We note that without any data augmentation, our method outperforms more complicated approaches which utilize numerous texture-like representations per sequence combined with several CNN streams and a fusion network (*Clips+CNN+MTLN*) [Ke et al., 2017]. This shows the effectiveness of our supervised domain adaptation on sequences of 3D body skeleton joints.

Table 4.4 shows on the UTK dataset that domain adaptation (*second-order alignment*) outperforms the baseline (*target*) and the naive fusion (*source+target*) by **2.4%** and **1.4%**.

Table 4.5 presents the transfer results from UTK and/or SBU to NTU. Transferring the knowledge from small- to large-scale datasets is a difficult task. However, by combining UTK and SBU to form a source dataset, we were able to still gain 0.8% improvement over the baseline (*target*). We obtain results similar to [Ke et al., 2017] with a much simpler pipeline.



(a) **(b)** **(c)** **(d)**

Figure 4.7: Sensitivity w.r.t. parameters σ_1 and σ_2 on UTK. Figures 4.7a, 4.7b, 4.7c and 4.7d show the accuracy w.r.t. σ_1 ($\sigma_2 = 0.6$), σ_2 ($\sigma_1 = 0.6$), Z_1 ($Z_2 = 15$) and Z_2 ($Z_1 = 5$), respectively.

4.7 Conclusions

In this chapter, we have demonstrated that sequences of 3D body skeleton joints can be easily encoded with the use of appropriately designed kernel function. A linearization of such a kernel function produces texture-like feature maps which constitute a first feed-forward layer further interconnected with off-the-shelf CNNs. Moreover, we have also demonstrated that the supervised domain adaptation can be performed on such representations and that small-scale Kinect-based datasets can benefit from the knowledge transfer from the large-scale NTU dataset. We believe our contributions lead to state-of-the-art results. They also open up interesting avenues on how to use the time sequences with traditional off-the-shelf CNNs and how to leverage the abundance of the skeleton-based action recognition datasets. In the next chapter, we work on an even more difficult adaptation task named multimodal conversations which requires understanding and adapting information from two very separate domains: text and images.

Chapter 5

Simple Dialogue System with AUDITED

5.1 Summary

We devise a multimodal conversation system for dialogue utterances composed of text, image, or both modalities. We leverage Auxiliary UnsuperviseD vIsual and TExtual Data (AUDITED). To improve the performance of the text-based task, we utilize translations of target sentences from English to French to form the assisted supervision. For the image-based task, we employ the DeepFashion dataset in which we seek nearest neighbor images of positive and negative target images of the multimodal dialogue dataset. These nearest neighbors form the nearest neighbor embedding providing an external context for target images. We develop two methods to create neighbor embedding vectors: Neighbor Embedding by Hard Assignment (NEHA) and Neighbor Embedding by Soft Assignment (NESA), which generate context subspaces per target image. Subsequently, these subspaces are learned by our pipeline as a context for the target data. We also propose a discriminator which switches between the image- and text-based tasks. We show improvements over baselines on the large-scale Multimodal Dialogue Dataset (MMD) and SIMMC.

In this chapter, we work on the multimodal conversation problem. Multimodal conversations include dialogues that are constructed from both sentences and images. The Multimodal Dialogue dataset [Saha et al., 2018] provides an extensive collection of multimodal conversations between a shopper and retail agent. It proposes two benchmarking tasks, image and text. We propose novel methods for each task by adapting external knowledge through our assisted supervision methods. This chapter has been accepted as a conference paper: "Yusuf Tas, Piotr Koniusz. Simple Dialogue System with AUDITED. In British Machine Vision Conference (BMVC) 2021".

5.2 Introduction

Deep learning is popular in many areas *e.g.*, object detection [Girshick et al., 2014], speech recognition [Graves et al., 2013], image super-resolution [Dong et al., 2015], text and natural language processing [Devlin et al., 2018], domain adaptation [Koniusz et al., 2017, 2018; Tas and Koniusz, 2018], few-shot learning [Zhang and Koniusz, 2019; Zhang et al., 2020b, 2021; Koniusz and Zhang, 2020], and even arts recognition [Zhang et al., 2017; Koniusz et al., 2018]. Realistic problems such as Visual Question Answering (VQA) are often multimodal. Image Captioning (IC) [Xu et al., 2015] learns from text and images to generate image captions. VQA [Zeng et al., 2017] answers questions about a video by leveraging the spatio-temporal visual data and the accompanying text. Multimodal conversation systems use text and images used together as chat bots [Ram et al., 2018], autonomous retail agents [Saha et al., 2018] and task-specific dialogue systems [Wen et al., 2016]. Saha *et al.*[Saha et al., 2018] introduced one of the largest multimodal conversation datasets called Multimodal Dialogue (MMD) dataset, containing over 150K shopper-retail agent dialogues. Figure 5.1a shows dialogues of shoppers asking about/referring to items or asking for items from a given image. MMD contains the image- and text-based tasks. In the image-based task, the model has to retrieve/rank the correct image from given positive and negative images in response to the multimodal context. The text-based task predicts the agent's response within the context.

In this chapter, we go beyond separate protocols of Saha *et al.*[Saha et al., 2018] by introducing a discriminator whose role is to learn/predict an appropriate task.As limited number of utterances contain images, we leverage external visual and textual knowledge via the so-called assisted supervision. Figure 5.1a shows our pipeline. Our contributions are listed below:

(i) We propose a novel assisted supervision to create a context for target images and thus implicitly incorporate more images in unsupervised manner into the learning process of image-based task. The DeepFashion dataset [Liu et al., 2016b] is used to search for closest matching images to given positive and negative target images. Through the perspective of sampling the natural manifold of images, we capture context images for target images.

(ii) We design two embeddings for neighbor images: Neighbor Embedding by Hard Assignment (NEHA) and Neighbor Embedding by Soft Assignment (NESA). NEHA retrieves η nearest neighbors for positive/negative target images to encode them into subspace descriptors by SVD. NESA also reweights the contribution of each context image by the membership probability in a GMM-like model [Koniusz et al., 2013, 2016b] spanned on target

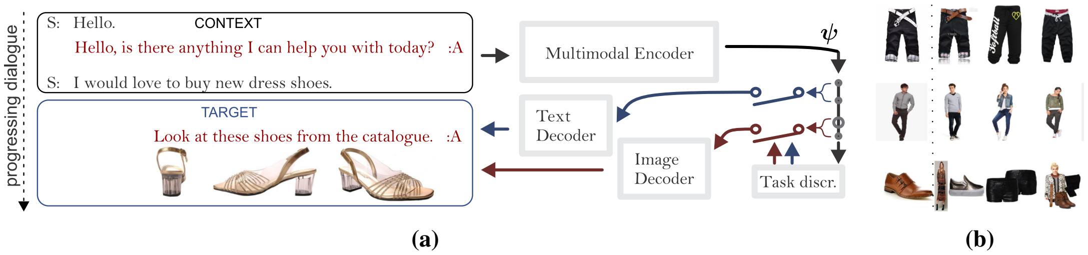

progressing dialogue
S: Hello.
CONTEXT
Hello, is there anything I can help you with today? :A
S: I would love to buy new dress shoes.
TARGET
Look at these shoes from the catalogue. :A
Multimodal Encoder
ψ
Text Decoder
Image Decoder
Task discr.
(a)
(b)

Figure 5.1: Our pipeline includes the Multimodal Encoder, Text Decoder, Image Decoder (Feature Matching Head) and the Task Discriminator (Fig. 5.1a). The MMD dataset contains dialogues between shoppers (*S*) and retail agents (*A*) which progress in time. Dialogues are split by the sliding window (default protocol) to form the input (*CONTEXT*) fed to the Multimodal Encoder. The output (*TARGET*) may contain text, images, or both modalities, which are imposed via dedicated losses on the Text and/or Image Decoders. The switches indicate that one half of the Context Descriptor $\boldsymbol{\psi}$ may be passed to the Text Decoder and the other half to the Image Decoder depending on the Task Discriminator. The details of Multimodal Encoder, Text Decoder and Image Decoder are shown in Figures 5.2a, 5.2b and 5.3, respectively. Figure 5.1b shows 3 nearest neighbors (columns 2–4) retrieved (decreasing similarity order) from DeepFashion [Liu et al., 2016b] for query samples (column 1) from the MMD dataset [Saha et al., 2018]. Feature descriptors were encoded by ResNet-50, the approximate nearest neighbor search was performed by the FAISS library [Johnson et al., 2019]. Such images form an external context for target images.

images.

(iii) For the text-based task, we propose an assisted supervision that uses translation decoders to generate predictions of text in multiple languages to learn a universal representation of conversations by limiting ambiguities of a single language model [Marian and Shook, 2012].

(iv) Finally, we introduce a discriminator whose role is to combine image- and text-based tasks by learning to predict an appropriate task in response to the given multimodal context.

The above strategy of leveraging unsupervised data can be seen as capturing the variance of linguistic and visual data to help the network capture how each utterance may vary.

5.3 Related Work

Below we describe popular dialogue systems and detail the Multimodal Hierarchical Encoder Decoder (M-HRED) [Saha et al., 2018] on which we build.

Conversation Systems. Early conversation systems [Banchs, 2012; Ameixa et al., 2014] use scripts and subtitles for retrieval of responses in a dialogue. Ritter *et al.*[Ritter et al., 2010] uses generative probabilistic models for conversations on blogging websites. VQA approaches [Antol et al., 2015; Wu et al., 2018] answer questions about images. Approaches [Das et al., 2017; Kottur et al., 2019] tackle visual dialogs about individual images. Approach [Thomason et al., 2020] focuses on the visual dialog navigation. Bhattacharya *et al.*[Bhattacharya et al.,

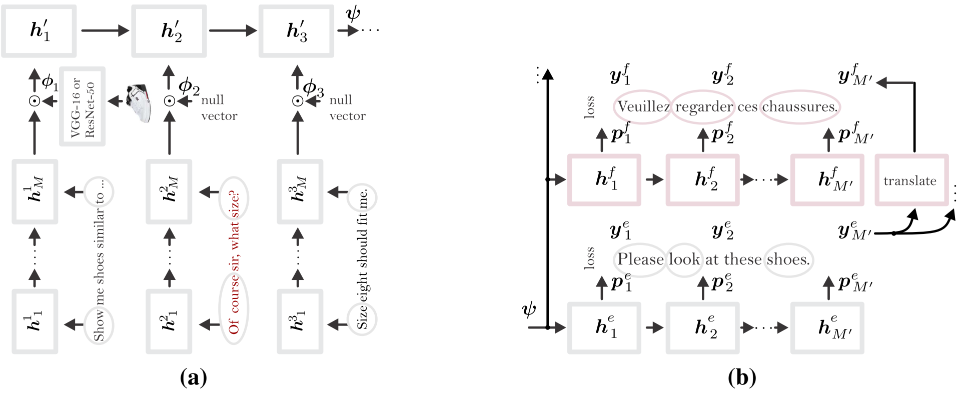


Figure 5.2: In Multimodal Encoder, shown in Fig. 5.2a, the text is processed by a first-level GRU while images are encoded by ResNet-50 to obtain compact embeddings. We concatenate text and avg-pooled image representations (if image is not present, we use a null vector) by ⊙ into utterance descriptors $\phi_1, ..., \phi_3$ and process them with a second-level GRU to produce the Context Descriptor ψ, which we pass it to the Text Decoder with the assisted supervision in Figure 5.2b. The text is translated from English (ground truth) into French (and other languages). The losses (per language) encourage the network to absorb syntactic differences which implicitly helps capture the true dynamics of the dialogue better. Standard Text Encoder [Saha et al., 2018] consists of the gray blocks while pink blocks form our assisted supervision.

2019] retrieves images via textual queries. FashionIQ [Guo et al., 2019] is concerned with the NLP-based image retrieval.

Recent dialogue systems use an RNN encoder-decoder [Sordoni et al., 2015; Shang et al., 2015]. Hierarchical Recurrent Encoder-Decoder [Serban et al., 2015] uses a two-level RNN to create a context-aware conversation system. Approach [Saha et al., 2018] predicts answers of a shopping assistant from natural conversations of the large scale MMD dataset, which we use. Below, we describe and build on models [Serban et al., 2016, 2017].

Multimodal-Hierarchical Encoder Decoder. M-HRED [Saha et al., 2018] is an extension of Hierarchical Recurrent Encoder Decoder (HRED) models [Serban et al., 2016, 2017]. HRED consists of two different levels of Recurrent Neural Network (RNN) [Mikolov et al., 2010] combined together, which represent an encoder which captures the so-called word and sentence context, respectively. The first RNN in HRED model learns to generate the next word in a given sentence by using the word context. The second RNN takes the final representation of a given sentence to generate the representation of next sentence by using the sentence context. An RNN decoder receives a sentence-level representation to decode it and generate a full sentence. Moreover, M-HRED and HRED use the interconnected encoder and decoder but M-HRED also uses images.

Multimodal Encoder (ME). ME receives a sequence of N utterances (so-called context) to produce the Context Descriptor via GRU [Chung et al., 2014]. An utterance contains a sentence, image or both modalities. Images are encoded by VGG-16 [Simonyan and Zisserman,

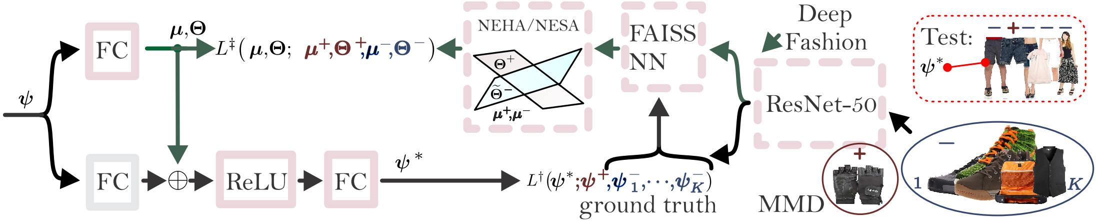


Figure 5.3: Our Image Decoder a.k.a. Feature Matching Head consists of the main stream (FC→ReLU→FC) whose role is to take Context Descriptors $\boldsymbol{\psi}$ and produce visual features $\boldsymbol{\psi}^{*}$ that are combined with loss $L^{\dagger}$ in Eq. (5.1). The traditional head (older method) contains one FC layer (gray block). For each ground truth target positive and negative image descriptors $\boldsymbol{\psi}^{+}$ and $\boldsymbol{\psi}^{-}_{1},...,\boldsymbol{\psi}^{-}_{K}$ from the MMD dataset (encoded by ResNet-50), we find η and ηK approximate nearest neighbor image descriptor from DeepFashion [Liu et al., 2016b] with the FAISS library [Johnson et al., 2019]. Then we create positive and negative mean descriptors $\boldsymbol{\mu}^{+}$ and $\boldsymbol{\mu}^{-}$ as well as subspaces $\boldsymbol{\Theta}^{+}$ and $\boldsymbol{\Theta}^{-}$ with NEHA or NESA step. They capture the mean and variability of positive and negative images. The role of another FC layer is to learn positive visual context representations $(\boldsymbol{\mu},\boldsymbol{\Theta})$ via the assisted supervision loss $L^{\ddagger}$ in Eq. (5.2) which attracts $(\boldsymbol{\mu},\boldsymbol{\Theta})$ towards $(\boldsymbol{\mu}^{+},\boldsymbol{\Theta}^{+})$ and repels it from $(\boldsymbol{\mu}^{-},\boldsymbol{\Theta}^{-})$. Finally, $(\boldsymbol{\mu},\boldsymbol{\Theta})$ are combined with the main stream via a residual link (operator ⊕). Blocks with dashed borders/losses are not used during testing. During testing, $\boldsymbol{\psi}^{*}$ is matched against test images of an utterance. In the above example, the correct ground truth is ranked as second (R@1 fails but R@2 succeeds).

2014b] (4096 ch. of the last FC layer [Saha et al., 2018]).

Multimodal Utterance Encoder (MUE) in Fig. 5.2a consists of two levels of GRU [Chung et al., 2014]. The first-level GRU (bottom) contains hidden states $\boldsymbol{h}^{n}_{1},...,\boldsymbol{h}^{n}_{M}$, where M is the maximum number of input words per utterance, each word is one-hot encoded with a discrete vocabulary of size $V=7457$, $n=1,...,N$ and N is the context size *e.g.*, $N=3$ utterances. The first-level GRU and ResNet-50 encode words and images, respectively. The last state and the output of ResNet-50 are concatenated by ⊙ into $\boldsymbol{\phi}$ and padded with zeros if image or text is missing. Encoded utterances are passed to the Context Encoder (CE), a second-level GRU, with hidden states $\boldsymbol{h}'_{1},...,\boldsymbol{h}'_{N'}$ shown in Fig. 5.2a (top) to obtain a Context Descriptor $\boldsymbol{\psi}$ per context. Fig. 5.1a shows examples of context and target utterances. We use encoder networks from the M-HRED model [Saha et al., 2018] (results in the same testbed, based on ResNet-50).

Multimodal Decoder (MD). MD receives the Context Descriptor $\boldsymbol{\psi}$ from CE. For the text-based task, the target is a sentence. A GRU decoder [Serban et al., 2016] with hidden states $\boldsymbol{h}^{e}_{1},...,\boldsymbol{h}^{e}_{M'}$ generates the target sentence word-by-word, starting with the start-of-sentence and ending with end-of-sentence token. Given the target ground truth sentence with one-hot representation of words $\mathbf{y}^{e}$ and the final output predictions $\mathbf{p}^{e}$ from the model (e indicates English), the combined Multimodal Encoder Decoder is trained via the cross-entropy loss. At the test time, the quality of generated utterance is evaluated against target ground truth sentences via so-called BLEU and NIST metrics [Saha et al., 2018]. Figure 5.2b shows our extended Text Decoder (pink plus gray blocks) and the baseline Text Decoder (gray block) [Saha et al., 2018]. The image-based task is the ranking-based task. Given a positive target image, and K negative images, Context Descriptor $\boldsymbol{\psi}$ is ranked against these positive and negative images at the test time. During training, M-HRED uses the cosine similarity and the hinge loss:

$$L^{\dagger}\left(\boldsymbol{\psi}^{*};\boldsymbol{\psi}^{+},\boldsymbol{\psi}_{1}^{-},...,\boldsymbol{\psi}_{K}^{-}\right)=\max\left(0,1-\boldsymbol{\psi}^{*T}\left(\boldsymbol{\psi}^{+}-\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{\psi}_{k}^{-}\right)\right), \tag{5.1}$$

where $\boldsymbol{\psi}^{*}\in\mathbb{R}^{4096}$ is a feature vector obtained by passing the Context Descriptor $\boldsymbol{\psi}\in\mathbb{R}^{1024}$ from CE via an FC layer, and $\boldsymbol{\psi}^{+}\in\mathbb{R}^{4096}$ and $\boldsymbol{\psi}^{-}\in\mathbb{R}^{4096}$ correspond to image descriptors (VGG-16) for the positive and negative ground truth images, resp. The Hinge loss encourages $\boldsymbol{\psi}^{*}$ to be close to $\boldsymbol{\psi}^{+}$ and away from $\boldsymbol{\psi}^{-}$. L is minimized w.r.t. network parameters.

Self-supervised Learning. Pretext tasks such as sampling and predicting patch locations (left, right, top left, top right), rotations (0°, 90°, 180°, 270°) or other transformations are popular in self-supervision [Doersch et al., 2015; Dosovitskiy et al., 2015; Gidaris et al., 2018; Zhang et al., 2020a, 2021]. Note self-supervision by mutual information estimation [Hjelm et al., 2018], egomotion prediction [Agrawal et al., 2015], and multi-task self-supervised learning [Doersch and Zisserman, 2017].

```
Input: $\eta'\leq\eta$, $K$, $L$
$\boldsymbol{\psi}_1^{+}$, $\boldsymbol{\Psi}^{-}\equiv\{\boldsymbol{\psi}_1^{-},...,\boldsymbol{\psi}_K^{-}\}$ ← ground truth positive and negative target descriptors from MMD,
$\{\boldsymbol{\psi}'_1,...,\boldsymbol{\psi}'_L\}$ ← unsupervised feature descriptors from DeepFashion [Liu et al., 2016b].
$(\boldsymbol{\psi}'^{+}_1,...,\boldsymbol{\psi}'^{+}_{\eta})=\text{FAISS\_NN}(\boldsymbol{\psi}^{+},\eta;\ \{\boldsymbol{\psi}'_1,...,\boldsymbol{\psi}'_L\})$
for $n=1,...,\eta$:
    $\boldsymbol{\psi}'^{+}_n\leftarrow s^{+}(\boldsymbol{\psi}'^{+}_n,\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})\cdot\boldsymbol{\psi}'^{+}_n$
$\boldsymbol{\mu}^{+}=\frac{1}{\eta}\sum_{n=1}^{\eta}\boldsymbol{\psi}'^{+}_n$
$(\boldsymbol{\Theta}^{+},\boldsymbol{\lambda}^{+})=\text{SVD}(\boldsymbol{\psi}'^{+}_1-\boldsymbol{\mu}^{+},...,\boldsymbol{\psi}'^{+}_{\eta}-\boldsymbol{\mu}^{+};\ \eta')$
for $k=1,...,K$:
    $(\boldsymbol{\psi}'^{-}_{1k},...,\boldsymbol{\psi}'^{-}_{\eta k})=$
        $\text{FAISS\_NN}(\boldsymbol{\psi}_k^{-},\eta;\ \{\boldsymbol{\psi}'_1,...,\boldsymbol{\psi}'_L\})$
for $k=1,...,K$:
    for $n=1,...,\eta$:
        $\boldsymbol{\psi}'^{-}_{nk}\leftarrow s^{+}(\boldsymbol{\psi}'^{-}_{nk},\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})\cdot\boldsymbol{\psi}'^{-}_{nk}$
$\boldsymbol{\mu}^{-}=\frac{1}{\eta K}\sum_{n=1}^{\eta}\sum_{k=1}^{K}\boldsymbol{\psi}'^{-}_{nk}$
$(\boldsymbol{\Theta}^{-},\boldsymbol{\lambda}^{-})=\text{SVD}(\boldsymbol{\psi}'^{-}_1-\boldsymbol{\mu}^{-},...,\boldsymbol{\psi}'^{-}_{\eta K}-\boldsymbol{\mu}^{-};\ \eta')$
Output: $(\boldsymbol{\mu}^{+},\boldsymbol{\Theta}^{+})$ and $(\boldsymbol{\mu}^{-},\boldsymbol{\Theta}^{-})$
```

Algorithm 2: Neighbor Embedding by Hard Assignment (black color). Neighbor Embedding by Soft Assignment (black/blue colors).

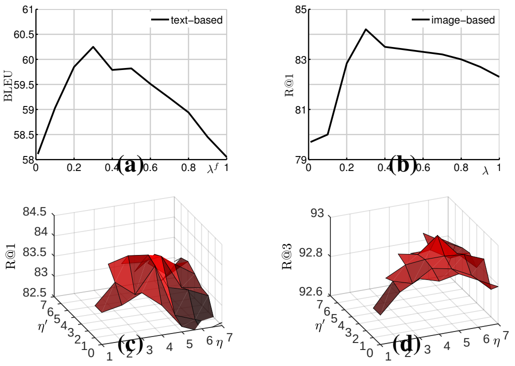


Fig. 5.4a & 5.4b are cross-validation results w.r.t. λ^f and λ for the text- and image-based tasks. Fig. 5.4c & 5.4d are cross-val. results (R@1 & R@3 metric) w.r.t. η' and η for the image-based task. If $\eta'=0$, we use μ^- and μ^+ only.

One may use Bag-of-Words on hand-crafted descriptors for an alignment task [Wang et al., 2019; Wang and Koniusz, 2021], or form positive and negative sampling for a contrastive learning strategy [Zhu and Koniusz, 2021b,a; Zhu et al., 2021]. GAN-based pipelines [Goodfellow et al., 2014; Shiri et al., 2019b,a] also perform self-supervision by generator-discriminator competition.

Motivation from Cognitive Psychology. For the text-based task, we use a translating network

[Ott et al., 2018] and decoders to predict target responses in several languages. This limits the quantization noise resulting from the single language syntactic thus helping capture universal concepts better. Cognitive psychology notes that multilingual babies exhibit better attention and conflict management, and adjust to new rules quicker than monolingual babies [Marian and Shook, 2012].

For the image-related task, we retrieve the η and ηK nearest neighbors from the DeepFashion [Liu et al., 2016b] dataset for positive and negative target images to form subspace descriptors which represent the learning context of target images, and form the manifold of fashion images. From the psychological point of view, our approach is motivated by knowledge transfer, which is '*the dependency of human conduct, learning or performance on prior experience*' a.k.a. '*transfer of particle*' [Woodworth and Thorndike, 1901]. Notice that pre-training our visual task on the DeepFashion is impossible as DeepFashion dataset is not organised in the form of dialogue.

In conclusion, providing multiple translations and multiple positive images (subspaces are second-order statistics) helps our pipeline capture better the innate variance of data.

5.4 Our Approach

Notations. Bold lowercase symbols are vectors *e.g.*, $\boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{\psi}$. Regular lowercase/uppercase symbols are scalars *e.g.*, η, K, N. Bold uppercase symbols are matrices or sets of parameters *e.g.*, $\boldsymbol{\Theta}$. Symbols $\odot$ and $\oplus$ are the vector concatenation & summation (residual link).

Pipeline. Our pipeline in Figure 5.1a follows the baseline model [Saha et al., 2018] in that we use the Multimodal Encoder, Text Decoder and Image Decoder (Feature Matching Head only). The Multimodal Encoder receives the context, a collection of $N = 3$ utterances which are snippets of dialogues between a shopper and a retail agent obtained by a sliding window, a standard protocol on the MMD dataset of retail dialogues. The context window may contain text, image, or both modalities. The Multimodal Encoder takes N utterances, based on a discrete vocabulary of size V, and ResNet-50 encoded images to produce the Context Descriptor which is fed to the Text Decoder and Image Decoder, whose roles are to predict a target ground truth text responses (within discrete vocabulary space) and/or generate ResNet-50 image features to retrieve a visual recommendation from the MMD (or SIMMC) dataset (also encoded with ResNet-50). As the baseline model [Saha et al., 2018] is formulated as two separate tasks, it requires ground truth test labels about the type of output task to perform. In contrast, we introduce the Task Discriminator (the pink box in Figure 5.1a which resolves this issue. To improve predictions, our Text and Image Decoders use the assisted supervision by leveraging the knowledge from the DeepFashion [Liu et al., 2016b] dataset and the translation model [Ott et al., 2018] in an unsupervised way. Section 5.3 details the Multimodal Encoder. Below we detail our decoders.

Image-based Task. Figure 5.3 shows our Feature Matching Head (Image Decoder). The image-based task finds the closest match between a predicted image descriptor and one positive and K negative ground truth descriptor candidates per target utterance. The image-based task uses two losses, the standard loss $L^{\dagger}$ given by Eq. (5.1) and our assisted supervision loss:

$$L^{\ddagger}\left(\boldsymbol{\mu},\boldsymbol{\Theta};\ \boldsymbol{\mu}^{+},\boldsymbol{\Theta}^{+},\boldsymbol{\mu}^{-},\boldsymbol{\Theta}^{-}\right)=\max\Big(0,1-\boldsymbol{\mu}^{T}(\boldsymbol{\mu}^{+}-\boldsymbol{\mu}^{-})-\sum_{n=1}^{\eta'}\mathbf{u}_{n}^{T}(\mathbf{u}_{n}^{+}-\mathbf{u}_{n}^{-})\Big), \tag{5.2}$$

where $\boldsymbol{\psi}^{*}\in\mathbb{R}^{D}$ is a feature vector of size $D=2048$ obtained by passing the Context Descriptor $\boldsymbol{\psi}\in\mathbb{R}^{1024}$ from CE via an FC layer. Moreover, $\boldsymbol{\mu}\in\mathbb{R}^{D}$ and $\boldsymbol{\Theta}\equiv[\mathbf{u}_{1},...,\mathbf{u}_{\eta'}]\in\mathbb{R}^{D\times\eta'}$ are the context feature vectors generated by an FC layer, indicated in Figure 5.3, which are encouraged by a Hinge loss to approach the mean $\boldsymbol{\mu}^{+}\in\mathbb{R}^{D}$ and eigenvectors $\boldsymbol{\Theta}^{+}\equiv[\mathbf{u}_{1}^{+},...,\mathbf{u}_{\eta'}^{+}]\in\mathbb{R}^{D\times\eta'}$ and stay repelled from the mean $\boldsymbol{\mu}^{-}\in\mathbb{R}^{D}$ and eigenvectors $\boldsymbol{\Theta}^{-}\equiv[\mathbf{u}_{1}^{-},...,\mathbf{u}_{\eta'}^{-}]\in\mathbb{R}^{D\times\eta'}$. Visual Feature Descriptors (VFD) $(\boldsymbol{\mu}^{+},\boldsymbol{\Theta}^{+})$ and $(\boldsymbol{\mu}^{-},\boldsymbol{\Theta}^{-})$ represent the positive and negative context for the ground truth positive and negative target descriptors $\boldsymbol{\psi}^{+}\in\mathbb{R}^{D}$ and $\boldsymbol{\psi}_{1}^{-},...,\boldsymbol{\psi}_{K}^{-}\in\mathbb{R}^{D}$ obtained from ResNet-50. Below we explain Neighbor Embedding by Hard Assignment (NEHA) and Neighbor Embedding by Soft Assignment (NESA) which produce VFDs.

NEHA is obtained by applying SVD to η and ηK nearest neighbors $\boldsymbol{\psi'}_{1}^{+},...,\boldsymbol{\psi'}_{\eta}^{+}\in\mathbb{R}^{D}$ and $\boldsymbol{\psi'}_{11}^{-},...,\boldsymbol{\psi'}_{\eta K}^{-}\in\mathbb{R}^{D}$ found among images of DeepFashion [Liu et al., 2016b] dataset encoded by ResNet-50, represented by L feature descriptors $\boldsymbol{\psi}'_{1},...,\boldsymbol{\psi}'_{L}$. The search is performed by FAISS [Johnson et al., 2019], an extremely efficient approximate nearest neighbor search library, by searching feature descriptors of DeepFashion against the ground truth positive/negative target descriptors $\boldsymbol{\psi}^{+}$ and $\boldsymbol{\psi}_{1}^{-},...,\boldsymbol{\psi}_{K}^{-}$ from the MMD dataset, respectively. Figure 5.1b shows the quality of matching images from DeepFashion against ground truth images from MMD.

Algorithm 2 shows steps of NEHA. FAISS_NN$(\boldsymbol{\psi},\eta;\ \{\boldsymbol{\psi'}_{1},...,\boldsymbol{\psi'}_{L}\})$ denotes the FAISS search which retrieves η approximate nearest neighbors of $\boldsymbol{\psi}$ from $\{\boldsymbol{\psi'}_{1},...,\boldsymbol{\psi'}_{L}\}$. Moreover, SVD$(\boldsymbol{\psi}_{1},...,\boldsymbol{\psi}_{\eta};\ \eta')$ returns $\eta'\leq\eta$ leading eigenvectors and eigenvalues $(\boldsymbol{\Theta},\boldsymbol{\lambda})$. We note that NEHA does not take into account the effect of decreasing similarity between the ground truth positive/negative target descriptors and searched feature descriptors of DeepFashion as one progresses over consecutive $1,...,\eta$ nearest neighbors. Thus, Visual Feature Descriptors $(\boldsymbol{\mu}^{+},\boldsymbol{\Theta}^{+})$ and $(\boldsymbol{\mu}^{-},\boldsymbol{\Theta}^{-})$ may provide gradually worsening visual context for target descriptors of MMD. To his end, we introduce an improved strategy below.

NESA follows NEHA but it uses reweighting by so-called Soft Assignment applied prior to SVD steps. We use the two weighting functions for positive $\boldsymbol{\psi'}^{+}$ and negative $\boldsymbol{\psi'}^{-}$:

$$s^{+}(\boldsymbol{\psi}',\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})=\tfrac{1}{\tau(\boldsymbol{\psi}',\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})}\,\mathrm{e}^{-\frac{\|\boldsymbol{\psi}'-\boldsymbol{\psi}^{+}\|_2^2}{2\sigma^2}}\ \text{and}\ s^{-}(\boldsymbol{\psi}',\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})=\tfrac{1}{\tau(\boldsymbol{\psi}',\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})}\max_{k=1,...,K}\mathrm{e}^{-\frac{\|\boldsymbol{\psi}'-\boldsymbol{\psi}_{k}^{-}\|_2^2}{2\sigma^2}}, \tag{5.3}$$

where $\boldsymbol{\Psi}^{-}\equiv\{\boldsymbol{\psi}_k^-\}_{k=1}^{K}$. Expression $\tau(\boldsymbol{\psi}',\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})$ given below normalizes probability partitions in Eq. (5.3):

$$\tau(\boldsymbol{\psi}',\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-})=\mathrm{e}^{-\frac{\|\boldsymbol{\psi}'-\boldsymbol{\psi}^{+}\|_2^2}{2\sigma^2}}+\sum_{k=1}^{K}\mathrm{e}^{-\frac{\|\boldsymbol{\psi}'-\boldsymbol{\psi}_k^{-}\|_2^2}{2\sigma^2}}, \tag{5.4}$$

while σ determines the steepness of likelihood partitions. The Soft Assignment step is performed by reweighting $\boldsymbol{\psi}'^{+}_{1},...,\boldsymbol{\psi}'^{+}_{\eta}$ by $s^{+}(\boldsymbol{\psi}'^{+}_{1},\cdot,\cdot),...,s^{+}(\boldsymbol{\psi}'^{+}_{\eta},\cdot,\cdot)$ and $\boldsymbol{\psi}'^{-}_{11},...,\boldsymbol{\psi}'^{-}_{\eta K}$ by $s^{-}(\boldsymbol{\psi}'^{-}_{11},\cdot,\cdot),...,s^{-}(\boldsymbol{\psi}'^{-}_{\eta K},\cdot,\cdot)$. Algorithm 2 (with steps highlighted in blue) realizes NESA. NEHA and NESA use combination losses: $L^{\dagger}\left(\boldsymbol{\psi}^{*};\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-}\right)+\lambda L^{\ddagger}\left(\boldsymbol{\mu},\boldsymbol{\Theta};\ \boldsymbol{\mu}^{+},\boldsymbol{\Theta}^{+},\boldsymbol{\mu}^{-},\boldsymbol{\Theta}^{-}\right)$.

NNO. Nearest Neighbor Only (NNO) strategy is given for completeness. NNO simply encourages the standard head with one FC layer (gray block in Figure 5.3) to get closer not only to target samples of MMD but also to the positive approximate nearest neighbor(s) retrieved from DeepFashion. NNO uses combined losses: $L^{\dagger}\left(\boldsymbol{\psi}^{*};\boldsymbol{\psi}^{+},\boldsymbol{\Psi}^{-}\right)+\lambda\|\boldsymbol{\psi}^{*}-\frac{1}{\eta}\sum_{n=1}^{\eta}\boldsymbol{\psi}'^{+}_{n}\|_2^2$.

Text-based Task. Figure 5.2b shows that apart from the standard GRU decoder (gray blocks), we use translating network [Ott et al., 2018] to translate [Ott et al., 2018] ground truth sentences from English into French, German and Russian, with one GRU per language. For English, we have a GRU with hidden states $\boldsymbol{h}^{e}_{1},...,\boldsymbol{h}^{e}_{M'}$, output predictions $\mathbf{p}^{e}_{1},...,\mathbf{p}^{e}_{M'}$ and ground truth one-hot vectors $\mathbf{y}^{e}_{1},...,\mathbf{y}^{e}_{M'}$. By analogy, we use analogous streams for other languages. Moreover, every $\mathbf{p}^{e}_{m}\in\mathbb{R}^{7457}$ is an output of an FC layer connected to the corresponding hidden state $\boldsymbol{h}^{e}_{m}\in\mathbb{R}^{1024}$. The FC layer translates hidden states into word activation vectors corresponding to a 7457 dimensional dictionary. Note that for every language, the dictionary size differs. For French, we have 9519 words after considering words with the occurrence of at least 5× given the training data. Each sentence starts with the start-of-sentence token, ends with the end-of-sentence token and is padded to the maximum sentence length of $M'=20$ with the pad-sentence token. Pink blocks realize the assisted supervision for the text-based task. At the test time, they are removed. The final loss for the Text Decoder becomes:

$$L\left(\left\{(\mathbf{p}^{e}_{m},\mathbf{y}^{e}_{m}),\ (\mathbf{p}^{f}_{m},\mathbf{y}^{f}_{m}),...\right\}_{m=1}^{M'};\ \lambda^{f},...\right)=\sum_{m=1}^{M'}{\mathbf{y}^{e}_{m}}^{T}\log\left(\mathbf{p}^{e}_{m}\right)+\lambda^{f}{\mathbf{y}^{f}_{m}}^{T}\log\left(\mathbf{p}^{f}_{m}\right)+...,\tag{5.5}$$

where λ^{f} is the relevance constant of the French translation task, and λ^{g} and λ^{r} are relevance constants for German/Russian but we omit them from notations for brevity.

Task Discriminator (TD). The Context Descriptor[1] $\boldsymbol{\psi}\in\mathbb{R}^{2048}$ is passed to an FC layer (2048×3 size) following the cross-entropy loss with task labels: *text-based*, *image-based* and *text+image-based*. During training, we can access such labels. Thus, during testing, we can go beyond separate protocols of the baseline model [Saha et al., 2018]. Figure 5.1a shows TD and the switches that pass relevant halves of $\boldsymbol{\psi}$ to subsequent modules.

[1] For evaluations where we use TD, the Context Descriptor $\boldsymbol{\psi}$ is in fact 2048 dimensional as its both halves are dedicated to text- and image-based tasks, respectively. For individual tasks, $\boldsymbol{\psi}$ are 1024 dimensional.

			BLEU	NIST
MMD *v1*	T-HRED	[Saha et al., 2018]	14.58	2.61
	M-HRED	[Saha et al., 2018]	20.42	3.09
	M-HRED+attention	[Saha et al., 2018]	19.58	2.46
	M-HRED+attention+KB	[Agarwal et al., 2018]	-	-
	(Ours) Pre-training (French)		24.35	4.12
	(Ours) Assisted sup. (French)		**26.21**	**4.45**
MMD *v2*	T-HRED	[Saha et al., 2018]	35.9	5.14
	M-HRED	[Saha et al., 2018]	56.67	7.51
	M-HRED+attention	[Saha et al., 2018]	50.20	6.64
	M-HRED+attention+KB	[Agarwal et al., 2018]	46.36	-
	(Ours) Augmentation (random deletion)		56.83	7.55
	(Ours) Augmentation (sentence compr. [Hou et al., 2020])		57.65	7.62
	(Ours) Augmentation (back translation [Ott et al., 2018])		59.06	7.96
	(Ours) Pre-training (on SIMMC dataset [Moon et al., 2020])		58.91	7.95
	(Ours) Training on MMD+SIMMC)		59.03	7.98
	(Ours) Pre-training (French)		58.78	7.91
	(Ours) Assisted sup. (French)		**60.12**	**8.11**
	(Ours) Assisted sup. (French+German)		**60.51**	**8.17**
	(Ours) Assisted sup. (French+German+Russian)		**60.75**	**8.22**
Tran.	(Ours) Pre-training (French)		60.88	9.28
	(Ours) Assisted sup. (French)		**64.47**	**11.18**
	(Ours) Assisted sup. (French+German)		**65.54**	**12.41**
	(Ours) Assisted sup. (French+German+Russian)		**66.19**	**12.89**

Table 5.1: Text-based task (MMD *v1* & *v2*). T-HRED / M-HRED are text-only HRED / Multimodal HRED. *Tran.*: transformer backb. [Vaswani et al., 2017].

5.5 Experiments

Datasets. Our experiments are conducted on the MMD datasets [Saha et al., 2018] *v1* and *v2* containing ∼150000 dialogues and the SIMMC dataset [Moon et al., 2020], with ∼13K human-human dialogues and ∼169K utterances. The assisted supervision for the text-based task is achieved via model [Ott et al., 2018] trained on the WMT [Bojar et al., 2014] and Paracrawl [Bañón et al., 2020] datasets containing ∼150M sentence pairs. The assisted supervision for the image-based task is achieved by retrieving relevant feature descriptors from the DeepFashion dataset [Liu et al., 2016b] (∼0.8M images).

			R@1	R@2	R@3
MMD *v1*	T-HRED	[Saha et al., 2018]	46.0	64.0	75.0
	M-HRED	[Saha et al., 2018]	72.0	86.0	92.0
	M-HRED+attention	[Saha et al., 2018]	79.0	88.0	93.0
	(Ours) NNO $\eta = 1$		82.6	88.8	93.2
	(Ours) NNO $\eta = 2$*		83.0	88.9	93.2
	(Ours) NEHA $\eta = 4$*		84.5	89.7	93.6
	(Ours) NESA $\eta = 4$*		**85.3**	**90.3**	**94.0**
MMD *v2*	T-HRED	[Saha et al., 2018]	44.0	60 .0	72.0
	M-HRED	[Saha et al., 2018]	69.0	85.0	90.0
	M-HRED+attention	[Saha et al., 2018]	78.0	87.0	92.3
	(Ours) NNO $\eta = 1$		82.5	88.6	92.8
	(Ours) NNO $\eta = 2$*		83.1	88.8	92.9
	(Ours) NEHA $\eta = 4$*		84.5	89.5	93.2
	(Ours) NESA $\eta = 4$*		**85.2**	**90.1**	**93.7**

Table 5.2: Image-based task (MMD *v1* & *v2*) for one positive and $K = 5$ negative target images. T-HRED is HRED with context images ignored in training. M-HRED is the Multimodal HRED. See Recall at top-1, 2 and 3, '*' is the optimal η.

MMD dataset [Saha et al., 2018] contains 105439 train, 22595 validation and 22595 test dialogues, each with $\sim$40 shopper-retailer utterances containing a sentence, images or both modalities. We used train, validation and test splits to train, select hyperparameters and report final results, respectively. MMD *v2* does not contain additional image descriptions from the agent.

SIMMC dataset [Moon et al., 2020] has $\sim$13K human-human dialogs and $\sim$169K utterances, it uses a multimodal Wizard-of-Oz (WoZ) setup, on two shopping domains, furniture (grounded in a shared virtual environment) and fashion (grounded in an evolving set of images).

Settings. Following Saha *et al.*[Saha et al., 2018], we perform the text- and image-based tasks for which we use the same hidden unit size, text encoding size and the learning rate as M-HRED [Saha et al., 2018]. For our combined task (TD module), the hidden unit size is doubled (Section 5.4). For the text- and image-based tasks, we report BLEU/NIST [Saha et al., 2018] and Recall at top-l cut-off (R@l).

Results. Below we start with cross-validation of key hyperparameters followed by presenting our main results for text-, image- and mixed (text+images) tasks.

Cross-validation of λ^f and λ. For joint training of French auxiliary decoder with the base English decoder, we cross-validated $\lambda^f \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$ on the validation set (see Figure 5.4a). We fixed $\lambda^f = 0.3$ throughout experiments as this value yielded the highest score of **60.25%** (BLEU) on the MMD *v2* validation split. If we use two auxiliary decoders *e.g.*, French and German, we set $\lambda^f = \lambda^g = 0.15$. For three auxiliary decoders, we set $\lambda^f = \lambda^g = \lambda^r = 0.1$. For joint training of the main stream (FC→ReLU→FC) and the assisted supervision stream in Feature Matching Head from Figure 5.3, we set $\lambda = 0.5$ following cross-validation on the validation set given NEHA, shown Figure 5.4b.

Cross-validation of NESA. Figures 5.5, 5.6 5.7 evaluate the performance of NESA model with respect to its parameters given the validation split. In Figure 5.5, the bandwidth of the RBF kernel σ value is fixed to 0.5. We vary the number of nearest neighbors $\eta \in \{2, 3, 4, 5, 6, 7\}$ and the number of leading eigenvectors $\eta' \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. In Figure 5.6, number of eigenvectors is fixed to 2, that is $\eta' = 2$. We cross-validate the number of nearest neighbors $\eta \in \{2, 3, 4, 5, 6, 7\}$ and the RBF bandwidth set to $\sigma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. In Figure 5.7, number of nearest neighbors η is fixed to 4. We cross-validate the number of leading eigenvectors $\eta' \in \{0, 1, 2, 3, 4\}$ and the RBF bandwidth $\sigma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

As previously indicated, the best $\eta' \approx \eta$. The best $\sigma = 0.5$ which we use across all our experiments. Moreover, as $\eta' \approx \eta = 4$, this indicates that our NESA can create a rich visual context for target images (much better context than directly forcing target images to be close to their nearest neighbors in DeepFashion). Moreover, NESA outperforms NEHA.

	BLEU		R@1
M-HRED	52.17	M-HRED+att.	75.05
Assisted sup. (French)	55.29	NEHA	81.50
(French+German+Russian)	**56.11**	NESA	**82.43**

Table 5.3: Mixed task (MMD *v2*) with the Task Discr., assisted supervision (text and images).

	User 1	User 2	User 3	mean
clarity	61.6	58.4	64.2	61.4
compactness	52.0	52.8	54.6	53.1
helpfulness	62.0	60.2	63.0	61.7

Table 5.4: User study on the mixed task (MMD dataset *v2*). Our approach *vs*. M-HRED.

Component (or method)	runtime (h)
T-HRED / M-HRED	15 / 15
Pre-training (Fr) (+fine-tuning En)	16 + 6
Augmentation (back translation) + transl.	15 + 40
Assisted sup. (Fr / Fr+Ge / Fr+Ge+Ru)	20 / 29 / 38
Translator [Ott et al., 2018] (En → Fr / Ge / Ru)	20 / 20 / 20
Evaluating BLEU & NIST	0.5

Table 5.5: Runtim

	BLEU	R@1	R@5	R@10
HRE (SIMMC)	0.079	16.3	33.1	41.7
Ours F+R+G	**0.102**	n/a	n/a	n/a
Ours+Trans. F+R+G	**0.187**	n/a	n/a	n/a
Ours NEHA	n/a	17.3	33.7	42.2
Ours NESA	n/a	**20.1**	**35.5**	**43.1**

Table 5.6: SIMMC-Fashion (Task 2). Response Generation. *F+R+G* are French, Russian and German auxiliary tasks. *Tran.* is the transformer backbone [Vaswani et al., 2017].

Component (or method)	runtime (h)
T-HRED / M-HRED	15 / 15
NNO	16
NEHA / NESA	18 / 19
ResNet-50 features (MMD+DeepFashion)	6
FAISS search [Johnson et al., 2019] (+SVD)	1.5 (+2)
Evaluating R@1	0.1
T-HRED / M-HRED (text+image)	30 / 30
Mixed task (text+image+task discr.)	40

Table 5.7: Runtimes: image-based task (various comp.) and the mixed task (MMD *v2*).

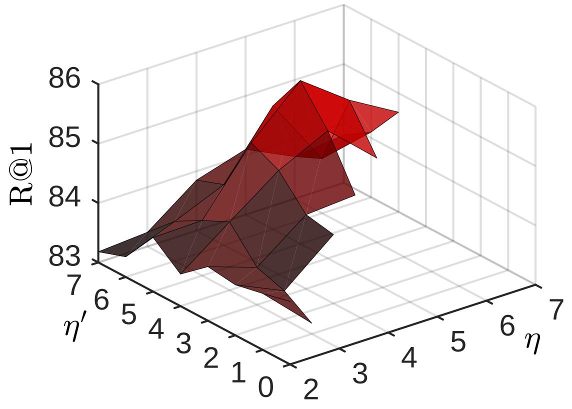


Figure 5.5: Performance (R@1) w.r.t. the number of nearest neighbors η and the number of leading eigenvectors η' on NESA ($\sigma = 0.5$).

Image-based Task. Firstly, we evaluate the baseline M-HRED+attention with ResNet-50 in place of VGG-16, and we note that the results are within ±0.3% of results given the original M-HRED+attention with VGG-16. Table 5.2 shows that using the assisted supervision via the NNO strategy with one nearest neighbor ($\eta = 1$) improves results over the baseline M-HRED+attention by ∼3.6% and ∼4.5% (R@1) given versions *v1* and *v2* of the MMD dataset. Choosing the optimal number of nearest neighbors for NNO ($\eta = 2$) improves results by further 0.4% (R@1) over NNO ($\eta = 1$) on both versions of MMD. Moreover, utilizing our subspace-based NEHA, we obtain 5.5% and 5.5% (R@1) improvement over the baseline M-HRED+attention given both versions of MMD. Our best performer, subspace-based NESA

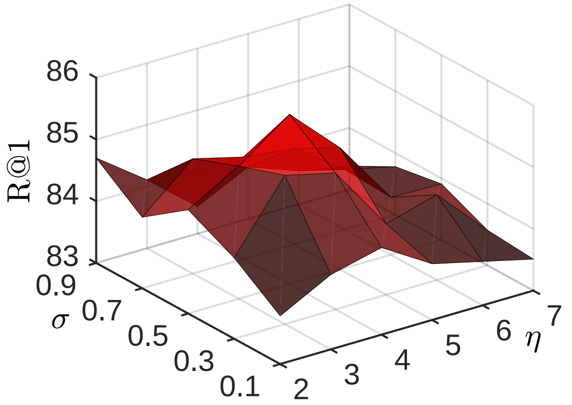


Figure 5.6: Performance (R@1) w.r.t. the number of nearest neighbors η and the RBF bandwidth σ on NESA ($\eta' = 2$).

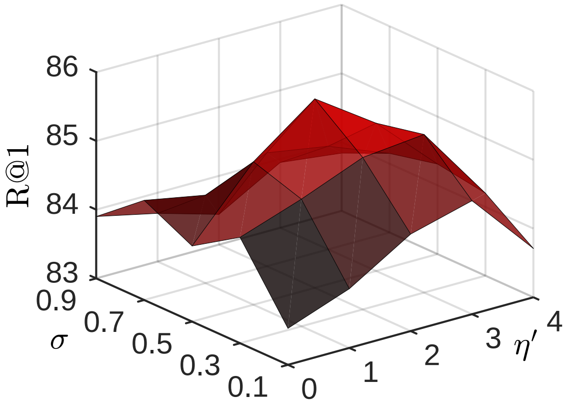


Figure 5.7: Performance (R@1) w.r.t. the number of leading eigenvectors η' and the RBF bandwidth σ on NESA ($\eta = 4$).

yields **6.3%** and **7.2%** (R@1) improvement over the baseline M-HRED+attention model.

Text-based Task. Table 5.1 shows results (BLEU and NIST metrics) by comparing target sentences against predicted sentences. Pre-training Text Decoder with French language prior to fine-tuning on English improves results by ~4% and ~2.1% (BLEU) over the M-HRED baseline on both MMD *v1* and *v2*. Using random word deletions for augmentation yielded gain of 0.16% (BLEU) over the M-HRED baseline (MMD *v2*). Augmentations via so-called sentence compression [Hou et al., 2020] scored ~1% over M-HRED, whereas augmentations via the so-called back-translation (using translating model [Ott et al., 2018]) scored ~2.4% over M-HRED. Pre-training on SIMMC [Moon et al., 2020] was marginally worse (and very similar to combined training on MMD+SIMMC). However, using the assisted supervision, that is, an auxiliary decoder for French, improves results by further ~3.5% (BLEU) over the M-HRED baseline (MMD *v2*). Augmentations by back translation require translating sentences twice English→French→English (additional 20 hours), whereas our assisted supervision requires only English→French translation. Adding auxiliary German and Russian decoders (to French) and the main decoder for English yields over **4**% (BLEU) over the M-HRED baseline (MMD *v2*). Finally, using the transformer backbone [Vaswani et al., 2017] results in a ∼5% boost. The benefit of adding multiple auxiliary language decoders is clear. In what follows, we use the GRU backbone not transformers (the backbone choice is a secondary matter). Pre-training the text backbone on the SIMMC dataset [Moon et al., 2020] before applying our assisted step may also boost results. Applying the sentence compression model [Hou et al., 2020] via an auxiliary decoder (in addition to French, German and Russian) in our assisted supervision is also possible.

Below, we present three randomly selected dialogues. Responses generated by our system, denoted by (*our_pred*), appear to make fewer mistakes and be more concise.

Example I:

User: *Show me something like the one in 4th image.*

Agent: *The similar looking ones are img_201, img_320, img_513.*

User: *show me something that will pair well with the 2nd image?*

Target_A: *It can go well with black colors sports shoes.*

MMD_pred: *It can go well with regular style, regular fit, solid and regular style, solid and regular.*

Our_pred: *It can go well with casual fitted footwear.*

Example II:

User: *and what about for the 1st image?*

Agent: *Image from the front, back and left directions respectively*

User: *Which of the celebrities usually wear watch similar looking to the one in the 2nd image?*

Target_A: *Celebrities cel_278,cel_255 and cel_444 endorse this kind of watch.*

MMD_pred: *Celebrities cel_30 and cel_252 endorse this type of handbag.*

Our_pred: *Celebrities cel_60 and cel_255 endorse this kind of watch.*

Example III:

Agent: *May I please ask your gender.*

User: *Female, I am a 15 yr old female.*

Agent: *Ok, this is helpful.*

Target_A: *I'll just take a few seconds to go through my catalog of items.*

MMD_pred: *Absolutely. Please wait a bit while I search through my catalog quickly.*

Our_pred: *Good! Just wait a few seconds while I go through my item catalog.*

Mixed Task. Firstly, we evaluate our Task Discriminator on the MMD dataset (*v2*) and note that it achieves 97.0% accuracy. This means that results in Tables 5.1 and 5.2 represent upper bound scores for this paragraph as both tables report on two separate tasks (oracle knowledge regarding which task is which) according to protocol in Saha *et al.*[Saha et al., 2018]. Table 5.3 shows that results for the mixed task dropped marginally compared to results in Tables 5.1 and 5.2. Our best assisted supervision methods outperformed best baselines M-HRED and M-HRED+attention equipped with Task Discriminator by $\sim$4% (BLEU) and $\sim$7.4% (R@1).

Ablations on NEHA w.r.t. $\eta' \leq \eta$. Below we investigate the impact of subspace size w.r.t. η' and the impact of η nearest neighbors retrieved from DeepFashion on the performance of image-based task. Figure 5.4c shows that the best performance is attained for $\eta' = \eta = 4$ and the trend suggests that $\eta' \approx \eta$ is a good choice. Figure 5.4d shows that $\eta' = \eta = 5$ is a better choice for R@3, which allows two incorrect matches precede the correct one. Thus, including more nearest neighbors of positive/negative target images of MMD boosts the score.

Nearest Neighbors+the Hinge Loss. Positive/negative nearest neighbors retrieved from DeepFashion for positive/negative target images can be fed directly into our assisted supervision loss in Eq. (5.2). Figures 5.4c and 5.4d evaluate such a setting ($\eta'\!=\!0$) as it is a special case of our subspace-based approach if $\eta'\!=\!0$ (only μ^- and μ^+ are used if $\eta'\!=\!0$). On average, such a setting is $\sim$2% worse than the subspace-based context. Subspaces capture robustly second-order statistics by discarding eigenvalue scaling and the smallest factors.

User study. We asked 3 users to score our best performer *vs*. M-HRED on MMD (v2) (randomized test) in terms of *clarity*, *compactness* and *helpfulness* on 500 system responses. Table 5.4 shows that $\sim$61.0% responses of the assisted supervision were clearer and more helpful (*vs*. 39% of M-HRED). Both methods were generating similarly compact responses.

SIMMC. Table 5.6 shows that using the multilingual decoding head yields 2.3% and $\sim$10% gain (BLEU) on RNN and transformers backbone over the HRE baseline (see the SIMMC paper for details of HRE). Moreover, our visual NESA yielded $\sim$4% gain (R@1 score).

Runtimes. Our code is implemented in PyTorch and evaluated on an NVIDIA Tesla P100 (unless stated otherwise). Table 5.5 (runtimes for the text-based tasks) shows that the T-HRED and M-HRED baselines take $\sim$15 hours to train. Our assisted supervision (French) uses extra 5 hours. Translations are obtained off-line with translator [Ott et al., 2018]. However, the best augmentation strategy that we tried (back translation) takes 55 hours, whereas our assisted superv. takes 40 hours (including translation time). Table 5.7 (runtimes for the image-based tasks) shows that the T-HRED and M-HRED baselines take $\sim$15 hours to train. Nearest Neighbor Only, NEHA and NESA require 1, 3, and 4 extra hours. The off-line pre-processing includes the ResNet-50 feature extraction from MMD and DeepFashion (6 hours), nearest neighbor search with FAISS [Johnson et al., 2019] (1.5 hours, 4 GPUs) and running SVD (2 hours, 4 GPUs).

5.6 Conclusions

We have introduced the assisted supervision which boosts the performance by leveraging AUDITED. Sampling auxiliary nearest neighbors from the natural manifold of fashion images helps create a meaningful visual context for the image task. With appropriate Soft Assignment reweighting and subspace modeling, benefits become clear while (by design) not posing any extra burden at the testing time. Learning to decode target dialogue sentences in several languages helps reduce the noise of single language syntactic.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have worked on Deep Learning based Domain Adaptation problems. Throughout research conducted in this thesis, we have demonstrated the alignment problem in Domain Adaptation is complex and requires a better understanding. We have proposed several improved solutions, a new dataset, demonstrations of Deep Learning methods on different Domain Adaptation problems, and extensive experiments to address this problem.

(i) We proposed and formulated an improved alignment loss based on a mixture of alignments of second- or higher-order scatter tensors. Our method exploits the labeled samples in the target domain to transfer knowledge class-by-class from the source domain. Domain shift between the source and the target domains creates challenging problems in tasks like image recognition and others. By utilizing class-wise statistics between these domains, we learned a better alignment between source and target. One challenge in this task is to manage the alignment of each class separately, as not all classes need the same amount of alignment. To address this problem, we added a trainable weight for each class to control their alignment. This trainable weight has shown improved results but not as much as we expected. We believe it is still open to improvements with further research on the topic. Another problem we have faced with our proposed alignment is that it was not easy to tune the parameters. Introducing many control parameters required extensive experimentation on a validation set to find out good combinations of parameters. We have presented the state of the results on several datasets, reaching 90+% accuracy results on the commonly used Office dataset [Saenko et al., 2010]. This could indicate saturated results on the Office dataset, which we addressed in the next chapter.

(ii) Saturated results on Office dataset [Saenko et al., 2010] and the simple domain shift between source and target splits inspired us to create Open MIC dataset. We collected, processed, annotated, and presented a new larger, challenging Open MIC dataset from 10 different exhibitions. The annotation process required manual labeling and required a lot of time to process the whole set. We have utilized supporting software to make our job easier and shorter, but due to extensive manual work required, we might have unintentionally introduced a small number of noisy labels. We have provided several benchmark results on our dataset for both SDA and UDA on each split. Also, we have further improved our alignment loss for SDA by utilizing non-Euclidean distances as the alignment metric. Training on non-Euclidean distances was a challenging task. Computational expensiveness and memory footprints of such methods inspired us to resort to Nyström projections to make the training of deep networks tractable non-Euclidean metrics. Although JBLD and AIRM metrics have shown improvements over the baseline Euclidean metric, on some splits, improvements were negligible or non-existent.

(iii) We have extended our research space from RGB image-to-image DA to Action Recognition using 3D body skeleton joints. We have demonstrated that sequences of these joints can be encoded into texture-like feature maps with the linearization of carefully designed kernel functions. Deciding on the number of pivots and other parameters to linearize the kernel was one of the main challenges we have faced in creating such feature maps. Also, the number of pivots affects the dimensionality of the output, which is used as input to CNNs. This relation limited the range of parameters we could test, thus making it harder to tune and find optimal parameters. We have presented an SDA approach by using these feature maps as inputs to generic pre-trained CNNs. We have improved our mixture of alignment loss to work with domains where they only have partially overlapping classes. This change worked well in the action recognition problem, allowing us to use multiple source domains together for partial SDA.

(iv) We moved to a more challenging multi-domain problem, multimodal conversation systems. MMD data [Saha et al., 2018] provides multimodal dialogues, which are sequences of utterances where the modality is text, image, or both. To learn a common feature space for multimodal utterances, initially, we have tried a similar method we used in 4 to create feature maps from sequences of utterances together with a generative model to hallucinate these feature maps at the output. This approach did not work well for this type of task with sequences of different modalities. We have changed our focus to adapting external data through assisted supervision to improve image and text tasks. We have demonstrated that simple French translations based on external knowledge for text tasks improved the model's sentence predictions in English. For the image task, to enhance image ranking results, we have utilized external data through searching and embedding nearest neighbor images of targets with our proposed methods. Benchmarks and tasks presented by Saha et al. limited our scope of experimentation by dividing tasks into two separate categories.

6.2 Future Work

We have designed a mixture of alignment loss to improve Supervised Domain Adaptation to align each class using second- or higher-order scatter tensors. We have demonstrated various proposed loss applications that learn a better domain alignment while maintaining inter-class variance. But there is still work to be done on this topic. Here, we list some future work ideas:

(i) In Chapter 2 and 3, we demonstrate our mixture of alignment loss which aligns the matching classes from source and target domains based on second- or higher-order scatter tensors. This loss is integrated into the final layer of a two-stream network to calculate alignment loss from outputs of source and target networks. This is a relatively trivial approach in terms of where and how it is placed into the network. It is possible to calculate alignment loss from outputs of several different layers of the network. Although the final layer of the network provides distinctive feature vectors, it is still possible to learn different feature space alignments from earlier network layers. Future work could design our loss in a multi-layer fashion where the alignment loss is calculated from multiple layer levels, not only from the final layer.

(ii) Following the previous point, our loss is designed to align two domains class-by-class by using a mixture of alignments. Although it works well and achieves state-of-the-art results, it depends on many parameters to accomplish this class-by-class alignment. Future work can address this problem by aiming to reduce the number of hyper-parameters required, or at least simplifying the tuning process would be a good start point for future work.

(iii) In Chapter 4, we have demonstrated a method to encode sequences of 3D body skeleton joints into texture-like feature maps. This is a preprocessing step that processes sequences of skeleton joints into one image-like result. Future work can address this issue and propose a model that can be trained end-to-end manner. This would further simplify the process. Also, it creates an opportunity for the network to learn weights that would achieve the best results. Deep learning has shown us that learned features through CNNs work much better than handcrafted features [Krizhevsky et al., 2012].

(iv) In Chapter 5, we have moved to a new but more challenging multimodal learning problem. This problem requires the model to use multimodal dialogue data to learn some specific task, for example, generating the following sentence from the given dialogue context. Tasks defined by [Saha et al., 2018] are image and text tasks that require two separate models and two separate training. We followed their methodology to compare our proposed methods against their results, but having two individual tasks for multimodal data breaks its purpose where the output could be an image or text depending on the given text. Future work could address this issue by developing a new methodology of a combined task. The model must learn the type of utterance from the given context, and output should depend on the predicted type. This improvement would create a more realistic conversation system benchmark for future work and allow researchers to simulate and create full dialogue outputs from the trained model either from a given context or from a starting utterance.

(v) In M-HRED [Saha et al., 2018] model, given utterance's image and text embeddings are concatenated to generate utterance's representation vector. This is a rather trivial approach to utilizing multimodality in the given utterance. Future work could address this by looking into ways to utilize multimodality in utterances better.

Appendices

Appendix A

Derivatives of the alignment loss g w.r.t. the feature vectors

Suppose $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N]$ and $\boldsymbol{\Phi}^* = [\boldsymbol{\phi}^*_!, ..., \boldsymbol{\phi}^*_N]$ are some feature vectors of quantity N and N^*, respectively, which are used to evaluate $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^*$. For $r = 2$, we have to first compute the derivative of the covariance matrix $\boldsymbol{\Sigma}$ w.r.t. $\boldsymbol{\phi}_{m'n'}$. To do so, we proceed by computing derivatives of: i) the autocorrelation matrix in (A.1) and ii) the outer product of means $\boldsymbol{\mu}$ in (A.2) and (A.3):

$$\frac{\partial \sum_n \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T}{\partial \phi_{m'n'}} = \boldsymbol{j}_{m'} \boldsymbol{\phi}_{n'}^T + \boldsymbol{\phi}_{n'} \boldsymbol{j}_{m'}^T, \tag{A.1}$$

$$\frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \mu_{m'}} = \boldsymbol{j}_{m'} \boldsymbol{\mu}^T + \boldsymbol{\mu} \boldsymbol{j}_{m'}^T, \tag{A.2}$$

$$\frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \phi_{m'n'}} = \sum_m \frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \mu_m} \frac{\partial \mu_m}{\partial \phi_{m'n'}} = \frac{1}{N} \left(\boldsymbol{j}_{m'} \boldsymbol{\mu}^T + \boldsymbol{\mu} \boldsymbol{j}_{m'}^T \right), \tag{A.3}$$

where $\boldsymbol{j}_{m'}$ is a vector of zero entries except for position m' which is equal one. Putting together (A.1), (A.2) and (A.3) yields the derivative of $\boldsymbol{\Sigma}$ w.r.t. $\boldsymbol{\phi}_{m'n'}$:

$$\frac{\partial \left(\frac{1}{N} \sum_n \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \right) - \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \phi_{m'n'}} = \frac{1}{N} \left(\boldsymbol{j}_{m'} \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu} \right)^T + \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu} \right) \boldsymbol{j}_{m'}^T \right). \tag{A.4}$$

The derivatives of $||\boldsymbol{\Sigma} - \boldsymbol{\Sigma}^*||_F^2$ w.r.t. covariance $\boldsymbol{\Sigma}$ as well as $\boldsymbol{\phi}_{m'n'}$ and $\boldsymbol{\phi}^*_{m'n'}$ are provided below:

$$\frac{\partial||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial\boldsymbol{\Sigma}}=2\left(\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*\right) \tag{A.5}$$

$$\begin{aligned}\frac{\partial||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial\phi_{m'n'}}&=\sum_{m,n}\frac{\partial||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial\Sigma_{mn}}\left(\frac{\partial\boldsymbol{\Sigma}}{\partial\phi_{m'n'}}\right)_{mn}\\&=\frac{2}{N}\sum_{m,n}(\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*)_{mn}\left(\boldsymbol{j}_{m'}\left(\boldsymbol{\phi}_{n'}-\boldsymbol{\mu}\right)^T+\left(\boldsymbol{\phi}_{n'}-\boldsymbol{\mu}\right)\boldsymbol{j}_{m'}^T\right)_{mn}\\&=\frac{4}{N}\left(\boldsymbol{\Sigma}_{m',:}-\boldsymbol{\Sigma}^*_{m',:}\right)\left(\boldsymbol{\phi}_{n'}-\boldsymbol{\mu}\right).\end{aligned} \tag{A.6}$$

The derivatives of $||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2$ w.r.t. $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ are:

$$\frac{\partial||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial\boldsymbol{\Phi}}=\frac{4}{N}(\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*)\left(\boldsymbol{\Phi}-\boldsymbol{\mu}\mathbb{1}^T\right), \tag{A.7}$$

$$\frac{\partial||\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*||_F^2}{\partial\boldsymbol{\Phi}^*}=-\frac{4}{N^*}(\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^*)\left(\boldsymbol{\Phi}^*-\boldsymbol{\mu}^*\mathbb{1}^T\right). \tag{A.8}$$

The derivatives of $||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2$ w.r.t. $\boldsymbol{\mu}$, $\boldsymbol{\phi}_n$ and $\boldsymbol{\phi}^*_{n'}$ are:

$$\frac{\partial||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial\boldsymbol{\mu}}=2\left(\boldsymbol{\mu}-\boldsymbol{\mu}^*\right), \tag{A.9}$$

$$\frac{\partial||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial\boldsymbol{\phi}_{n'}}=\frac{2\left(\boldsymbol{\mu}-\boldsymbol{\mu}^*\right)}{N},\quad\frac{\partial||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial\boldsymbol{\phi}^*_{n'}}=\frac{2\left(\boldsymbol{\mu}-\boldsymbol{\mu}^*\right)}{N^*}. \tag{A.10}$$

Kernelized derivative of the Frobenius norm between tensors w.r.t. the feature vectors

Suppose that some feature vectors $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N]$ and $\boldsymbol{\Phi}^* = [\boldsymbol{\phi}^*_!, ..., \boldsymbol{\phi}^*_N]$ are given in quantities N and N^* and that the Frobenius norm between tensors $\boldsymbol{\mathcal{X}}^{(r)}$ and $\boldsymbol{\mathcal{Y}}^{(r)}$ of order $r \geq 1$ build from $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ is being evaluated. Then, the derivative of Equation (2.8) w.r.t. feature vector $\boldsymbol{\phi}_{n^\ddagger}$ becomes:

$$\begin{aligned}\frac{\partial||\boldsymbol{\mathcal{X}}^{(r)} - \boldsymbol{\mathcal{X}}^{*(r)}||_F^2}{\partial\boldsymbol{\phi}_{n^\ddagger}} &= \frac{1}{N^2} r \sum_{n=1}^{N} \sum_{n'=1}^{N} K_{nn'}^{r-1} \frac{\partial K_{nn'}}{\partial\boldsymbol{\phi}_{n^\ddagger}} \\ &- \frac{2}{NN^*} r \sum_{n=1}^{N} \sum_{n'=1}^{N^*} \bar{\bar{K}}_{nn'}^{r-1} \frac{\partial \bar{\bar{K}}_{nn'}}{\partial\boldsymbol{\phi}_{n^\ddagger}},\end{aligned} \tag{B.1}$$

where

$$\begin{aligned}\frac{\partial K_{nn'}}{\partial\boldsymbol{\phi}_{n^\ddagger}} &= \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\phi}_{n'} \right\rangle}{\partial\boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\phi}_{n'} \right\rangle}{\partial\boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\mu} \right\rangle}{\partial\boldsymbol{\phi}_{n^\ddagger}} + \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\mu} \right\rangle}{\partial\boldsymbol{\phi}_{n^\ddagger}} \\ &= \left\{\begin{array}{l} \boldsymbol{\phi}_{n'} \\ \boldsymbol{\phi}_n \\ 2\boldsymbol{\phi}_n \\ 0 \end{array}\right. - \begin{array}{c} \frac{\boldsymbol{\phi}_{n'}}{N} \\ (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_{n'}}{N}) \\ (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_{n'}}{N}) \\ \frac{\boldsymbol{\phi}_{n'}}{N} \end{array} - \begin{array}{c} (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_n}{N}) \\ \frac{\boldsymbol{\phi}_n}{N} \\ (\boldsymbol{\mu} + \frac{\boldsymbol{\phi}_n}{N}) \\ \frac{\boldsymbol{\phi}_n}{N} \end{array} + \tfrac{2}{N}\boldsymbol{\mu} \begin{array}{l} \text{, if } n = n^\ddagger, n' \neq n^\ddagger \\ \text{, if } n \neq n^\ddagger, n' = n^\ddagger \\ \text{, if } n = n^\ddagger, n' = n^\ddagger \\ \text{, if } n \neq n^\ddagger, n' \neq n^\ddagger, \end{array}\end{aligned} \tag{B.2}$$

$$
\begin{aligned}
\frac{\partial \bar{\bar{K}}_{nn'}}{\partial \boldsymbol{\phi}_{n^\ddagger}} &= \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\phi}^*_{n'} \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\phi}^*_{n'} \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} - \frac{\partial \left\langle \boldsymbol{\phi}_n, \boldsymbol{\mu}^* \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} + \frac{\partial \left\langle \boldsymbol{\mu}, \boldsymbol{\mu}^* \right\rangle}{\partial \boldsymbol{\phi}_{n^\ddagger}} \\
&= \begin{cases}
\boldsymbol{\phi}^*_{n'} & \qquad\qquad \boldsymbol{\mu}^* & \quad, \text{if } n = n^\ddagger, n' \neq n^\ddagger \\
0 & \qquad\qquad 0 & \quad, \text{if } n \neq n^\ddagger, n' = n^\ddagger \\
\quad -\frac{1}{N}\boldsymbol{\phi}^*_{n'} \quad - & \qquad\qquad +\frac{1}{N}\boldsymbol{\mu}^* & \\
\boldsymbol{\phi}^*_{n} & \qquad\qquad \boldsymbol{\mu}^* & \quad, \text{if } n = n^\ddagger, n' = n^\ddagger \\
0 & \qquad\qquad 0 & \quad, \text{if } n \neq n^\ddagger, n' \neq n^\ddagger.
\end{cases}
\end{aligned}
\tag{B.3}
$$

Putting together Equations (B.1), (B.2) and (B.3) and setting $q = r - 1$ yields the derivatives w.r.t. matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$:

$$
\begin{aligned}
&\frac{\partial ||\boldsymbol{\mathcal{X}}^{(r)} - \boldsymbol{\mathcal{X}}^{*(r)}||_F^2}{\partial \boldsymbol{\Phi}} = \frac{2}{N^2} r \boldsymbol{\Phi} \left(\mathbf{K}^{qT} - \frac{1}{N} (\mathbb{1}^T \mathbf{K}^q)^T \mathbb{1}^T \right) \\
&+ \frac{2r\boldsymbol{\mu}}{N^2} \left(\frac{1}{N} \mathbb{1}^T \mathbf{K}^q \mathbb{1} - \mathbb{1}^T \mathbf{K}^q \right) - \frac{2r\boldsymbol{\Phi}^*}{NN^*} \left(\bar{\bar{\mathbf{K}}}^{q^T} - \frac{1}{N} (\bar{\bar{\mathbf{K}}}^{q^T} \mathbb{1}) \mathbb{1}^T \right) \\
&+ \frac{2}{NN^*} r \boldsymbol{\mu}^* \left(\mathbb{1}^T \bar{\bar{\mathbf{K}}}^{q^T} - \frac{1}{N} \mathbb{1}^T \bar{\bar{\mathbf{K}}}^{q^T} \mathbb{1} \right)
\end{aligned}
\tag{B.4}
$$

and

$$
\begin{aligned}
&\frac{\partial ||\boldsymbol{\mathcal{X}}^{(r)} - \boldsymbol{\mathcal{X}}^{*(r)}||_F^2}{\partial \boldsymbol{\Phi}^*} = \frac{2}{N^{*2}} r \boldsymbol{\Phi}^* \left(\bar{\mathbf{K}}^{qT} - \frac{1}{N^*} (\mathbb{1}^T \bar{\mathbf{K}}^q)^T \mathbb{1}^T \right) \\
&+ \frac{2r\boldsymbol{\mu}^*}{N^{*2}} \left(\frac{1}{N^*} \mathbb{1}^T \bar{\mathbf{K}}^q \mathbb{1} - \mathbb{1}^T \bar{\mathbf{K}}^q \right) - \frac{2r\boldsymbol{\Phi}}{NN^*} \left(\bar{\bar{\mathbf{K}}}^q - \frac{1}{N^*} (\bar{\bar{\mathbf{K}}}^q \mathbb{1}) \mathbb{1}^T \right) \\
&+ \frac{2}{NN^*} r \boldsymbol{\mu} \left(\mathbb{1}^T \bar{\bar{\mathbf{K}}}^q - \frac{1}{N^*} \mathbb{1}^T \bar{\bar{\mathbf{K}}}^q \mathbb{1} \right).
\end{aligned}
\tag{B.5}
$$

Appendix C

Derivatives of d^2 and d'^2 w.r.t. feat. vectors

Suppose $\boldsymbol{\Phi}=[\boldsymbol{\phi}_1,...,\boldsymbol{\phi}_N]$ and $\boldsymbol{\Phi}^*=[\boldsymbol{\phi}^*_1,...,\boldsymbol{\phi}^*_N]$ are some feature vectors of quantity N and N^*, respectively, which are used to evaluate $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^*$. We have to first compute the derivative of the covariance matrix $\boldsymbol{\Sigma}$ w.r.t. $\boldsymbol{\phi}_{m'n'}$. We proceed by computing der. of: i) the autocorrelation matrix in (A.1) and ii) the outer product of means $\boldsymbol{\mu}$ in (A.2) and (A.3):

$$\frac{\partial \sum_n \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T}{\partial \phi_{m'n'}} = \boldsymbol{j}_{m'} \boldsymbol{\phi}_{n'}^T + \boldsymbol{\phi}_{n'} \boldsymbol{j}_{m'}^T, \tag{C.1}$$

$$\frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \mu_{m'}} = \boldsymbol{j}_{m'} \boldsymbol{\mu}^T + \boldsymbol{\mu} \boldsymbol{j}_{m'}^T, \tag{C.2}$$

$$\frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \phi_{m'n'}} = \sum_m \frac{\partial \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \mu_m} \frac{\partial \mu_m}{\partial \phi_{m'n'}} = \frac{1}{N}\left(\boldsymbol{j}_{m'} \boldsymbol{\mu}^T + \boldsymbol{\mu} \boldsymbol{j}_{m'}^T\right), \tag{C.3}$$

where $\boldsymbol{j}_{m'}$ is a vector of zero entries except for position m' which is equal one. Putting together (A.1), (A.2) and (A.3) yields the derivative of $\boldsymbol{\Sigma}$ w.r.t. $\boldsymbol{\phi}_{m'n'}$:

$$\frac{\partial \left(\frac{1}{N}\sum_n \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T\right) - \boldsymbol{\mu}\boldsymbol{\mu}^T}{\partial \phi_{m'n'}} = \frac{1}{N}\left(\boldsymbol{j}_{m'}\left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu}\right)^T + \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu}\right)\boldsymbol{j}_{m'}^T\right). \tag{C.4}$$

The derivatives of d_g^2 w.r.t. covariance $\boldsymbol{\Sigma}$ as well as $\boldsymbol{\phi}_{m'n'}$ and $\boldsymbol{\phi}^*_{m'n'}$ are provided below:

$$\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \boldsymbol{\Sigma}} = 2d\left(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*\right)\frac{\partial d(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \boldsymbol{\Sigma}} \tag{C.5}$$

$$\begin{aligned}\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \phi_{m'n'}} &= \sum_{m,n} \frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \Sigma_{mn}} \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \phi_{m'n'}}\right)_{mn} \\ &= \frac{1}{N}\sum_{m,n} \frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial \Sigma_{mn}} \left(\boldsymbol{j}_{m'}\left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu}\right)^T + \left(\boldsymbol{\phi}_{n'} - \boldsymbol{\mu}\right)\boldsymbol{j}_{m'}^T\right)_{mn}. \end{aligned} \tag{C.6}$$

The derivatives of $d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)$ (after simplifying summations) w.r.t. $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ are:

$$\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial\boldsymbol{\Phi}} = \frac{2}{N}\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial\boldsymbol{\Sigma}}\left(\boldsymbol{\Phi}-\boldsymbol{\mu}\mathbb{1}^T\right), \tag{C.7}$$

$$\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial\boldsymbol{\Phi}^*} = \frac{2}{N^*}\frac{\partial d^2(\boldsymbol{\Sigma},\boldsymbol{\Sigma}^*)}{\partial\boldsymbol{\Sigma}^*}\left(\boldsymbol{\Phi}^*-\boldsymbol{\mu}^*\mathbb{1}^T\right). \tag{C.8}$$

The derivatives of d'^2 w.r.t. $\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^*$ are derived from:

$$\sum_{m,n}\frac{\partial d'^2}{\partial\Phi'_{mn}}\frac{\partial(\boldsymbol{Ƶ}\Phi)_{mn}}{\partial\boldsymbol{\Phi}} = \frac{2\boldsymbol{Ƶ}^T}{N}\frac{\partial d^2(\boldsymbol{\Sigma}',\boldsymbol{\Sigma}^{'*})}{\partial\boldsymbol{\Sigma}'}\left(\boldsymbol{\Phi}'-\boldsymbol{\mu}'\mathbb{1}^T\right), \tag{C.9}$$

where $\boldsymbol{\Phi}'=\boldsymbol{Ƶ}\boldsymbol{\Phi}$, $\boldsymbol{\Phi}'^*=\boldsymbol{Ƶ}\boldsymbol{\Phi}^*$, $\boldsymbol{\mu}'=\boldsymbol{Ƶ}\boldsymbol{\mu}$ and $\boldsymbol{\mu}'^*=\boldsymbol{Ƶ}\boldsymbol{\mu}^*$ and $\boldsymbol{Ƶ}$ is some projection matrix. We get the following derivatives:

$$\frac{\partial d'^2}{\partial\boldsymbol{\Phi}} = \frac{2\boldsymbol{Ƶ}^T}{N}\frac{\partial d'^2}{\partial\boldsymbol{\Sigma}'}\left(\boldsymbol{\Phi}'-\boldsymbol{\mu}'\mathbb{1}^T\right), \quad \frac{\partial d'^2}{\partial\boldsymbol{\Phi}^*} = -\frac{2\boldsymbol{Ƶ}^T}{N}\frac{\partial d'^2}{\partial\boldsymbol{\Sigma}'^*}\left(\boldsymbol{\Phi}'^*-\boldsymbol{\mu}'^*\mathbb{1}^T\right). \tag{C.10}$$

Lastly, based on our Proposition 4 in the main submission, we know that our particular choice $\boldsymbol{Ƶ}$ deems $d^2=d'^2$, therefore $\frac{\partial d^2}{\partial\boldsymbol{\Phi}}=\frac{\partial d'^2}{\partial\boldsymbol{\Phi}}$ and $\frac{\partial d^2}{\partial\boldsymbol{\Phi}^*}=\frac{\partial d'^2}{\partial\boldsymbol{\Phi}^*}$.

The derivatives of $||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2$ w.r.t. $\boldsymbol{\mu}$, $\boldsymbol{\phi}_n$ and $\boldsymbol{\phi}^*_{n'}$ are:

$$\frac{\partial||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial\boldsymbol{\mu}} = 2\left(\boldsymbol{\mu}-\boldsymbol{\mu}^*\right), \tag{C.11}$$

$$\frac{\partial||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial\boldsymbol{\phi}_{n'}} = \frac{2\left(\boldsymbol{\mu}-\boldsymbol{\mu}^*\right)}{N}, \quad \frac{\partial||\boldsymbol{\mu}-\boldsymbol{\mu}^*||_2^2}{\partial\boldsymbol{\phi}^*_{n'}} = \frac{2\left(\boldsymbol{\mu}-\boldsymbol{\mu}^*\right)}{N^*}. \tag{C.12}$$

Bibliography

ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIÉGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; AND ZHENG, X., TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/. Software available from tensorflow.org. 3

AGARWAL, S.; DUSEK, O.; KONSTAS, I.; AND RIESER, V., A knowledge-grounded multimodal search-based conversational agent. *arXiv preprint arXiv:1810.11954*, (2018). 76

AGRAWAL, P.; CARREIRA, J.; AND MALIK, J., Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, 37–45. 72

AL-RFOU, R.; ALAIN, G.; ALMAHAIRI, A.; ANGERMUELLER, C.; BAHDANAU, D.; BALLAS, N.; BASTIEN, F.; BAYER, J.; BELIKOV, A.; BELOPOLSKY, A.; BENGIO, Y.; BERGERON, A.; BERGSTRA, J.; BISSON, V.; BLEECHER SNYDER, J.; BOUCHARD, N.; BOULANGER-LEWANDOWSKI, N.; BOUTHILLIER, X.; DE BRÉBISSON, A.; BREULEUX, O.; CARRIER, P.-L.; CHO, K.; CHOROWSKI, J.; CHRISTIANO, P.; COOIJMANS, T.; CÔTÉ, M.-A.; CÔTÉ, M.; COURVILLE, A.; DAUPHIN, Y. N.; DELALLEAU, O.; DEMOUTH, J.; DESJARDINS, G.; DIELEMAN, S.; DINH, L.; DUCOFFE, M.; DUMOULIN, V.; EBRAHIMI KAHOU, S.; ERHAN, D.; FAN, Z.; FIRAT, O.; GERMAIN, M.; GLOROT, X.; GOODFELLOW, I.; GRAHAM, M.; GULCEHRE, C.; HAMEL, P.; HARLOUCHET, I.; HENG, J.-P.; HIDASI, B.; HONARI, S.; JAIN, A.; JEAN, S.; JIA, K.; KOROBOV, M.; KULKARNI, V.; LAMB, A.; LAMBLIN, P.; LARSEN, E.; LAURENT, C.; LEE, S.; LEFRANCOIS, S.; LEMIEUX, S.; LÉONARD, N.; LIN, Z.; LIVEZEY, J. A.; LORENZ, C.; LOWIN, J.; MA, Q.; MANZAGOL, P.-A.; MASTROPIETRO, O.; MCGIBBON, R. T.; MEMISEVIC, R.; VAN MERRIËNBOER, B.; MICHALSKI, V.; MIRZA, M.; ORLANDI, A.; PAL, C.; PASCANU, R.; PEZESHKI, M.; RAFFEL, C.; RENSHAW, D.; ROCKLIN, M.; ROMERO, A.; ROTH, M.; SADOWSKI, P.; SALVATIER, J.; SAVARD, F.; SCHLÜTER, J.; SCHULMAN, J.; SCHWARTZ, G.; SERBAN, I. V.; SERDYUK, D.; SHA-

BANIAN, S.; SIMON, E.; SPIECKERMANN, S.; SUBRAMANYAM, S. R.; SYGNOWSKI, J.; TANGUAY, J.; VAN TULDER, G.; TURIAN, J.; URBAN, S.; VINCENT, P.; VISIN, F.; DE VRIES, H.; WARDE-FARLEY, D.; WEBB, D. J.; WILLSON, M.; XU, K.; XUE, L.; YAO, L.; ZHANG, S.; AND ZHANG, Y., Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688 (May 2016). http://arxiv.org/abs/1605.02688. 3

AMEIXA, D.; COHEUR, L.; FIALHO, P.; AND QUARESMA, P., Luke, I am your father: dealing with out-of-domain requests by using movies subtitles. In *International Conference on Intelligent Virtual Agents*, 13–21. Springer. 69

ANIRUDH, R.; TURAGA, P.; SU, J.; AND SRIVASTAVA, A., Elastic functional coding of human actions: From vector-fields to latent variables. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3147–3155. 64

ANTOL, S.; AGRAWAL, A.; LU, J.; MITCHELL, M.; BATRA, D.; LAWRENCE ZITNICK, C.; AND PARIKH, D., Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433. 69

BANCHS, R. E., Movie-DiC: a movie dialogue corpus for research and development. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 203–207. 69

BAÑÓN, M.; CHEN, P.; HADDOW, B.; HEAFIELD, K.; HOANG, H.; ESPLÀ-GOMIS, M.; FORCADA, M.; KAMRAN, A.; KIREFU, F.; KOEHN, P.; ORTIZ-ROJAS, S.; PLA, L.; RAMÍREZ-SÁNCHEZ, G.; SARRÍAS, E.; STRELEC, M.; THOMPSON, B.; WAITES, W.; WIGGINS, D.; AND ZARAGOZA, J., ParaCrawl: Web-Scale Acquisition of Parallel Corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4555–4567. doi:10.18653/v1/2020.acl-main.417. https://acl2020.org/. 76

BAXTER, J.; CARUANA, R.; MITCHELL, T.; PRATT, L. Y.; SILVER, D. L.; AND THRUN, S., Learning to learn: Knowledge consolidation and transfer in inductive systems. In *NIPS Workshop, http://plato. acadiau. ca/courses/comp/dsilver/NIPS95_LTL/transfer. workshop*. 16, 32

BHATIA, R., 2009. *Positive definite matrices*, vol. 24. Princeton university press. 34, 37, 43

BHATTACHARYA, I.; CHOWDHURY, A.; AND RAYKAR, V. C., Multimodal dialog for browsing large visual catalogs using exploration-exploitation paradigm in a joint embedding space. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 187–191. 69

BO, L.; LAI, K.; REN, X.; AND FOX, D., Object recognition with hierarchical kernel descriptors. In *CVPR 2011*, 1729–1736. IEEE. 56

BO, L. AND SMINCHISESCU, C., Efficient match kernel between sets of features for visual recognition. In *Advances in neural information processing systems*, 135–143. 37

BOBICK, A. F. AND DAVIS, J. W., The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23, 3 (2001), 257–267. 10, 54

BOJAR, O.; BUCK, C.; FEDERMANN, C.; HADDOW, B.; KOEHN, P.; LEVELING, J.; MONZ, C.; PECINA, P.; POST, M.; SAINT-AMAND, H.; SORICUT, R.; SPECIA, L.; AND TAMCHYNA, A. S., Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 12–58. http://www.aclweb.org/anthology/W/W14/W14-3302. 76

CAVAZZA, J.; ZUNINO, A.; SAN BIAGIO, M.; AND MURINO, V., Kernelized covariance for action recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, 408–413. IEEE. 56

CHEN, L.; LI, W.; AND XU, D., Recognizing RGB images by learning from RGB-D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1418–1425. 26, 27

CHERIAN, A.; SRA, S.; BANERJEE, A.; AND PAPANIKOLOPOULOS, N., Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. *IEEE transactions on pattern analysis and machine intelligence*, 35, 9 (2012), 2161–2174. xxi, 33, 34, 36, 37, 43

CHOPRA, S.; BALAKRISHNAN, S.; AND GOPALAN, R., Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML workshop on challenges in representation learning*, vol. 2. xxi, 18, 19, 23, 26, 32, 35, 36, 44

CHU, W.-S.; DE LA TORRE, F.; AND COHN, J. F., Selective transfer machine for personalized facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3515–3522. 7

CHUNG, J.; GULCEHRE, C.; CHO, K.; AND BENGIO, Y., Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, (2014). 70, 71

COLLOBERT, R. AND WESTON, J., A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, 160–167. 54

CRAMMER, K.; KEARNS, M.; AND WORTMAN, J., Learning from multiple sources. *Journal of Machine Learning Research*, 9, Aug (2008), 1757–1774. 18

CYBENKO, G., Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2, 4 (1989), 303–314. 3

DAS, A.; KOTTUR, S.; GUPTA, K.; SINGH, A.; YADAV, D.; MOURA, J. M.; PARIKH, D.; AND BATRA, D., Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 326–335. 69

DAUMÉ III, H.; KUMAR, A.; AND SAHA, A., Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 53–59. Association for Computational Linguistics. 18, 32

DE LATHAUWER, L.; DE MOOR, B.; AND VANDEWALLE, J., A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21, 4 (2000), 1253–1278. 21

DEVLIN, J.; CHANG, M.-W.; LEE, K.; AND TOUTANOVA, K., Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, (2018). 2, 68

DOERSCH, C.; GUPTA, A.; AND EFROS, A. A., Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 1422–1430. 72

DOERSCH, C. AND ZISSERMAN, A., Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2051–2060. 72

DONAHUE, J.; JIA, Y.; VINYALS, O.; HOFFMAN, J.; ZHANG, N.; TZENG, E.; AND DARRELL, T., Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, 647–655. xxi, 26, 44, 54

DONG, C.; LOY, C. C.; HE, K.; AND TANG, X., Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38, 2 (2015), 295–307. ix, 2, 68

DOSOVITSKIY, A.; FISCHER, P.; SPRINGENBERG, J. T.; RIEDMILLER, M.; AND BROX, T., Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38, 9 (2015), 1734–1747. 72

DU, Y.; WANG, W.; AND WANG, L., Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1110–1118. 54, 64, 65

EITZ, M.; HAYS, J.; AND ALEXA, M., How do humans sketch objects? *ACM Transactions on graphics (TOG)*, 31, 4 (2012), 1–10. 27, 29

EVERINGHAM, M.; VAN GOOL, L.; WILLIAMS, C. K.; WINN, J.; AND ZISSERMAN, A., The PASCAL visual object classes challenge 2007 (VOC2007) results. (2007). 26, 29

FEI-FEI, L.; FERGUS, R.; AND PERONA, P., One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28, 4 (2006), 594–611. 18, 32

FEICHTENHOFER, C.; PINZ, A.; AND WILDES, R. P., Spatiotemporal residual networks for video action recognition. CoRR abs/1611.02155 (2016). *arXiv preprint arXiv:1611.02155*, (2016). 54, 55

FUKUSHIMA, K., Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36, 4 (1980), 193–202. 18

GAIDON, A.; HARCHAOUI, Z.; AND SCHMID, C., A time series kernel for action recognition. 56

GANIN, Y.; USTINOVA, E.; AJAKAN, H.; GERMAIN, P.; LAROCHELLE, H.; LAVIOLETTE, F.; MARCHAND, M.; AND LEMPITSKY, V., Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17, 1 (2016), 2096–2030. 18, 32, 35, 36

GEBRU, T.; HOFFMAN, J.; AND FEI-FEI, L., Fine-grained recognition in the wild: A multi-task domain adaptation approach. In *Proceedings of the IEEE International Conference on Computer Vision*, 1349–1358. 50

GHEISARI, M. AND BAGHSHAH, M. S., Unsupervised domain adaptation via representation learning and adaptive classifier learning. *Neurocomputing*, 165 (2015), 300–311. 7

GHIFARY, M.; KLEIJN, W. B.; AND ZHANG, M., Domain adaptive neural networks for object recognition. In *Pacific Rim international conference on artificial intelligence*, 898–904. Springer. xxi, 26, 44

GIDARIS, S.; SINGH, P.; AND KOMODAKIS, N., Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, (2018). 72

GIRSHICK, R., Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448. ix, 2

GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; AND MALIK, J., Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587. ix, 2, 18, 32, 54, 68

GONG, B.; GRAUMAN, K.; AND SHA, F., Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, 222–230. 7

GONG, B.; SHI, Y.; SHA, F.; AND GRAUMAN, K., Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2066–2073. IEEE. 33, 35

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; AND BENGIO, Y., Generative adversarial nets. In *Conference on Neural Information Processing Systems*. 72

GRAVES, A.; MOHAMED, A.-R.; AND HINTON, G., Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, 6645–6649. IEEE. 2, 68

GRIFFIN, G.; HOLUB, A.; AND PERONA, P., Caltech-256 object category dataset. (2007). 26

GROSS, R.; MATTHEWS, I.; COHN, J.; KANADE, T.; AND BAKER, S., Multi-pie. *Image and Vision Computing*, 28, 5 (2010), 807–813. 35

GU, J.; WANG, G.; CAI, J.; AND CHEN, T., An empirical study of language cnn for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, 1222–1231. 54

GUO, X.; WU, H.; GAO, Y.; RENNIE, S.; AND FERIS, R., Fashion IQ: A New Dataset towards Retrieving Images by Natural Language Feedback. *arXiv preprint arXiv:1905.12794*, (2019). 70

HARANDI, M. AND FERNANDO, B., Generalized backpropagation,\'{E} tude de cas: Orthogonality. *arXiv preprint arXiv:1611.05927*, (2016). 18

HE, K.; ZHANG, X.; REN, S.; AND SUN, J., Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. ix, 2, 55, 62

HERATH, S.; HARANDI, M.; AND PORIKLI, F., Going deeper into action recognition: A survey. *Image and vision computing*, 60 (2017), 4–21. 54

HERATH, S.; HARANDI, M.; AND PORIKLI, F., Learning an invariant hilbert space for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3845–3854. 18

HERATH, S.; HARANDI, M.; AND PORIKLI, F., Learning an invariant hilbert space for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3845–3854. 33, 35, 36

HINTON, G.; DENG, L.; YU, D.; DAHL, G. E.; MOHAMED, A.-R.; JAITLY, N.; SENIOR, A.; VANHOUCKE, V.; NGUYEN, P.; SAINATH, T. N.; ET AL., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29, 6 (2012), 82–97. 54

HJELM, R. D.; FEDOROV, A.; LAVOIE-MARCHILDON, S.; GREWAL, K.; BACHMAN, P.; TRISCHLER, A.; AND BENGIO, Y., Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, (2018). 72

HOCHREITER, S. AND SCHMIDHUBER, J., Long short-term memory. *Neural computation*, 9, 8 (1997), 1735–1780. 18

HOPFIELD, J. J., Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79, 8 (1982), 2554–2558. 18

HOU, W.; SUOMINEN, H.; KONIUSZ, P.; CALDWELL, S. B.; AND GEDEON, T., A Token-Wise CNN-Based Method for Sentence Compression. In *Neural Information Processing - 27th International Conference, ICONIP*, vol. 12532 of *Lecture Notes in Computer Science*, 668–679. Springer. doi:10.1007/978-3-030-63830-6_56. 76, 79, 80

HUSSEIN, M. E.; TORKI, M.; GOWAYYED, M. A.; AND EL-SABAN, M., Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Twenty-Third International Joint Conference on Artificial Intelligence*. 56

INTRATOR, N. AND EDELMAN, S., Making a low-dimensional representation suitable for diverse tasks. In *Learning to learn*, 135–157. Springer. 18

JEBARA, T.; KONDOR, R.; AND HOWARD, A., Probability product kernels. *Journal of Machine Learning Research*, 5, Jul (2004), 819–844. 58

JI, S.; XU, W.; YANG, M.; AND YU, K., 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35, 1 (2012), 221–231. 10, 54, 55

JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; AND DARRELL, T., Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093*, (2014). 3

JOHANSSON, G., Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14, 2 (1973), 201–211. 56

JOHNSON, J.; DOUZE, M.; AND JÉGOU, H., Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, (2019). xviii, xix, 69, 71, 74, 78, 81

KARPATHY, A.; TODERICI, G.; SHETTY, S.; LEUNG, T.; SUKTHANKAR, R.; AND FEI-FEI, L., Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725–1732. 10, 54, 55

KE, Q.; BENNAMOUN, M.; AN, S.; SOHEL, F.; AND BOUSSAID, F., A new representation of skeleton sequences for 3d action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3288–3297. 54, 56, 60, 63, 64, 65, 66

KIM, T.-K.; WONG, S.-F.; AND CIPOLLA, R., Tensor canonical correlation analysis for action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE. 20

KLASER, A.; MARSZAŁEK, M.; AND SCHMID, C., A spatio-temporal descriptor based on 3d-gradients. 10, 54

KONIUSZ, P. AND CHERIAN, A., Sparse coding for third-order super-symmetric tensor descriptors with application to texture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5395–5403. 20, 21

KONIUSZ, P.; CHERIAN, A.; AND PORIKLI, F., Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European Conference on Computer Vision*, 37–53. Springer. 20, 56, 64

KONIUSZ, P.; TAS, Y.; AND PORIKLI, F., Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4478–4487. xvii, xxi, 31, 32, 33, 34, 36, 38, 43, 44, 54, 55, 57, 61, 62, 68

KONIUSZ, P.; TAS, Y.; ZHANG, H.; HARANDI, M.; PORIKLI, F.; AND ZHANG, R., Museum exhibit identification challenge for the supervised domain adaptation and beyond. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 788–804. 68

KONIUSZ, P.; YAN, F.; GOSSELIN, P.-H.; AND MIKOLAJCZYK, K., Higher-order occurrence pooling for bags-of-words: Visual concept detection. *IEEE transactions on pattern analysis and machine intelligence*, 39, 2 (2016), 313–326. 20, 22, 68

KONIUSZ, P.; YAN, F.; AND MIKOLAJCZYK, K., Comparison of Mid-Level Feature Coding Approaches and Pooling Strategies in Visual Concept Detection. *Computer Vision and Image Understanding*, 117, 5 (May 2013), 479–492. doi:10.1016/j.cviu.2012.10.010. https://doi.org/10.1016/j.cviu.2012.10.010. 68

KONIUSZ, P. AND ZHANG, H., Power Normalizations in Fine-grained Image, Few-shot Image and Graph Classification. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE. doi:10.1109/TPAMI.2021.3107164. 68

KOTTUR, S.; MOURA, J. M.; PARIKH, D.; BATRA, D.; AND ROHRBACH, M., Clevr-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. *arXiv preprint arXiv:1903.03166*, (2019). 69

KRIZHEVSKY, A.; SUTSKEVER, I.; AND HINTON, G. E., Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105. ix, 1, 2, 16, 18, 27, 28, 32, 54, 62, 85

KUEHNE, H.; JHUANG, H.; GARROTE, E.; POGGIO, T.; AND SERRE, T., HMDB: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, 2556–2563. IEEE. 55

KUZBORSKIJ, I.; MARIA CARLUCCI, F.; AND CAPUTO, B., When naive Bayes nearest neighbors meet convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2100–2109. 18, 19, 28, 35, 36

LAI, K.; BO, L.; REN, X.; AND FOX, D., A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, 1817–1824. IEEE. 26

LAPTEV, I., On space-time interest points. *International journal of computer vision*, 64, 2-3 (2005), 107–123. 10, 54

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; AND HAFFNER, P., Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 11 (1998), 2278–2324. 1

LI, W.; TOMMASI, T.; ORABONA, F.; VÁZQUEZ, D.; LÓPEZ, M.; XU, J.; AND LAROCHELLE, H., Task-cv: Transferring and adapting source knowledge in computer vision. In *ECCV Workshop, http://adas. cvc. uab. es/task-cv2016*. 16, 32

LIU, J.; SHAHROUDY, A.; XU, D.; AND WANG, G., Spatio-temporal lstm with trust gates for 3d human action recognition. In *European conference on computer vision*, 816–833. Springer. 54, 64, 65

LIU, Z.; LUO, P.; QIU, S.; WANG, X.; AND TANG, X., Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1096–1104. xviii, xix, 68, 69, 71, 72, 73, 74, 76

LONG, M.; CAO, Y.; WANG, J.; AND JORDAN, M. I., Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, (2015). 18

LONG, M.; ZHU, H.; WANG, J.; AND JORDAN, M. I., Unsupervised domain adaptation with residual transfer networks. In *Advances in neural information processing systems*, 136–144. 49

LONG, M.; ZHU, H.; WANG, J.; AND JORDAN, M. I., Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2208–2217. JMLR. org. 49

LU, H.; PLATANIOTIS, K. N.; AND VENETSANOPOULOS, A. N., A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44, 7 (2011), 1540–1551. 20

LV, F. AND NEVATIA, R., Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *European conference on computer vision*, 359–372. Springer. 56

MAIRAL, J.; KONIUSZ, P.; HARCHAOUI, Z.; AND SCHMID, C., Convolutional kernel networks. In *Advances in neural information processing systems*, 2627–2635. 56, 60

MARIAN, V. AND SHOOK, A., The Cognitive Benefits of Being Bilingual. *Cerebrum : the Dana forum on brain science*, 2012 (10 2012), 13. 69, 73

MIKOLOV, T.; KARAFIÁT, M.; BURGET, L.; ČERNOCKỲ, J.; AND KHUDANPUR, S., Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*. 70

MOON, S.; KOTTUR, S.; CROOK, P. A.; DE, A.; PODDAR, S.; LEVIN, T.; WHITNEY, D.; DIFRANCO, D.; BEIRAMI, A.; CHO, E.; SUBBA, R.; AND GERAMIFARD, A., Situated and Interactive Multimodal Conversations. 76, 77, 79, 80

MOTIIAN, S. AND DORETTO, G., Information bottleneck domain adaptation with privileged information for visual recognition. In *European Conference on Computer Vision*, 630–647. Springer. 26, 27

NAIR, V. AND HINTON, G. E., Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814. 1

NATHAN SILBERMAN, P. K., DEREK HOIEM AND FERGUS, R., Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*. 8

OFLI, F.; CHAUDHRY, R.; KURILLO, G.; VIDAL, R.; AND BAJCSY, R., Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25, 1 (2014), 24–38. 56

OHN-BAR, E. AND TRIVEDI, M., Joint angles similarities and HOG2 for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 465–470. 56

OTT, M.; EDUNOV, S.; GRANGIER, D.; AND AULI, M., Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*. 73, 75, 76, 78, 79, 81

PAN, S. J.; TSANG, I. W.; KWOK, J. T.; AND YANG, Q., Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22, 2 (2010), 199–210. 7

PARAMESWARAN, V. AND CHELLAPPA, R., View invariance for human action recognition. *International Journal of Computer Vision*, 66, 1 (2006), 83–101. 56

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; ET AL., PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 8024–8035. 3

PENNEC, X.; FILLARD, P.; AND AYACHE, N., A Riemannian framework for tensor computing. *International Journal of computer vision*, 66, 1 (2006), 41–66. xxi, 33, 34, 36, 37, 43

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; AND ZETTLEMOYER, L., Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, (2018). 2

PRESTI, L. L. AND LA CASCIA, M., 3D skeleton-based human action classification: A survey. *Pattern Recognition*, 53 (2016), 130–147. 56

RAM, A.; PRASAD, R.; KHATRI, C.; VENKATESH, A.; GABRIEL, R.; LIU, Q.; NUNN, J.; HEDAYATNIA, B.; CHENG, M.; NAGAR, A.; ET AL., Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*, (2018). 68

REBUFFI, S.-A.; BILEN, H.; AND VEDALDI, A., Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, 506–516. 35

REN, S.; HE, K.; GIRSHICK, R.; AND SUN, J., Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99. ix, 2, 54

RITTER, A.; CHERRY, C.; AND DOLAN, B., Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 172–180. 69

ROZANTSEV, A.; SALZMANN, M.; AND FUA, P., Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41, 4 (2018), 801–814. 7

RUMELHART, D. E.; HINTON, G. E.; AND WILLIAMS, R. J., Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science. 1

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; ET AL., Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 3 (2015), 211–252. 2, 16, 27, 32, 34, 42

SAENKO, K.; KULIS, B.; FRITZ, M.; AND DARRELL, T., Adapting visual category models to new domains. In *European conference on computer vision*, 213–226. Springer. ix, xv, 5, 7, 8, 25, 31, 32, 33, 34, 62, 83

SAHA, A.; KHAPRA, M. M.; AND SANKARANARAYANAN, K., Towards building large scale multimodal domain-aware conversation systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*. x, xviii, 11, 13, 67, 68, 69, 70, 71, 73, 75, 76, 77, 81, 84, 85, 86

SCHOLKOPF, B. AND SMOLA, A. J., 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press. 10, 55

SCHULDT, C.; LAPTEV, I.; AND CAPUTO, B., Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, 32–36. IEEE. 9, 55

SERBAN, I. V.; SORDONI, A.; BENGIO, Y.; COURVILLE, A.; AND PINEAU, J., Building end-to-end dialogue systems using generative hierarchical neural network models. *arXiv preprint arXiv:1507.04808*, (2015). 70

SERBAN, I. V.; SORDONI, A.; BENGIO, Y.; COURVILLE, A.; AND PINEAU, J., Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*. 70, 71

SERBAN, I. V.; SORDONI, A.; LOWE, R.; CHARLIN, L.; PINEAU, J.; COURVILLE, A.; AND BENGIO, Y., A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*. 70

SERMANET, P.; EIGEN, D.; ZHANG, X.; MATHIEU, M.; FERGUS, R.; AND LECUN, Y., Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, (2013). 18, 32

SHAHROUDY, A.; LIU, J.; NG, T.-T.; AND WANG, G., Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1010–1019. 9, 55, 63, 65

SHANG, L.; LU, Z.; AND LI, H., Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, (2015). 70

SHASHUA, A. AND HAZAN, T., Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, 792–799. 20

SHIRI, F.; YU, X.; PORIKLI, F.; HARTLEY, R.; AND KONIUSZ, P., Identity-Preserving Face Recovery from Stylized Portraits. *International Journal of Computer Vision*, 127, 6-7 (2019), 863–883. doi:10.1007/s11263-019-01169-1. 72

SHIRI, F.; YU, X.; PORIKLI, F.; HARTLEY, R.; AND KONIUSZ, P., Recovering Faces From Portraits with Auxiliary Facial Attributes. In *Winter Conference on Applications of Computer Vision*, 406–415. 72

SHOTTON, J.; FITZGIBBON, A.; COOK, M.; SHARP, T.; FINOCCHIO, M.; MOORE, R.; KIPMAN, A.; AND BLAKE, A., Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, 1297–1304. Ieee. 56

SIMONYAN, K. AND ZISSERMAN, A., Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 568–576. ix, 2, 10, 54, 55

SIMONYAN, K. AND ZISSERMAN, A., Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, (2014). ix, 2, 18, 27, 28, 32, 34, 42, 70

SORDONI, A.; GALLEY, M.; AULI, M.; BROCKETT, C.; JI, Y.; MITCHELL, M.; NIE, J.-Y.; GAO, J.; AND DOLAN, B., A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*, (2015). 70

SUN, B.; FENG, J.; AND SAENKO, K., Return of frustratingly easy domain adaptation. In *Thirtieth AAAI Conference on Artificial Intelligence*. 18, 19, 29, 32, 33, 35, 36

SUN, B. AND SAENKO, K., Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, 443–450. Springer. 7

SUTSKEVER, I.; VINYALS, O.; AND LE, Q. V., Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112. 2

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; AND RABINOVICH, A., Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9. ix, 2, 42

TAS, Y. AND KONIUSZ, P., CNN-based action recognition and supervised domain adaptation on 3D body skeletons via kernel feature maps. *arXiv preprint arXiv:1806.09078*, (2018). 68

THOMASON, J.; MURRAY, M.; CAKMAK, M.; AND ZETTLEMOYER, L., Vision-and-dialog navigation. In *Conference on Robot Learning*, 394–406. 69

THRUN, S., Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, 640–646. 18

TOMMASI, T.; LANZI, M.; RUSSO, P.; AND CAPUTO, B., Learning the roots of visual domain shift. In *European Conference on Computer Vision*, 475–482. Springer. 18, 19, 35, 36

TOMMASI, T.; ORABONA, F.; AND CAPUTO, B., Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3081–3088. IEEE. 16, 18, 29, 32, 34

TOMMASI, T. AND TUYTELAARS, T., A testbed for cross-dataset analysis. In *European Conference on Computer Vision*, 18–31. Springer. 35

TZENG, E.; HOFFMAN, J.; DARRELL, T.; AND SAENKO, K., Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, 4068–4076. xxi, 7, 18, 26, 32, 35, 36, 44, 54

VAPNIK, V. AND VASHIST, A., A new learning paradigm: Learning using privileged information. *Neural networks*, 22, 5-6 (2009), 544–557. 16

VASILESCU, M. A. O. AND TERZOPOULOS, D., Multilinear analysis of image ensembles: Tensorfaces. In *European conference on computer vision*, 447–460. Springer. 20

VASILESCU, M. A. O. AND TERZOPOULOS, D., TensorTextures: Multilinear image-based rendering. In *ACM SIGGRAPH 2004 Papers*, 336–342. 20

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; AND POLOSUKHIN, I., Attention is All You Need. https://arxiv.org/pdf/1706.03762.pdf. xxii, xxiii, 76, 78, 80

VEDALDI, A. AND LENC, K., MatConvNet – Convolutional Neural Networks for MATLAB. In *Proceeding of the ACM Int. Conf. on Multimedia*. 3

VEMULAPALLI, R.; ARRATE, F.; AND CHELLAPPA, R., Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 588–595. 56, 64

VENKATESWARA, H.; EUSEBIO, J.; CHAKRABORTY, S.; AND PANCHANATHAN, S., Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5018–5027. 49, 50

WANG, J.; LIU, Z.; WU, Y.; AND YUAN, J., Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1290–1297. IEEE. 56

WANG, L. AND KONIUSZ, P., Hallucinating Statistical Moment and Subspace Descriptors for Action Recognition. *ACM Multimedia*, (2021). 72

WANG, L.; KONIUSZ, P.; AND HUYNH, D. Q., Hallucinating IDT Descriptors and I3D Optical Flow Features for Action Recognition with CNNs. *International Conference on Computer Vision*, (2019). 72

WANG, M. AND DENG, W., Deep visual domain adaptation: A survey. *Neurocomputing*, 312 (2018), 135–153. 7

WANG, P.; LI, W.; OGUNBONA, P.; WAN, J.; AND ESCALERA, S., RGB-D-based human motion recognition with deep learning: A survey. *Computer Vision and Image Understanding*, 171 (2018), 118–139. 9

WANG, Y.-X. AND HEBERT, M., Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision*, 616–634. Springer. 18, 19, 23, 35, 36

WEN, T.-H.; VANDYKE, D.; MRKSIC, N.; GASIC, M.; ROJAS-BARAHONA, L. M.; SU, P.-H.; ULTES, S.; AND YOUNG, S., A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, (2016). 68

WEST, J.; VENTURA, D.; AND WARNICK, S., Spring research presentation: A theoretical foundation for inductive transfer. *Brigham Young University, College of Physical and Mathematical Sciences*, 1, 08 (2007). 16

WOODWORTH, R. S. AND THORNDIKE, E., The influence of improvement in one mental function upon the efficiency of other functions.(I). *Psychological review*, 8, 3 (1901), 247. 16, 73

WU, J.; HU, Z.; AND MOONEY, R. J., Joint image captioning and question answering. *arXiv preprint arXiv:1805.08389*, (2018). 69

XIA, L.; CHEN, C.-C.; AND AGGARWAL, J. K., View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 20–27. IEEE. 9, 55, 64

XU, F.; YU, J.; AND XIA, R., Instance-based domain adaptation via multiclustering logistic approximation. *IEEE Intelligent Systems*, 33, 1 (2018), 78–88. 7

XU, K.; BA, J.; KIROS, R.; CHO, K.; COURVILLE, A.; SALAKHUDINOV, R.; ZEMEL, R.; AND BENGIO, Y., Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057. 11, 68

YACOOB, Y. AND BLACK, M. J., Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73, 2 (1999), 232–247. 56

YANG, X. AND TIAN, Y., Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25, 1 (2014), 2–11. 56

YEH, Y.-R.; HUANG, C.-H.; AND WANG, Y.-C. F., Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *IEEE Transactions on Image Processing*, 23, 5 (2014), 2009–2018. 18, 19, 36

YUN, K.; HONORIO, J.; CHATTOPADHYAY, D.; BERG, T. L.; AND SAMARAS, D., Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 28–35. IEEE. 9, 55, 63, 64

ZATSIORSKY, V. M. AND ZACIORSKIJ, V. M., 2002. *Kinetics of human motion*. Human Kinetics. 56

ZEILER, M. D. AND FERGUS, R., Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer. 55

ZENG, K.-H.; CHEN, T.-H.; CHUANG, C.-Y.; LIAO, Y.-H.; NIEBLES, J. C.; AND SUN, M., Leveraging video descriptions to learn video question answering. In *Thirty-First AAAI Conference on Artificial Intelligence*. 11, 68

ZHANG, H. AND KONIUSZ, P., Power Normalizing Second-order Similarity Network for Few-shot Learning. *Winter Conference on Applications of Computer Vision*, (2019). 68

ZHANG, H.; KONIUSZ, P.; JIAN, S.; LI, H.; AND TORR, P. H. S., Rethinking Class Relations: Absolute-Relative Supervised and Unsupervised Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9432–9441. 68, 72

ZHANG, H.; ZHANG, L.; QI, X.; LI, H.; TORR, P.; AND KONIUSZ, P., Few-shot action recognition with permutation-invariant attention. In *The European Conference on Computer Vision*. 72

ZHANG, J.; LI, W.; OGUNBONA, P.; AND XU, D., Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Computing Surveys (CSUR)*, 52, 1 (2019), 1–38. 7

ZHANG, R.; TAS, Y.; AND KONIUSZ, P., Artwork Identification from Wearable Camera Images for Enhancing Experience of Museum Audiences. In *Museums and the Web*. 68

ZHANG, S.; LUO, D.; WANG, L.; AND KONIUSZ, P., Few-Shot Object Detection by Second-order Pooling. *Asian Conference on Computer Vision*, (2020). 68

ZHOU, B.; LAPEDRIZA, A.; XIAO, J.; TORRALBA, A.; AND OLIVA, A., Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, 487–495. 32

ZHU, H. AND KONIUSZ, P., REFINE: Random RangE FInder for Network Embedding. In *ACM International Conference on Information and Knowledge Management*. doi:0.1145/3459637.3482168. 72

ZHU, H. AND KONIUSZ, P., Simple Spectral Graph Convolution. In *International Conference on Learning Representations*. 72

ZHU, H.; SUN, K.; AND KONIUSZ, P., Contrastive Laplacian Eigenmaps. In *Conference on Neural Information Processing Systems*. 72

ZHU, W.; LAN, C.; XING, J.; ZENG, W.; LI, Y.; SHEN, L.; AND XIE, X., Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Thirtieth AAAI Conference on Artificial Intelligence*. 54, 63, 64

Zhu, Y.; Chen, W.; and Guo, G., Fusing spatiotemporal features and joints for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 486–491. 64