UNIVERSITY OF CALIFORNIA
RIVERSIDE

# Source Free Domain Adaptive Machine Learning and Unlearning

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Sk Miraj Ahmed

June 2024

Dissertation Committee:

Dr. Amit K. Roy-Chowdhury, Chairperson
Dr. Samet Oymak
Dr. Basak Guler

The Dissertation of Sk Miraj Ahmed is approved:

---

---

---

Committee Chairperson

University of California, Riverside

## Acknowledgments

First and foremost, I would like to express my deepest gratitude to my PhD advisor, Dr. Amit K. Roy-Chowdhury, for his unwavering support and invaluable guidance throughout my research journey. I am profoundly grateful for the countless hours we have spent engaging in insightful discussions about my work, career choices, and life in general. Amit, your mentorship has been instrumental in shaping my approach to research. Through your guidance, I have learned to tackle problems at a high level and identify novel and relevant research questions, which is often the most challenging aspect of research. Moreover, I have gained invaluable skills in efficiently presenting my work, ensuring clarity and accessibility to readers from diverse backgrounds. Your trust and encouragement have empowered me with the freedom to explore and think at a fundamental level, ultimately contributing to my personal and academic growth. Additionally, I am grateful for your unwavering motivation during moments of doubt and for lifting my spirits when I needed it the most. Thank you, Amit, for your outstanding mentoring and constant support; I am truly fortunate to have had you as my advisor. I extend my heartfelt gratitude to my PhD committee members, Samet Oymak and Basak Guler, for their invaluable feedback and guidance throughout the process of completing this dissertation.

Samet, I am immensely thankful for your insightful contributions to my research. Your exceptional teaching abilities have been a source of deep motivation for me. Your clear explanations and vivid illustrations of complex equations have elevated my understanding of probability and deep learning to a profound level, making my research journey all the more enjoyable.

Basak, it was a privilege to collaborate with you on my latest work. Your keen insights enabled me to identify the loopholes in the problem and encouraged me to approach it from a fresh perspective. I am optimistic that together, we can address these concerns in future endeavors. Working with both of you has been an immense pleasure, and I am truly grateful for the wisdom and guidance you have provided throughout my academic journey.

I consider myself incredibly fortunate to have had the opportunity to meet and be mentored by some truly remarkable individuals during my internships at Mitsubishi Electric Research Labs. I am deeply grateful to all of them, and I would like to extend special thanks to Suhas Lohit, Kuan-Chaun Peng, and Michael Jones. Their guidance, expertise, and mentorship have been invaluable in shaping my professional development and enhancing my skills in the field.

I am also grateful to Dr. Kunal Narayan Chaudhury, under whom I pursued my master's degree at IISc Bangalore, India. Dr. Chaudhury's profound knowledge and mathematical insights have been instrumental in inspiring my academic pursuits. His exceptional mentoring, characterized by a commitment to solving problems at a fundamental level with mathematical rigor, has deeply influenced my academic approach and contributed to shaping my academic personality. I am immensely thankful for his guidance, encouragement, and unwavering support throughout my academic journey.

I have had the privilege of being surrounded by an exceptional group of fellow students at UCR. They are not only funny and intelligent but also incredibly caring individuals who have made my PhD experience truly memorable. I would like to extend my heartfelt thanks to those with whom I have worked closely: Dripta, Sujoy, Rameswar, Cody,

Niloy, Arindam, Rohit, Chang, Sayak, and Shazid. Although we did not publish together, I am grateful for the many insightful discussions I had with Abhishek, Sudipta, and Aakash. Thank you all for enriching my journey.

Outside of research, I am also grateful to Maksud, Dripta, Sujoy, Sourya, Tayafa, Rohit, Arindam, Sarosij, Sayak, Disha, Pritha, and Udita. Together, we have taken countless trips and enjoyed evenings cooking delicious Indian food and relaxing with Netflix. These moments provided the essential balance I needed to keep my mind fresh for research. I am grateful for their invaluable support and contributions to making this journey enjoyable and fulfilling.

I would like to extend my deepest appreciation to my high school friends who have been steadfast companions: Souvik, Hirak, Abhishek, Kunal, Preetam, Pritam, Kaustav, Archishman, and Subhasis. Thank you, guys. Additionally, from my undergraduate years: Shouvik, Shibu, Abhishek, Niladri, and Kaushik, to name a few. There are many other equally important friends from those days, but I can only list so many in this thesis. Thank you all for being a constant source of support and encouragement throughout my academic journey.

A special thanks to Shouvik and Shibu, who are not only funny and incredibly intelligent but have also played a phenomenal role in shaping my understanding of some of the toughest subjects. We can talk for hours on a deep philosophical level and never get bored. Thank you so much, guys, for igniting my passion for research.

During my master's at IISc, I was fortunate to meet some incredibly smart and funny friends. To name a few: Debasish, Niladri, Soubhik, Rajat, Taskar, Indra, Masiur,

Premjit, Gopal, and Sayantan, whom I refer to as Da (brother). They made my time there truly worthwhile. I also had wonderful peers like Soumyajit, Aritra, Soham, Aratrik, Mousumi, Debaleena, and Saptarshi, with whom I spent some of my most enjoyable days. Thank you all, guys.

To my late parents, I lost my mother when I was four, and my father raised me on his own. Thank you so much, Abba, for your support, guidance, and unconditional love. May your souls rest in peace.

I deeply admire my elder brother Mustaque, without whose guidance I may not be in my current position. Thank you for guiding me from the very start of my academic journey and being my friend, philosopher, and guide. My other brother, Manjur, thank you so much for taking care of my needs when I was at home. Ruxana, my sister, you were always there to compensate for the loss of our mother.

My deepest gratitude also goes to my sisters-in-law, Nafisa and Sabina, who constantly fed me with good food and took care of me.

Lastly, I want to thank my recent better half, Sameeha. You have been a constant support for me over the past two years. In this foreign country, you alleviated my loneliness and were always there when I needed you most. Thank you for always uplifting my spirits. Be there always as you are.

**Acknowledgment of previously published materials.** The text of this dissertation, in part or in full, is a reprint of the material as appeared in four previously published/submitted papers for which I am the lead author. The co-author Amit K. Roy-Chowdhury, listed in all five publications, directed and supervised the research which forms

the basis for this dissertation. The papers are as follows:

1. Ahmed, S. M., Lejbølle, A. R. , Panda, R., and Roy-Chowdhury, A. K. (2020). Camera On-boarding for Person Re-identification using Hypothesis Transfer Learning. CVPR 2020.

2. Ahmed, S. M., Raychaudhuri, D. S., Paul, S., Oymak, S., and Roy-Chowdhury, A. K. (2021). Unsupervised Multi-source Domain Adaptation Without Access to Source Data. CVPR 2021.

3. Ahmed, S. M., Lohit, S., Peng, K.C., Jones, M.J. and Roy-Chowdhury, A. K.(2022). Cross-Modal Knowledge Transfer Without Task-Relevant Source Data. ECCV 2022.

4. Ahmed, S. M, Niloy, F. F., Raychaudhuri, D. S., Chang, X., Oymak, S. and Roy-Chowdhury, A. K. (2024). CONTRAST: Continual Multi-source Adaptation to Dynamic Distributions. Under Review.

5. Ahmed, S. M, Basaran, U., Raychaudhuri, D. S., Dutta, A., Niloy, F. F., Kundu, R., Guler, B. and Roy-Chowdhury, A. K. (2024). Source-Free Machine Unlearning. Under Review.

To my late parents, for all their support. Also, to Almighty Allah for the countless blessings.

# ABSTRACT OF THE DISSERTATION

Source Free Domain Adaptive Machine Learning and Unlearning

by

Sk Miraj Ahmed

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, June 2024
Dr. Amit K. Roy-Chowdhury, Chairperson

Deep neural networks have demonstrated remarkable efficacy across a wide range of tasks, yet they face a significant limitation in their ability to adapt to distributional shifts. In contrast, humans possess inherent adaptability, effortlessly adjusting to changes in data distributions and modifying task strategies to accommodate environmental variations without any external supervision. This adaptability has inspired the field of unsupervised domain adaptation (UDA). Most existing UDA methods rely on access to the source data on which the model was initially trained during the adaptation phase. However, a more practical scenario involves situations where only the trained model is available, rather than the source data. This approach mirrors human learning more closely, as humans do not use previous data directly; instead, their brains are pre-trained on source data and apply this knowledge to new situations.

Based on this observation, the field of source-free domain adaptation has emerged. In source-free domain adaptation, only the pre-trained source models and new target data are used during adaptation to a new environment. This dissertation encompasses five sig-

nificant contributions to this emerging field. First, we explore a scenario where we leverage multiple pretrained source models, each trained on different domains, during the adaptation phase without using source data. We develop an algorithm that effectively combines these models such that the most correlated source model with respect to the target data receives the highest weight, while the least correlated one receives the lowest weight. This approach ensures maximum knowledge transfer from all sources, resulting in final adaptation performance that surpasses any individual source model. This algorithm is designed for a static target distribution, where all target data are available during adaptation and do not change over time.

Expanding on this approach, we next consider a scenario where the target data is time-varying and arrives in a streaming fashion. This dynamic setting requires continual adaptation as new data becomes available, presenting unique challenges compared to the static scenario. We then explore two applications of these approaches:(i) adapting to target data from a modality different than the sources, and (ii) adding a new source model to the ensemble of source models with reliance on only a few labeled target data points. Finally, we focus on another emerging field of research called unlearning, where the trained model must forget certain data it has seen during training to meet user privacy concerns. Unlike existing approaches that require access to all training data during the unlearning process, we address this in a source-free manner, needing only the data to be forgotten during the unlearning procedure.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Deep neural networks have demonstrated remarkable proficiency in a wide range of computer vision tasks. However, they consistently struggle with adapting to shifts in visual distributions. In contrast, human recognition remains robust in the face of such shifts, allowing us to read text in a new font or recognize new objects of the same class in entirely unfamiliar environments. Instilling this kind of robustness to distributional shifts in deep models is crucial for their practical application.

A substantial body of existing work aims to address this issue by transferring knowledge from labeled source datasets to unlabeled target datasets, which may exhibit either static or continuously evolving distributional changes. In the static scenario, a notable limitation of many current transfer learning methods is their reliance on a transductive approach, where source data is required for knowledge transfer. In real-world settings, source data may not be accessible due to various concerns like privacy, security, and storage constraints [102]. Furthermore, in numerous application scenarios, multiple labeled source

domains are available, and efficiently leveraging them can yield more optimal solutions than simply relying on a single source [229].

Our research explores these two challenges concurrently by developing efficient algorithms that can function with multiple source domains and without access to source data. These algorithms are designed to work with various data scenarios, including those with few labeled samples, completely unlabeled data, or data from entirely different modalities or data from a dynamic distribution, across computer vision tasks such as classification, semantic segmentation, and object detection. We also explore a scenario where we forget some data using Machine Unlearning (MU) in a setting where no source data is available.

In Chapter 2, we focus on the problem of multiple source UDA with no access to the source data and make the following contributions: We propose a novel UDA algorithm, termed Data frEe multi-sourCe unsupervISed domain adaptatiON (DECISION), which operates without requiring access to the source data. To solve the problem, we deploy *Information Maximization (IM)* loss [102] on the weighted combination of target soft labels from all the source models. This approach enhances the confidence and diversity of predictions across all classes. Additionally, we utilize a pseudo-label strategy inspired by the deep cluster method [16], alongside the IM loss, to minimize noisy cluster assignment of the features. The overall optimization process jointly adapts the feature encoders from the sources and the corresponding source weights, resulting in the final target model. Our algorithm automatically identifies the optimal blend of source models to generate the target model by optimizing a carefully designed unsupervised loss. Under intuitive assumptions, we establish theoretical guarantees on the performance of the target model, showing that

it is consistently at least as good as deploying the single best source model, thus minimizing negative transfer. We validate our claims with extensive numerical experiments, demonstrating the practical benefits of our approach.

In Chapter 3, we introduce a framework for multi-source adaptation to dynamic distribution shifts from streaming test data without access to the source data. Our develop an algorithm, CONTRAST, that can merge the source models using appropriate combination weights during test time, enabling it to perform as well as, or even better than, the best-performing source model. Additionally, our framework effectively mitigates catastrophic forgetting when faced with long-term, fluctuating test distributions. We provide theoretical insights into CONTRAST, illustrating how it addresses domain shift by optimally combining source models and prioritizing updates to the model least prone to forgetting. To demonstrate the real-world advantages of our methodology, we perform experiments on a diverse range of benchmark datasets.

In Chapter 4, we formulate a novel problem for knowledge transfer from a model trained for a source modality to a different target modality without any access to task-relevant source data and when the target data is unlabeled. To bridge the gap between modalities, we propose a novel framework, SOCKET, for cross-modal knowledge transfer without access to source data by using an external task-irrelevant paired dataset and by matching the moments obtained from the normalization layers in the source models with the moments computed on the unlabeled target data. Extensive experiments on multiple datasets, both for knowledge transfer from RGB to depth and from RGB to IR, and both for single-source and multi-source cases, show that SOCKET is effective in reducing the modal-

ity gap in the feature space, producing significantly better performance, with improvements of up to 12% in some cases, over existing source-free domain adaptation baselines that do not account for the modality difference between the source and target modalities. We also show empirically that for the datasets of interest, the problem of knowledge transfer between modalities like RGB and depth is harder than domain shifts within the same modality, such as sensor changes and viewpoint shifts, considered previously in the literature.

In Chapter 5, we address the problem of swiftly on-boarding new camera(s) into an existing person re-identification network without having access to the source camera data and relying on only a small amount of labeled target data in the transient phase, i.e., after adding the new cameras. Towards solving this problem, we make the following contributions. We propose a robust and efficient multiple metric hypothesis transfer learning algorithm to adapt a newly introduced camera to an existing person re-id framework without access to the source data. We theoretically analyze the properties of our algorithm, demonstrating that it minimizes the risk of negative transfer and performs closely to the fully supervised case even with a small amount of labeled data. Additionally, we conduct rigorous experiments on multiple benchmark datasets to show the effectiveness of our proposed approach over existing alternatives.

In Chapter 6, we propose a Machine Unlearning (MU) algorithm in a source-free setting, where unlearning is performed without access to the original training data. We only have the source model and the data to be forgotten during the unlearning process. Our main contributions in this work can be summarized as follows: To the best of our knowledge, this work proposes the first source-free unlearning method for linear classifiers

that can effectively forget random instances of data from all classes while also providing robust theoretical guarantees regarding data removal and privacy. Since we cannot compute the Hessian directly from the remaining data, we propose two novel methods for estimating the Hessian: (i) for any general convex loss, and (ii) for the specialized case of quadratic mean squared error (MSE) loss. The first approach can approximately unlearn (i.e., with bounded error) for any convex loss functions, while the second is tailored specifically for quadratic loss and enables exact unlearning. We provide theoretical guarantees for our unlearning mechanism through extensive proofs and validate our claims with experiments and ablations on linear classifiers using multiple benchmark datasets.

Finally, in Chapter 7 we summarize the findings of this thesis and discuss possible extensions and future avenues for further research.

# Chapter 2

# Multi-Source Free UDA

## 2.1 Introduction

Deep neural networks have achieved proficiency in a multiple array of vision tasks [58, 109, 83, 148], however, these models have consistently fallen short in adapting to visual distributional shifts [111]. Human recognition, on the other hand, is robust to such shifts, such as reading text in a new font or recognizing objects in unseen environments. Imparting such robustness towards distributional shifts to deep models is fundamental in applying these models to practical scenarios.

Unsupervised domain adaptation (UDA) [10, 153] seeks to bridge this performance gap due to domain shift via adaptation of the model on small amounts of unsupervised data from the target domain. The majority of current approaches [44, 65] optimize a two-fold objective: (i) minimize the empirical risk on the source data, (ii) make the target and source features indistinguishable from each other. Minimizing distribution divergence between domains by matching the distribution statistical moments at different orders have

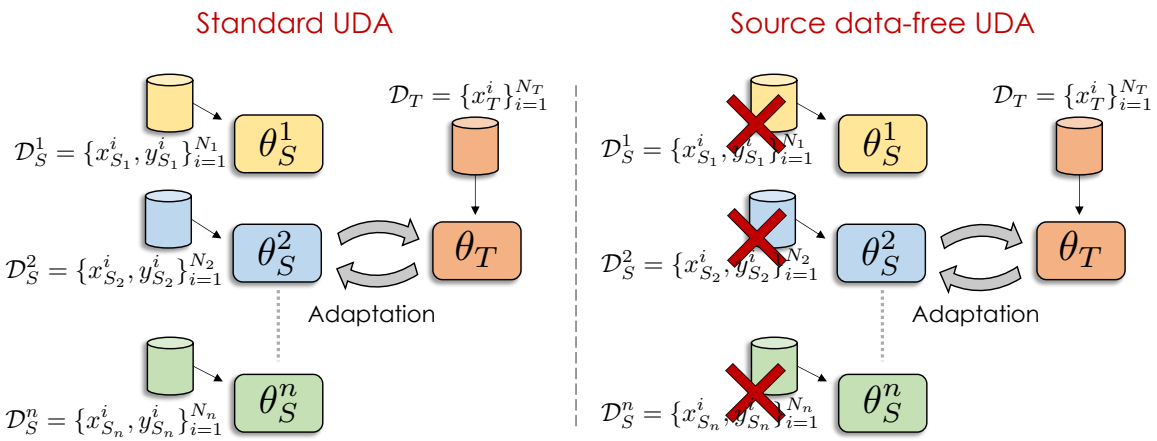



Figure 2.1: **Problem setup.** Standard unsupervised multi-source domain adaptation (UDA) utilizes the source data, along with the models trained on the source, to perform adaptation on a target domain. In contrast, we introduce a setting which adapts multiple models without requiring access to the source data.

also been explored extensively [165, 139].

A shortcoming of all the above approaches is the transductive scenario in which they operate, i.e., the source data is required for adaptation purposes. In a real-world setting, source data may not be available for a variety of reasons. Privacy and security are the primary concern, with the data possibly containing sensitive information. Another crucial reason is storage issues, i.e., source datasets may contain videos or high-resolution images and it might not be practical to transfer or store on different platforms. Consequently, it is imperative to develop unsupervised adaptation approaches which can adapt the source models to the target domain *without access to the source data*.

Recent works [97, 102] attempt this by adapting a single source model to a target domain without accessing the source data. However, an underlying assumption of these

methods is that the most correlated source model is provided by an oracle for adaptation purposes. A more challenging and practical scenario entails adaptation from a *bag of source models* - each of these source domains are correlated to the target by different amounts and adaptation involves not only incorporating the combined prior knowledge from multiple models, but simultaneously preventing the possibility of negative transfer. In this chapter, we introduce the problem of unsupervised *multi-source adaptation without access to source data*. We develop an algorithm based on the principles of pseudo-labeling and information maximization and provide intuitive theoretical insights to show that our framework guarantees performance better than the best available source and minimize the effect of negative transfer.

To solve this problem of multiple source model adaptation without accessing the source data, we deploy *Information Maximization (IM)* loss [102] on the weighted combination of target soft labels from all the source models. We also use the pseudo-label strategy inspired from deep cluster method [16], along with the IM loss to minimize noisy cluster assignment of the features. The overall optimization jointly adapts the feature encoders from sources as well as the corresponding source weights, combining which the target model is obtained.

**Main Contributions.** We address the problem of multiple source UDA, with no access to the source data. Towards solving the problem, we make the following contributions:

- We propose a novel UDA algorithm which operates without requiring access to the source data. We term it as Data frEe multi-sourCe unsupervISed domain adaptatiON

(DECISION). Our algorithm automatically identifies the optimal blend of source models to generate the target model by optimizing a carefully designed unsupervised loss.

• Under intuitive assumptions, we establish theoretical guarantees on the performance of the target model which shows that it is consistently at least as good as deploying the single best source model, thus, minimizing negative transfer.

• We validate our claim by extensive numerical experiments, demonstrating the practical benefits of our approach.

## 2.2 Related works

In this section we present a brief overview of the literature in the area of unsupervised domain adaptation in both the single and multiple sources scenario, as well as the closely related setting of hypothesis transfer learning.

**Unsupervised domain adaptation.** UDA methods have been used for a variety of tasks, including image classification [177], semantic segmentation [137] and object detection [67]. Besides the feature space adaptation methods based on the paradigms of moment matching [165, 139] and adversarial learning [44, 177], recent works have explored pixel space adaptation via image translation [65]. All existing UDA methods require access to labeled source data, which may not be available in many applications.

**Hypothesis transfer learning.** Similar to our objective, hypothesis transfer learning (HTL) [161, 142, 4] aims to transfer learnt source hypotheses to a target domain without access to source data. However, data is assumed to be labeled in the target domain in contrast to our scenario, limiting its applicability to real-world settings. Recently, [97, 102]

extend the standard HTL setting to unsupervised target data (U-HTL) by adapting single source hypotheses via pseudo-labeling. Our thesis takes this one step further by introducing multiple source models, which may or may not be positively correlated with the target domain.

**Multi-source domain adaptation.** Multi-source domain adaptation (MSDA) extends the standard UDA setting by incorporating knowledge from multiple source models. Latent space transformation methods [231] aim to align the features of different domains by optimizing a discrepancy measure or an adversarial loss. Discrepancy based methods seek to align the domains by minimizing measures such as maximum mean discrepancy [56, 231] and Rényi-divergence [64]. Adversarial methods aim to make features from multiple domains indistinguishable to a domain discriminator by optimizing GAN loss [205], $\mathcal{H}-$divergence [229] and Wasserstein distance [185, 101]. Domain generative methods [152, 106] use some form of domain translation, such as the CycleGAN [236], to perform adaptation at the pixel level. All these methods assume access to the source data during adaptation.

## 2.3 Methodology

**Problem setting.** We address the problem of jointly adapting multiple models, trained on a variety of domains, to a new target domain with access to only samples without annotations from the target. In this work, we will be considering the adaptation of classification models with $K$ categories and the input space being $\mathcal{X}$. Formally, let us consider we have a set of source models $\{\theta_S^j\}_{j=1}^n$, where the $j^{th}$ model $\theta_S^j : \mathcal{X} \rightarrow \mathbb{R}^K$, is a classification model learned using the source dataset $\mathcal{D}_S^j = \{x_{S_j}^i, y_{S_j}^i\}_{i=1}^{N_j}$, with $N_j$ data points, where

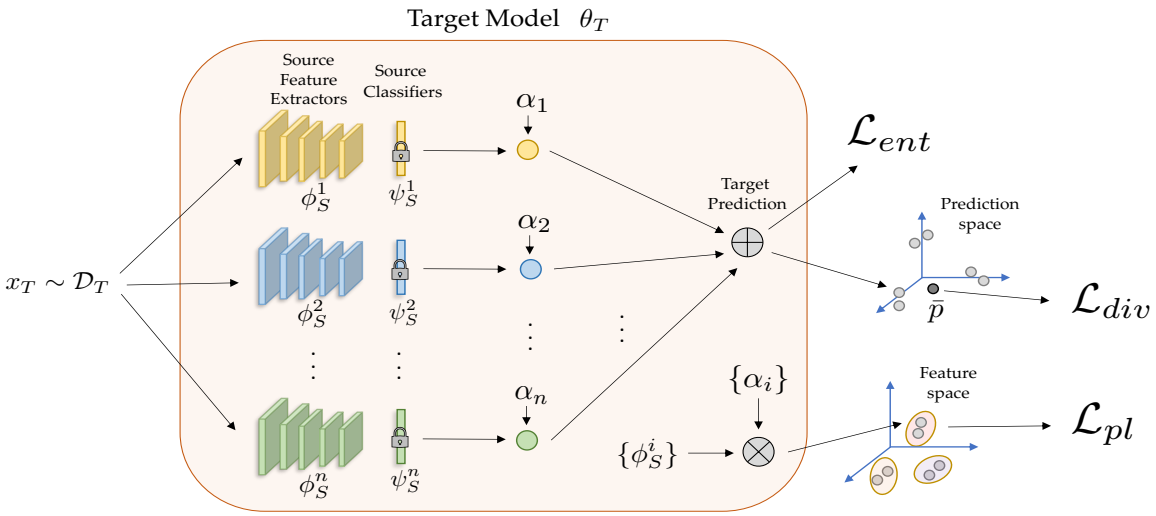



Figure 2.2: **Overall framework of our approach:** We freeze the final classification layers of all the sources and jointly optimize for the source feature encoders along with it's corresponding weights to get the target predictor by combining those.

$x^i_{S_j}$ and $y^i_{S_j}$ denote the $i$-th source image and the corresponding label respectively. Now, given a target unlabeled dataset $\mathcal{D}_T = \{x^i_T\}_{i=1}^{N_T}$, the problem is to learn a classification model $\theta_T : \mathcal{X} \to \mathbb{R}^K$, using only the learned source models, without any access to the source datasets. Note that this is different from multi-source domain adaptation methods in literature, which also utilize the source data while learning the target model $\theta_T$.

**Overall Framework.** We can decompose each of the source models into two modules: the feature extractor $\phi^i_S : \mathcal{X} \to \mathbb{R}^{d_i}$ and the classifier $\psi^i_S : \mathbb{R}^{d_i} \to \mathbb{R}^K$. Here, $d_i$ refers to the feature dimension of the $i$-th model while $K$ refers to the number of categories. We aim to estimate the target model $\theta_T$ by combining knowledge only from the given source models in a manner that automatically rejects poor source models, i.e., those which are

irrelevant for the target domain. At the core of our framework lies a model aggregation scheme [113, 64], wherein we learn a set of weights $\{\alpha_i\}_{i=1}^n$ corresponding to each of the source models, such that, $\alpha_k \geq 0$ and $\sum_{k=1}^n \alpha_k = 1$. These weights represent a probability mass function over the source domains, with a higher value implying higher transferability from that particular domain, and are used to combine the source hypotheses accordingly. However, unlike previous works, we jointly adapt each individual model and simultaneously learn these weights by utilizing solely the unlabeled target instances. In what follows, we describe our training strategy used to achieve this in detail.

### 2.3.1 Weighted Information Maximization

As we do not have access to the labeled source or target data, we propose to fix the source classifiers,$\{\psi_S^i\}_{i=1}^n$, since it contains the class distribution information of the source domain and adapt solely the feature maps $\{\phi_S^i\}_{i=1}^n$ via the principle of information maximization [13, 85, 129, 102]. Our motivation behind the adaptation process stems from the *cluster assumption* [19] in semi-supervised learning, which hypothesizes that the discriminative model's decision boundaries should be located in regions of the input space which are not densely populated. To achieve this, we minimize a conditional entropy term (i.e., for a given input example) [51] as follows:

$$\mathcal{L}_{\text{ent}} = -\mathbb{E}_{x_T \in \mathcal{D}_T} \Big[ \sum_{j=1}^{K} \delta_j(\theta_T(x_T)) \log(\delta_j(\theta_T(x_T))) \Big] \tag{2.1}$$

where $\theta_T(x_T) = \sum_{j=1}^n \alpha_j \theta_S^j(x_T)$, and $\delta(\cdot)$ denotes the softmax operation with $\delta_j(v) = \frac{\exp(v_j)}{\sum_{i=1}^K \exp(v_i)}$ for $v \in \mathbb{R}^K$. Intuitively, if a source $\theta_S^j$ has good transferability on the target and consequently, has smaller value of the conditional entropy, optimizing the term (2.1)

over $\left\{\theta_S^j, \alpha_j\right\}$, will result in higher value of $\alpha_j$ than rest of the weights. While entropy minimization effectively captures the cluster assumption when training with partial labels, in an unsupervised setting, it may lead to degenerate solutions, such as, always predicting a single class in an attempt to minimize conditional entropy. To control such degenerate solutions, we incorporate the idea of class diversity: configurations in which class labels are assigned evenly across the dataset are preferred. A simple way to encode our preference towards class balance is to maximize the entropy of the empirical label distribution [13] as follows,

$$\mathcal{L}_{\text{div}} = \sum_{j=1}^{K} -\bar{p}_j \log \bar{p}_j \tag{2.2}$$

where $\bar{p} = \mathbb{E}_{x_T \in \mathcal{D}_T}[\delta(\theta_T(x_T))]$. Combining the terms (2.1) and (2.2), we arrive at,

$$\mathcal{L}_{\text{IM}} = \mathcal{L}_{\text{div}} - \mathcal{L}_{\text{ent}} \tag{2.3}$$

which is the empirical estimate of the mutual information between the target data and the labels under the aggregate model $\theta_T$. Although maximizing this loss makes the predictions on the target data more confident and globally diverse, it may sometime still fail to restrict erroneous label assignment. Inspired by [102], we propose a pseudo-labeling strategy in an effort to contain this mislabeling.

### 2.3.2 Weighted Pseudo-labeling

As a result of domain shift, information maximization may result in some instances being clubbed with the wrong class cluster. These wrong predictions get reinforced over the course of training and lead to a phenomenon termed as *confirmation bias* [170]. Aiming to contain this effect we adopt a self-supervised clustering strategy [102] inspired from the

DeepCluster technique [16]. First, we calculate the cluster centroids induced by each source model for the whole target dataset as follows,

$$\mu_{k_j}^{(0)} = \frac{\sum_{x_T \in \mathcal{D}_T} \delta_k(\hat{\theta}_S^j(x_T))\hat{\phi}_S^j(x_T)}{\sum_{x_T \in \mathcal{D}_T} \delta_k(\hat{\theta}_S^j(x_T))} \tag{2.4}$$

where the cluster centroid of class $k$ obtained from source $j$ at iteration $i$ is denoted as $\mu_{k_j}^{(i)}$, and $\hat{\theta}_S^j = (\psi_S^j \circ \hat{\phi}_S^j)$ denotes the source from the previous iteration. These source-specific centroids are combined in accordance to the current aggregation weights on each source model as follows,

$$\mu_k^{(0)} = \sum_{j=1}^{n} \alpha_j \mu_{k_j}^{(0)} \tag{2.5}$$

Next, we compute the pseudo-label of each sample by assigning it to its nearest cluster centroid in the feature space,

$$\hat{y}_T^{(0)} = \arg\min_k \|\hat{\theta}_T(x_T) - \mu_k^{(0)}\|_2^2 \tag{2.6}$$

We reiterate this process to get the updated centroids and pseudo-labels as follows,

$$\mu_{k_j}^{(1)} = \frac{\sum_{x_T \in \mathcal{D}_T} \mathbb{1}\{\hat{y}_T^{(0)} = k\}\hat{\phi}_S^j(x_T)}{\sum_{x_T \in \mathcal{D}_T} \mathbb{1}(\hat{y_{t_0}} = k)} \tag{2.7}$$

$$\mu_k^{(1)} = \sum_{j=1}^{n} \alpha_j \mu_{k_j}^{(1)} \tag{2.8}$$

$$\hat{y}_T^{(1)} = \arg\min_k \|\hat{\theta}_T(x_T) - \mu_k^{(1)}\|_2^2 \tag{2.9}$$

where $\mathbb{1}(\cdot)$ is an indicator function which gives a value of 1 when the argument is true. While this alternating process of computing cluster centroids and pseudo-labeling can be repeated multiple times to get stationary pseudo-labels, one round is sufficient for all practical purposes. We then obtain the cross-entropy loss w.r.t. these pseudo-labels as follows:

$$\mathcal{L}_{\text{pl}}(Q_T, \theta_T) = -\mathbb{E}_{x_T \in \mathcal{D}_T} \sum_{k=1}^{K} \mathbb{1}\{\hat{y}_T = k\} \log \delta_k(\theta_T(x_T)). \tag{2.10}$$

Note that the pseudo-labels are updated regularly after a certain number of iterations as discussed in Section 2.5.

### 2.3.3 Optimization

In summary, given $n$ source hypothesis $\{\theta_S^j\}_{j=1}^n = \{\psi_S^j \circ \phi_S^j\}_{j=1}^n$ and target data $\mathcal{D}_T = \{x_T^i\}_{i=1}^{n_T}$, we fix the classifier from each of the sources and optimize over the parameters of $\{\phi_S^j\}_{j=1}^n$ and the aggregation weights $\{\alpha_j\}_{j=1}^n$. The final objective is given by,

$$\mathcal{L}_{tot} = \mathcal{L}_{\text{ent}} - \mathcal{L}_{\text{div}} + \lambda \mathcal{L}_{\text{pl}} \tag{2.11}$$

The above objective is used to solve the following optimization problem,

$$\begin{aligned} \underset{\{\phi_S^j\}_{j=1}^n, \{\alpha_j\}_{j=1}^n}{\text{minimize}} \quad & \mathcal{L}_{tot} \\ \text{subject to} \quad & \alpha_j \geq 0, \forall j \in \{1, 2, \ldots, n\}, \\ & \sum_{j=1}^{n} \alpha_j = 1 \end{aligned} \tag{2.12}$$

Once we obtain the optimal set of $\phi_S^{j*}$ and $\alpha_j^*$, the optimal target hypothesis is computed as $\theta_T = \sum_{j=1}^n \alpha_j^* (\psi_S^j \circ \phi_S^{j*})$. To solve the optimization (2.12) we follow the steps of Algorithm (1) stated below.

## 2.4 Theoretical Insights

**Theoretical motivation behind our approach.** Our algorithm aims to find the optimal weights $\{\alpha_j\}_{j=1}^n$ for each source and takes a convex combination of the source predictors to obtain the target predictor. Here, we shall show that under intuitive assumptions on the source and target distributions, there exists a simple choice of target predictor, which can

**Algorithm 1** Algorithm to Solve Eq. 2.12

**Input:** Trained source models $\{\theta_S^j\}_{j=1}^n = \{\psi_S^j \circ \phi_S^j\}_{j=1}^n$, unlabeled target data $\{x_T^i\}_{i=1}^{N_T}$, weight parameters $\{\alpha_j\}_{j=1}^n$, max number of epochs $E$, regularization parameter $\lambda$, number of batches $B$

**Output:** Optimal feature encoders $\{\phi_S^{j*}\}_{j=1}^n$, optimal source weights $\{\alpha_j^*\}_{j=1}^n$

**Initialization**: Freeze final classification layers $\{\psi_S^j\}_{j=1}^n$, set $\alpha_j = 1$ for all $j$

**for** *epoch* $= 1$ **to** $E$ **do**

    Calculate pseudo-labels from equation (2.6)

    Calculate the mean embedding $\bar{p}$ from equation (2.2)

    **for** *iteration* $= 1$ **to** $B$ **do**

        Sample a mini batch from target and pass it through each of the source models

        Calculate all the losses from equation (2.1), (2.2) and (2.10)

        Calculate total loss from equation (2.3)

        Update the parameters in $\{\phi_S^j\}_{j=1}^n$ and $\{\alpha_j\}_{j=1}^n$ from optimization (2.12)

        Make $\alpha$ positive by setting $\alpha_j = 1/(1 + e^{-\alpha_j})$

        Normalize $\alpha$ by setting $\alpha_j = \alpha_j / \sum_{i=1}^n \alpha_i$

    **end for**

**end for**

perform better than or equal to the best source model being applied directly on the target data. Formally, let $L$ be a loss function which maps the pair of model-predicted label and the ground-truth label to a scalar. Denote the expected loss over $k$-th source distribution $Q_S^k$ using the source predictor $\theta$ via $\mathcal{L}(Q_S^k, \theta) = \mathbb{E}_x[L(\theta(x), y)] = \int_x L(\theta(x), y) Q_S^k(x) dx$. Now let $\theta_S^k$ be the optimal source predictor given by $\theta_S^k = \arg\min_\theta \mathcal{L}(Q_S^k, \theta)\ \forall\ 1 \leq k \leq n$. Let us also assume that the target distribution is in the span of source distributions. We formalize this by expressing the target distribution as an affine combination of source distributions i.e., $Q_T(x) = \sum_{k=1}^n \lambda_k Q_S^k(x) : \lambda_k \geq 0, \sum_{k=1}^n \lambda_k = 1$. Under this assumption, if we express our target predictor as $\theta_T(x) = \sum_{k=1}^n \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^n \lambda_j Q_S^j(x)} \theta_S^k(x)$, then we establish our theoretical claim stated in Lemma 6.

**Lemma 1** *Assume that the loss $L(\theta(x), y)$ is convex in its first argument and that there exists a $\lambda \in \mathbb{R}^n$ where $\lambda \geq 0$ and $\lambda^\top \mathbb{1} = 1$, such that the target distribution is exactly equal to the mixture of source distributions, i.e., $Q_T = \sum_{i=1}^n \lambda_i Q_S^i$. Set the target predictor as the following convex combination of the optimal source predictors*

$$\theta_T(x) = \sum_{k=1}^n \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^n \lambda_j Q_S^j(x))} \theta_S^k(x).$$

*Recall the pseudo-labeling loss* (2.10)*. Then, for this target predictor, over the target distribution, the unsupervised loss induced by the pseudo-labels and the supervised loss are both less than or equal to the loss induced by the best source predictor. In particular,*

$$\mathcal{L}(Q_T, \theta_T) \leq \min_{1 \leq j \leq n} \mathcal{L}(Q_T, \theta_S^j).$$

*Let $\alpha = \arg\min_{1 \leq j \leq n}\ \mathcal{L}(Q_T, \theta_S^j)$. Additionally, this inequality is strict if the entries of $\lambda$ are strictly positive and there exists a source $i$ for which the strict inequality $\mathcal{L}(Q_S^i, \theta_S^i) < \mathcal{L}(Q_S^i, \theta_S^\alpha)$ holds.*

**Proof.** See proof in the Appendix-1(2.7.1). ■ Observe that the expected loss $\mathcal{L}$ defined in Lemma 1 is the supervised loss where one does have the label information. Our proposed target predictor $\theta_T$ achieves a supervised loss at least as good as the best individual source model. Importantly, the inequality is strict under a natural mild condition: The best individual source model $\beta$ (for the target $Q_T$) is strictly worse than some source model $i$ on the source distribution $Q_S^i$. We also note the key differences between our algorithm and the predictor in Lemma 1. In our algorithm's combination rule, we fine-tune the feature extractors of each source model unlike Lemma 1. However each source has an individual weight which is agnostic to the source data, whereas Lemma 1 uses different weights per input instance. Below we provide an intuitive justification for choosing this input-agnostic weighting strategy.

Since we do not know the source distributions (due to the unavailability of source data), let us consider the least informative of all the distributions i.e. uniform distribution for sources by the *Principle of Maximum Entropy* [75]. This uniformity is assumed over the target support set $\mathcal{X}$. In what follows, we will consider the restrictions of the source distributions to the target support $\mathcal{X}$. Mathematically, our assumption is $Q_S^k(x) = c_k \mathcal{U}(x)$ when restricted to the support set $x \in \mathcal{X}$, where $c_k$ is a scaling factor which captures the relative contribution of source $k$ and $\mathcal{U}(x)$ has value 1. If we plug this value of the distribution in the combination rule in Lemma 1, we get $\theta_T(x) = \sum_{k=1}^{n} \frac{\lambda_k c_k}{\sum_{j=1}^{n} \lambda_j c_j} \theta_S^k(x)$ (see Appendix-1 (2.7.2) for more details). This term consisting of $\lambda_k$ and $c_k$ essentially becomes the weighting term $\alpha_k$ in our algorithm. We put this value of $\theta_T$ to solve the optimization (2.12) jointly with respect to this $\alpha_k$ and $\phi_S^k$. Thus, our optimization will

return us a favorable combination of source hypotheses, satisfying the bounds in Lemma 1, under the uniformity assumption of source distributions.

## 2.5 Experiments

**Datasets.** To test the effectiveness of our algorithm, we experiment on various visual benchmarks described as follows.

• *Office* [65]: In this benchmark DA dataset there are three domains under the office environment namely Amazon (A), DSLR (D) and Webcam (W) with a total of 31 object classes in each domain.

• *Office-Caltech* [49]: This is an extension of the Office dataset, with Caltech-256 (C) added on top of the 3 existing domains by extracting 10 classes common to all domains.

• *Office-Home* [179]: Office-Home consists of four domains, namely, Art(Ar), Clipart(Cl), Product(Pr) and Real-world(Re). Each of these domains contain 65 object classes.

• *Digits*: The Digits dataset is a benchmark for DA in digit recognition. Following [139], we utilize five subsets, namely MNIST (MT), USPS (UP), SVHN (SV), MNIST-M (MM) and Synthetic Digits (SY) for our experiments.

In all of our experiments, we take turns and fix one of the domains as the target and the rest as the source domains. The source data is discarded after training the source models.

**Baseline Methods.** We compare our method against a wide array of baselines. Similar to our setting, SHOT [102] attempts unsupervised adaptation without source data. However, it adapts a single source at a time. We compare against a multi-source extension

| Source | Method | MT,UP,SV,SY → MM | MM,UP,SV,SY → MT | MM,MT,SV,SY → UP | MM,MT,UP,SY → SV | MM,MT,UP,SV → SY | Avg. |
|---|---|---|---|---|---|---|---|
| Multiple(w) | DAN[110] | 63.7 | 96.31 | 94.2 | 62.5 | 85.4 | 80.4 |
| | DANN[43] | 71.3 | 97.6 | 92.3 | 63.5 | 85.3 | 82.0 |
| | MCD[154] | 72.5 | 96.21 | 95.3 | 78.9 | 87.5 | 86.1 |
| | CORAL[164] | 62.5 | 97.2 | 93.4 | 64.4 | 82.7 | 80.1 |
| | ADDA[177] | 71.6 | 97.9 | 92.8 | 75.5 | 86.5 | 84.8 |
| | $M^3$SDA-$\beta$[139] | 72.8 | 98.4 | 96.1 | 81.3 | 89.6 | 87.6 |
| Single(w/o) | Source-best | 60.7 | 98.2 | 74.5 | 89.5 | 89.4 | 82.5 |
| | Source-worst | 21.3 | 64 | 29.3 | 7.4 | 25.7 | 29.5 |
| | SHOT[102]-best | **94.0** | 98.7 | **97.9** | **83.5** | **97.5** | **94.3** |
| | SHOT[102]-worst | 44.5 | 97.2 | 96.2 | 29.5 | 32.5 | 60.0 |
| Multiple(w/o) | SHOT[102]-Ens | 90.4 | 98.9 | 97.7 | 58.3 | 83.9 | 85.8 |
| | DECISION(Ours) | 93.0 | **99.2** | 97.8 | 82.6 | **97.5** | 94.0 |

Table 2.1: **Results on digit recognition.** MT, MM, UP, SV, SY are abbreviations of *MNIST, MNIST-M, USPS, SVHN* and *Synthetic Digits* respectively. Multiple and Single denotes the methods which uses multiple and single sources respectively for domain adaptation, while (w) and (w/o) are abbreviations of *with source data* and *without source data* respectively. *Source* is the accuracy with the unadapted models, whereas *-best* and *-worst* refer to the best and worst sources.

| Source | Method | A,D → W | A,W → D | D,W → A | Avg. |
|---|---|---|---|---|---|
| Single | Source-best | 96.3 | 98.4 | 62.5 | 85.7 |
| | Source-worst | 75.6 | 80.9 | 62.0 | 72.8 |
| | SHOT [102]-best | 98.2 | 99.6 | 75.1 | 90.9 |
| | SHOT [102]-worst | 90.6 | 94.2 | 72.9 | 85.9 |
| Multiple | SHOT [102]-Ens | 94.9 | 97.8 | 75.0 | 89.3 |
| | DECISION(Ours) | **98.4** | **99.6** | **75.4** | **91.1** |

Table 2.2: **Results on Office**: A,D and W are abbreviations of *Amazon*, *DSLR* and *Webcam*. For single source methods, Source-best and Source-worst denote the best and worst unadapted source models, whereas SHOT-best, SHOT-worst are the best and worst accuracies of adapted source models.

of SHOT via ensembling - we pass the target data through each of the adapted source model and take an average of the soft prediction to obtain the test label. In our comparisons, we name this method SHOT-ens. We also compare against single source baselines, namely SHOT-best and SHOT-worst, which refer to the best adapted source model and the worst one respectively, learned using SHOT. Additionally, we run comparisons against traditional multi-source adaptation methods M$^3$SDA-$\beta$[139], DAN [110], DANN [43], MCD [154], CORAL [164], ADDA [177], DCTN[205]. All these methods, except for SHOT, have access to source data during adaptation.

| Source | Method | Ar,Cl,Pr → Rw | Ar,Cl,Rw → Pr | Ar,Pr,Rw → Cl | Cl,Pr,Rw → Ar | Avg. |
|---|---|---|---|---|---|---|
| Single(w/o) | Source-best | 74.1 | 78.3 | 46.2 | 65.8 | 66.1 |
| | Source-worst | 64.8 | 62.8 | 40.9 | 53.3 | 55.5 |
| | SHOT[102]-best | 81.3 | 83.4 | 57.2 | 72.1 | 73.5 |
| | SHOT[102]-worst | 80.8 | 77.9 | 53.8 | 66.6 | 69.8 |
| Multiple(w/o) | SHOT[102]-Ens | 82.9 | 82.8 | 59.3 | 72.2 | 74.3 |
| | DECISION(Ours) | **83.6** | **84.4** | **59.4** | **74.5** | **75.5** |

Table 2.3: **Results on Office-Home.**: AR,CL,RW and PR are abbreviations of *Art*, *Clipart*,*Real-world* and *Product*. We see that our method outperforms all the baselines including the best source accuracy as well as ensemble method. The abbreviations under the column SOURCE and METHOD are same as described in Table 3.2.

### 2.5.1 Implementation details

**Network architecture.** For the object recognition tasks, we use a pre-trained ResNet-50 [58] as the feature extractor backbone, similar to [139, 206]. Following [102, 43], we replace the penultimate fully-connected layer with a bottleneck layer and a task specific classifier layer. Batch normalization [72] is utilized after the bottleneck layer, along with weight normalization [157] in the final layer. For the digit recognition task, we use a variant of the LeNet [95] similar to [102].

**Source model training.** Following [102], we train the source models using smooth labels, instead of the usual one-hot encoded labels. This increases the robustness of the model and helps in the adaptation process by encouraging features to lie in tight, evenly separated clusters [121]. The maximum number of epochs for Digits, Office, Office-Home and Office-

| Source | Method | A,C,D → W | A,C,W → D | C,D,W → A | A,D,W → C | Avg. |
|---|---|---|---|---|---|---|
| Multiple(w) | ResNet-101[58] | 99.1 | 98.2 | 88.7 | 85.4 | 92.9 |
| | DAN[110] | 99.5 | 99.1 | 91.6 | 89.2 | 94.8 |
| | DCTN[205] | 99.4 | 99.0 | 92.7 | 90.2 | 95.3 |
| | MCD[154] | 99.5 | 99.1 | 92.1 | 91.5 | 95.6 |
| | $M^3$SDA-$\beta$[139] | 99.5 | 99.2 | 94.5 | 99.2 | 96.4 |
| Single(w/o) | Source-best | 98.9 | 99.3 | 94.8 | 86.5 | 94.9 |
| | Source-worst | 86.7 | 89.8 | 89.6 | 83.2 | 87.4 |
| | SHOT-best | 99.6 | 100 | 95.8 | 95.5 | 97.7 |
| | SHOT-worst | 97.3 | 96.2 | 95.7 | 93.9 | 95.8 |
| Multiple(w/o) | SHOT-Ens | 99.6 | 96.8 | 95.7 | 95.8 | 97.0 |
| | DECISION(Ours) | **99.6** | **100** | **95.9** | **95.9** | **98.0** |

Table 2.4: **Results on Office-Caltech Dataset**:A,D,C and W are abbreviations of *Amazon*, *DSLR*, *Caltech-256* and *Webcam*. Our method consistently outperform all the baselines across all the domains as target.The abbreviations under the column SOURCE and METHOD are same as described in Table 3.2.

Caltech is set to 30, 100, 50 and 100, respectively. Additionally, for our experiments on digit recognition, we resize images from each domain to 32×32 and convert the gray-scale images to RGB.

**Hyper-parameters.** The entire framework is trained in an end-to-end fashion via back-propagation. Specifically, we utilize stochastic gradient descent with momentum value 0.9 and weight decay equalling $10^{-3}$. The learning rate is set at $10^{-2}$ for the bottleneck and classifier layers, while the backbone is trained at a rate of $10^{-3}$. In addition, we use the learning rate scheduling strategy from [43], where the initial rate is exponentially decayed as learning progresses. The batch size is set to 32. We use $\lambda = 0.3$ for all the object recognition tasks and $\lambda = 0.1$ for the digits benchmark. For adaptation, maximum number of epochs is set to 15, with the pseudo-labels updated at the start of every epoch. We use PyTorch [135] for all our experiments.

### 2.5.2 Digit recognition

The results on digit recognition are shown in Table 2.1. The digit benchmark is characterised by the presence of very poor sources in some scenarios, notably when treating MNIST-M, SVHN or Synthetic Digits as the target domain. For example, on SVHN as the target, the best and worst source models adapted using SHOT [102] exhibit a performance gap of more than 50%. Combining these models via uniform ensembling results in a predictor which greatly underperforms the best adapted source. In contrast, our method restricts this severe negative transfer via a joint adaptation over the models and the ensembling weights, and outperforms the baseline by **24.3**%. The corresponding increase in performance when using Synthetic Digits and MNISTM as the target are **13.5**% and **2.6**%

respectively. Overall, we obtain an average increase of **8.2**% across all the digit adaptation tasks over SHOT-Ens. In spite of such disparities among the sources, our framework also achieves performance at par with the best adapted source and actually outperforms the latter on the MNIST transfer task. We also outperform the traditional multi-source adaptation methods, which use source data, on all the tasks by an average of **6.4**%.

### 2.5.3 Object recognition

**Office.** The results for the 3 adaptation tasks on the Office dataset are shown in Table 2.2. We achieve performance at par with the best adapted source models on all the tasks and obtain an average increase of **5.2**% over SHOT-Ens. In the task of adapting to the Webcam (W) domain, negative transfer from the Amazon (A) model brings the ensemble performance down - our model is able to prevent this, and not only outperforms the ensemble by **3.5**% but also achieves higher performance than the best adapted source.

**Office-Home.** On the Office-Home dataset, we outperform all baselines as shown in Table 2.3. Across all tasks, our method achieves a mean increase in accuracy of **2**% over the respective best adapted source models. This can be attributed to the relatively small performance gap between the best and worst adapted sources in comparison to other datasets. This suggests that, as the performance gap between the best and worst sources gets smaller, or outlier sources are removed, our method can generalize even better to the target.

**Office-Caltech.** The results follow a similar trend on the Office-Caltech dataset, as shown in Table 2.4. With a mean accuracy of **98**% across all tasks, we outperform all baselines.

| Method | A,D → W | A,W → D | D,W → A | Avg. |
|---|---|---|---|---|
| $\mathcal{L}_{pl}$ | 97.6 | 98.5 | 75.3 | 90.5 |
| $-\mathcal{L}_{ent}$ | 96.6 | 99.0 | 68.5 | 88.0 |
| $-\mathcal{L}_{ent} + \mathcal{L}_{div}$ | 95.9 | 99.0 | 71.6 | 88.9 |
| $-\mathcal{L}_{ent} + \mathcal{L}_{div} + \lambda\mathcal{L}_{pl}$ | **98.4** | **99.6** | **75.4** | **91.1** |

Table 2.5: **Loss-wise ablation.** Contribution of each component in adaptation on the Office dataset.

### 2.5.4 Ablation study

**Contribution of each loss.** Our framework is trained using a combination of three distinct losses: $\mathcal{L}_{\text{div}}$, $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{pl}}$. We study the contribution of each component of our framework to the adaptation task in Table 2.5. First, we remove both the diversity loss and the pseudo-labeling, and train using only $\mathcal{L}_{\text{ent}}$. Next, we add in $\mathcal{L}_{\text{div}}$ and perform weighted information maximization. Finally, we also compare the results of solely using $\mathcal{L}_{\text{pl}}$.

**Analysis on the learned weights.** Our framework jointly adapts the the source models and learns the weights on each such source. To understand the impact of the weights, we propose to freeze the feature extractors and optimize solely over the weights $\{\alpha_j\}_{i=1}^{n}$. Naturally, this setup yields better performance compared to trivially assigning equal weights to all source models, as shown in Table 2.6. More interestingly, the learned weights correctly indicate which source model performs better on the target and could serve as a proxy indicator in a model selection framework. See Figure 2.3.

| Method | Ar,Cl,Pr → Rw | Ar,Cl,Rw → Pr | Ar,Pr,Rw → Cl | Cl,Pr,Rw → Ar | Avg. |
|---|---|---|---|---|---|
| Source-Ens | 67.6 | 51.4 | 77.7 | 80.1 | 69.2 |
| DECISION-weights | **68.8** | **52.3** | **79.2** | **80.4** | **70.2** |

Table 2.6: **Performance on freezing backbone network on Office-Home.** DECISION-weight is optimized solely over the source weights and consistently performs better than uniform weighting.

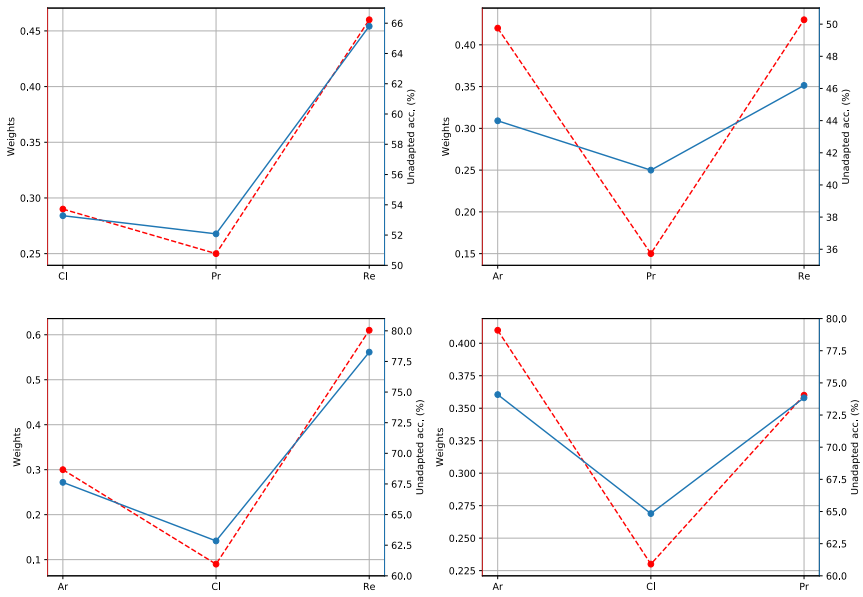



Figure 2.3: **Weights as model selection proxy.** The weights learnt by our framework on Office-Home correlates positively with the unadapted source model performance. (Left axis corresponds to the red plot and right to the blue plot, best viewed in color.)

**Distillation into a single model.** Since we are dealing with multiple source models, inference time is of the order $\mathcal{O}(m)$, where $m$ is the number of source models. If $m$ is

large, this can lead to inference being quite time consuming. To ameliorate this overhead, we follow a knowledge distillation [60] strategy to obtain a single target model. Teacher supervision is obtained by linearly combining the adapted models via the learned weights. These annotations are subsequently used to train the single student model via vanilla cross-entropy loss. Results obtained using this strategy are presented in the Appendix-1(2.7.3).

## 2.6 Conclusion

We developed a new UDA algorithm that can learn from and optimally combine multiple source models without requiring source data. We provide theoretical intuitions for our algorithm and verify its effectiveness in a variety of domain adaptation benchmarks. There are multiple exciting directions to pursue including: First, we suspect that our algorithm's performance can be further boosted by incorporating data augmentation techniques during training. Second, when there are too many source models to utilize, it would be interesting to study whether we can automatically select an optimal subset of the source models without requiring source data in an unsupervised fashion.

## 2.7 Appendix-1

### 2.7.1 Proof of Lemma 1

**Lemma 2** *Assume that the loss $L(\theta(x), y)$ is convex in its first argument and that there exists a $\lambda \in \mathbb{R}^n$ where $\lambda \geq 0$ and $\lambda^\top \mathbb{1} = 1$, such that the target distribution is exactly equal to the mixture of source distributions, i.e $Q_T = \sum_{i=1}^{n} \lambda_i Q_S^i$. Set the target predictor as the*

*following convex combination of the optimal source predictors*

$$\theta_T(x) = \sum_{k=1}^{n} \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^{n} \lambda_j Q_S^j(x))} \theta_S^k(x).$$

*Recall the pseudo-labeling loss (10). Then, for this target predictor, over the target distribution, the unsupervised loss induced by the pseudo-labels and the supervised loss are both less than or equal to the loss induced by the best source predictor. In particular,*

$$\mathcal{L}(Q_T, \theta_T) \leq \min_{1 \leq j \leq n} \mathcal{L}(Q_T, \theta_S^j).$$

**Proof.** We can see that the left hand-side of the inequality can be upper-bounded by some loss as follows,

$$\begin{aligned}
\mathcal{L}(Q_T, \theta_T) &= \int_x Q_T(x) L(\theta_T(x), y) = \int_x Q_T(x) L\left( \sum_{i=1}^{n} \frac{\lambda_i Q_S^i(x)}{\sum_{j=1}^{n} \lambda_j Q_S^j(x))} \theta_S^i(x), y \right) dx \\
&\leq \int_x Q_T(x) \sum_{i=1}^{n} \frac{\lambda_i Q_S^i(x)}{\sum_{j=1}^{n} \lambda_j Q_S^j(x))} L(\theta_S^i(x), y) dx \quad \text{(from Jensen's inequality)} \\
&= \int_x Q_T(x) \sum_{i=1}^{n} \frac{\lambda_i Q_S^i(x)}{Q_T(x)} L(\theta_S^i(x), y) dx \quad \text{(from distribution assumption)} \\
&= \sum_{i=1}^{n} \lambda_i \int_x Q_S^i(x) L(\theta_S^i(x), y) dx \quad \text{(changing the order of summation)} \\
&= \sum_i \lambda_i \mathcal{L}(Q_S^i(x), \theta_S^i)
\end{aligned} \tag{2.13}$$

Now for the R.H.S. we can write this loss as follows,

$$\begin{aligned}\mathcal{L}(Q_T, \theta_S^j) &= \int_x Q_T(x) L(\theta_S^j(x), y) dx \\ &= \int_x \sum_{i=1}^{n} \lambda_i Q_S^i(x) L(\theta_S^j(x), y) dx \\ &= \sum_{i=1}^{n} \lambda_i \int_x Q_S^i L(\theta_S^j(x), y) dx \\ &= \sum_{i=1}^{n} \lambda_i \mathcal{L}(Q_S^i(x), \theta_S^j)\end{aligned} \tag{2.14}$$

Now recall from main chapter 2 that,

$$\theta_S^k = \arg\min_{\theta} \mathcal{L}(Q_S^k, \theta) \quad \text{for} \quad 1 \leq k \leq n.$$

. This means $\theta_S^i$ is the best predictor for the source $i$, which has distribution $Q_S^i$. Thus we find that $\mathcal{L}(Q_S^i, \theta_S^i) \leq \mathcal{L}(Q_S^i, \theta_S^j)\ \forall j$, which implies $\sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^i) \leq \sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^j)$. This further implies that $\mathcal{L}(Q_T, \theta_T) \leq \mathcal{L}(Q_T, \theta_S^j)\ \forall j$, which in turn concludes the proof $\mathcal{L}(Q_T, \theta_T) \leq \min_{1 \leq j \leq n} \mathcal{L}(Q_T, \theta_S^j)$. Finally, suppose the entries of $\lambda$ are strictly positive and let $\beta = \arg\min_j \mathcal{L}(Q_T, \theta_S^j)$. Observe that, if there is a source $i$ such that the strict inequality $\mathcal{L}(Q_S^i, \theta_S^i) < \mathcal{L}(Q_S^i, \theta_S^\beta)$ holds, then the main claim of the lemma also becomes strict as we find

$$\mathcal{L}(Q_T, \theta_T) \leq \sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^i) < \sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^\beta) \leq \min_j \mathcal{L}(Q_T, \theta_S^j).$$

Verbally, this strict inequality has a natural meaning that the model $j$ is strictly worse than model $i$ for the source data $i$. ■

### 2.7.2 Detailed steps of combination rule under source distribution uniformity assumption

See the discussion after **Lemma 1** in the main chapter 2 for reference.

$$\begin{aligned}\theta_T(x) &= \sum_{k=1}^{n} \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^{n} \lambda_j Q_S^j(x)} \theta_S^k(x) \\ &= \sum_{k=1}^{n} \frac{\lambda_k c_k \mathcal{U}(x)}{\sum_{j=1}^{n} \lambda_j c_j \mathcal{U}(x)} \theta_S^k(x) \\ &= \sum_{k=1}^{n} \frac{\lambda_k c_k}{\sum_{j=1}^{n} \lambda_j c_j} \theta_S^k(x)\end{aligned} \tag{2.15}$$

### 2.7.3 Additional Experiments

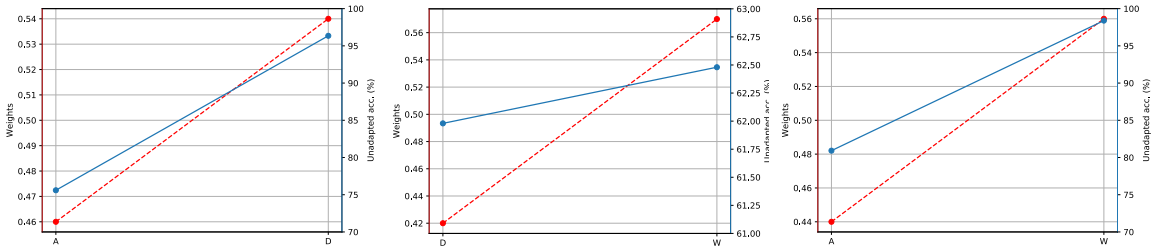



Figure 2.4: **Weights as model selection proxy.** The weights learnt by our framework on Office-31 correlates positively with the unadapted source model performance. (Left axis corresponds to the red plot and right to the blue plot, best viewed in color.)

From Figure 2.4, we can clearly see that for the model which gives higher accuracy for the unadapted scenario, it is automatically given higher weightage by our algorithm. As a result, we can easily infer about the quality of the source domain, in relation to the target, from the weights learnt by our framework.

**Effect of weight on pseudo-labeling.** We investigate the effect of the weight $\lambda$ on $\mathcal{L}_{\text{pl}}$. We perform experiments on the Office dataset by varying the value of $\lambda$ and plot the results

in Figure 2.5. As shown in the plot, the proposed method performs best at $\lambda = 0.3$

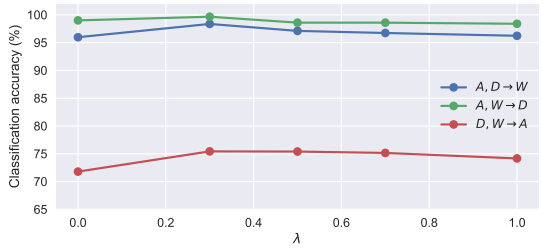



Figure 2.5: **Effect of $\lambda$.** The variations in classification as the weight on $\mathcal{L}_{\text{pl}}$ is varied. (Best viewed in color)

**Effect of outlier source models.** Our method is clearly robust to outlier source models. In Table 2 of the main chapter 2, when *MNIST-M* is the target, transferring from only *USPS*, leads to an extremely poor performance of **21.3**% - here, USPS is a strong outlier. Despite the presence of such a poor source, our framework is mostly able to correctly negate the transfer from USPS, achieving a performance of **93**%, close to the best source performance of **94**%. On removing *USPS* as a source, DECISION outperforms the best source by achieving an accuracy of **94.5**%. In the future, we plan to actively use the weights to simultaneously remove poor sources while adaptation in order to boost the performance.

**DomainNet [32]:** This is a relatively new and large dataset where there are six domains under the common object categories, namely quickdraw (Q), clipart (C), painting (P), infograph (I), sketch (S) and real (R) with a total of 345 object classes in each domain. Experimental results on this dataset are shown in Table 2.7. Our method consistently outperforms the best adapted source baselines (SHOT-best) except for *infograph* as a target. However the average performance over all the domains as target is slightly less than the

| Source | Method | C,P,I,S,R → Q | Q,P,I,S,R → C | Q,C,I,S,R → P | Q,C,P,S,R → I | Q,C,P,I,R → S | Q,C,P,I,S → R | Avg. |
|---|---|---|---|---|---|---|---|---|
| Multiple(w) | DAN[25] | 16.2 | 39.1 | 33.3 | 11.4 | 29.7 | 42.1 | 28.6 |
| | DCTN[46] | 7.2 | 48.6 | 48.8 | 23.4 | 47.3 | 53.5 | 38.1 |
| | MCD[37] | 7.6 | 54.3 | 45.7 | 22.1 | 43.5 | 58.4 | 38.6 |
| | $M^3$SDA-$\beta$[32] | 6.3 | 58.6 | 52.3 | 26 | 49.5 | 62.7 | 42.5 |
| Single(w/o) | Source-best | 11.9 | 49.9 | 47.5 | 20 | 41.1 | 57.7 | 38 |
| | Source-worst | 2.3 | 12.2 | 2.2 | 1.1 | 8.7 | 4.8 | 5.2 |
| | SHOT[22]-best | 18.7 | 58.3 | 53 | 22.7 | 48.4 | 65.9 | 44.5 |
| | SHOT[22]-worst | 3.8 | 14.8 | 3.5 | 1 | 11.9 | 6.6 | 7 |
| Multiple(w/o) | SHOT[22]-Ens | 15.3 | 58.6 | **55.3** | **25.2** | **52.4** | **70.5** | **46.2** |
| | DECISION(Ours) | **18.9** | **61.5** | 54.6 | 21.6 | 51 | 67.5 | 45.9 |

Table 2.7: **Results on DomainNet**:Q,C,P,I,S and R are abbreviations of *quickdraw*, *clipart*, *painting*, *infograph*, *sketch* and *real*.

SHOT-Ens. Note that for *quickdraw* and *clipart* as target, our method outperforms all the state of the art methods including source free and with source data single and multi source state-of-the-art DA methods.

**Distillation.** Our results on using the distillation strategy outlined in Section 5.4 of the main chapter 2 are shown in Table 2.8. Despite the model compression, the performance remains consistent.

| Method | Office-home | | | | Office-Caltech | | | | Office | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rw | Pr | Cl | Ar | A | C | D | W | A | D | W |
| DECISION (original) | 83.6 | 84.4 | 59.4 | 74.5 | 95.9 | 95.9 | 100 | 99.6 | 75.4 | 99.6 | 98.4 |
| DECISION (distillation) | 83.7 | 84.4 | 59.1 | 74.4 | 96.0 | 95.7 | 99.4 | 99.6 | 75.4 | 99.6 | 98.1 |

Table 2.8: **Distillation results on object recognition tasks.** Performance remains consistent across all datasets despite distilling into a single target model.

# Chapter 3

# Multi source Test Time Adaptation

## 3.1 Introduction

Deep neural networks have shown impressive performance on test inputs that closely resemble the training distribution. However, their performance degrades significantly when they encounter test inputs from a different data distribution. Unsupervised domain adaptation (UDA) techniques [177, 175] aim to mitigate this performance drop. Addressing the distribution shift in case of *dynamic data distributions* is even more challenging and practically relevant - in many real-world applications like autonomous navigation, models often encounter dynamically evolving distributions. Furthermore, test data is often accessed in streaming batches rather than all at once, and source data may not always be available due to privacy and storage concerns.

For domain adaptation to dynamically evolving environments, employing a model ensemble can be beneficial, as it allows leveraging the learned knowledge of different models to more effectively mitigate dynamic distribution shifts. Additionally, situations may arise

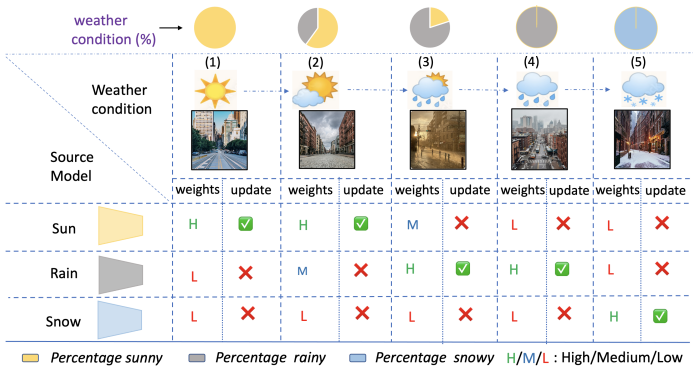



Figure 3.1: **Problem setup.** Consider several source models trained using data from different weather conditions. During the deployment of these models, they may encounter varying weather conditions that could be a combination of multiple conditions in varying proportions (represented by the pie charts on top). Our goal is to infer on the test data using the ensemble of models by automatically figuring out proper combination weights and adapting the appropriate models on the fly.

wherein the user has access to a diverse set of pre-trained models across distinct source domains, and no access to source domain data corresponding to each model due to privacy, storage or other constraints. Consequently, training a unified model using the combined source data becomes unfeasible. In those scenarios, it is both reasonable and effective to employ and adapt the entire available array of source models during testing, thereby enhancing performance beyond the scope of single source model adaptation. Moreover, employing a model ensemble provides the flexibility to effortlessly incorporate or exclude models post-deployment, aligning with the user's preferences and the needs of the given task. This flexibility is not achievable with a single domain-generalized model trained on combined source data.

As an example, consider a scenario where a recognition model, initially trained on clear weather conditions, faces data from mixed weather scenarios, like sunshine interspersed with rain (see Figure 3.1). In such cases, employing multiple models - specifically those trained on clear weather and rain — with appropriate weighting can potentially reduce the test error as opposed to relying on a single source model. In this context, the models for clear weather and rain would be assigned higher weights, while models for other weather conditions would receive relatively lesser weightage.

The main challenge of developing such a model ensembling method is to *learn appropriate combination weights to optimally combine the source model ensemble during the test phase as data is streaming in, such that it results in a test error equal or lower than that of the best source model.* To solve this, we propose CONtinual mulTi-souRce Adaptation to dynamic diStribuTions (CONTRAST) that handles multiple source models and optimally combines them to adapt to the test data.

The efficacy of using multiple source models also extends to preventing *catastrophic forgetting* that may arise when adapting to dynamic distributions for a prolonged time. Consider again the scenario of multiple source models, each trained on a different weather condition. During inference, only the parameters of the models most closely related to the weather encountered during test time will get updates, and the unrelated ones will be left untouched. This ensures that the model parameters do not drift too far from the initial state, since only those related to the test data are being updated. This mechanism mitigates forgetting when the test data distribution varies over a long time scale, as is likely to happen in most realistic conditions. Even if an entirely unrelated distribution appears

during testing and there is no one source model to handle it, the presence of multiple sources can significantly reduce the rate at which the forgetting occurs. This is again because only the most closely related models (clear and rainy weather in the example above) are updated, while others (e.g., snow) are left untouched. Our setting is closely related to Test Time Adaptation methods (TTA) [182], and ours is the first work to consider *dynamically evolving multi-source* adaptation at test time.

**Main Contributions.** Our proposed approach, CONTRAST, makes the following contributions.

- We propose a framework for multi-source adaptation to dynamic distribution shifts from streaming test data and without access to the source data. Our approach has the ability to merge the source models using appropriate combination weights during test time, enabling it to perform just as well as the best-performing source or even surpass it.
- Our framework achieves performance on par with the best-performing source and also effectively mitigates catastrophic forgetting when faced with long-term, fluctuating test distributions.
- We provide theoretical insights on CONTRAST, illustrating how it addresses domain shift by optimally combining source models and prioritizing updates to the model least prone to forgetting.
- To demonstrate the real-world advantages of our methodology, we perform experiments on a diverse range of benchmark datasets.

## 3.2 CONTRAST Framework

### 3.2.1 Problem Setting

In this problem setting, we propose to combine multiple pre-trained models during test time through the application of suitable combination weights, determined based on a limited number of test samples. Specifically, we will focus on the classification task that involves $K$ categories. Consider the scenario where we have a collection of $N$ source models, denoted as $\{\mathrm{f}_S^j\}_{j=1}^N$, that we aim to deploy during test time. In this situation, we assume that a sequence of test data $\{x_i^{(1)}\}_{i=1}^B \to \{x_i^{(2)}\}_{i=1}^B \to \ldots \{x_i^{(t)}\}_{i=1}^B \to \ldots$ are coming batch by batch in an online fashion, where $t$ is the index of time-stamp and $B$ is the number of samples in the test batch. We also denote the test distribution at time-stamp $t$ as $\mathcal{D}_T^{(t)}$, which implies $\{x_i^{(t)}\}_{i=1}^B \sim \mathcal{D}_T^{(t)}$. Motivated by [6], we model the test distribution in each time-stamp $t$ as a linear combination of source distributions where the combination weights are denoted by $\{\mathrm{w}_j^{(t)}\}_{j=1}^N$. Thus, our inference model on test batch $t$ can be written as $\mathrm{f}_T^{(t)} = \sum_{j=1}^N \mathrm{w}_j^{(t)} \mathrm{f}_S^{j(t)}$ where $\mathrm{f}_S^{j(t)}$ is the adapted $j$-th source in time stamp $t$. Based on this setup our objective is twofold:

1. We want to determine the optimal combination weights $\{\mathrm{w}_j^{(t)}\}_{j=1}^N$ for the current test batch such that the test error for the optimal inference model is lesser than or equal to the test error of best source model. Mathematically we can write this as follows:

$$\epsilon_{test}^{(t)}(\mathrm{f}_T^{(t)}) \leq \min_{1 \leq j \leq N} \epsilon_{test}^{(t)}(\mathrm{f}_S^j), \tag{3.1}$$

   where $\epsilon_{test}^{(t)}(\cdot)$ evaluates the test error on $t$-th batch.

2. We also aim for the model to maintain consistent performance on source domains, as

it progressively adapts to the changing test conditions. This is necessary to ensure that the model has not catastrophically forgotten the original training distribution of the source domain and maintains its original performance if the source data is re-encountered in the future We would ideally want to have:

$$\epsilon_{src}(\mathrm{f}_S^{j(t)}) \approx \epsilon_{src}(\mathrm{f}_S^{j}) \quad \forall j, t, \tag{3.2}$$

where, $\epsilon_{src}(\mathrm{f}_S^{j})$ denote the test error of $j$-th source on its corresponding test data when using the original source model $\mathrm{f}_S^{j}$, whereas $\epsilon_{src}(\mathrm{f}_S^{j(t)})$ represents the test error on the same test data using the $j$-th source model adapted up to time step $t$, denoted as $\mathrm{f}_S^{j(t)}$.

### 3.2.2 Overall Framework

Our framework undertakes two operations on each test batch. First, we learn the combination weights for the current batch at time step $t$ by freezing the model parameters. Then, we update the model corresponding to the largest weight with existing state-of-the-art TTA methods, which allows us to fine-tune the model and improve its performance. This implies that the model parameters of source $j$ might get updated up to $p$ times at time-step $t$, where $0 \leq p \leq (t-1)$.

In other words, the states of the source models evolve over time depending on the characteristics of the test batches up to the previous time step. To formalize this concept, we define the state of the source model $j$ at time-step $t$ as $\mathrm{f}_S^{j(t)}$. In the next section, we will provide a detailed explanation of both aspects of our framework: (i) learning the combination weights, and (ii) updating the model parameters. By doing so, we aim to provide a comprehensive understanding of how our approach works in practice.

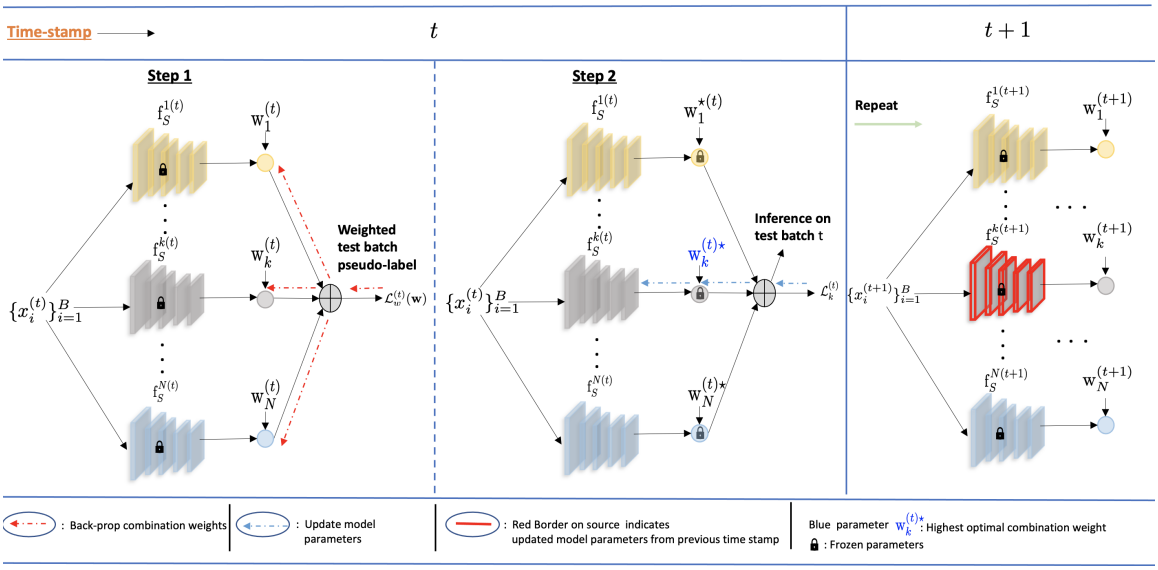



Figure 3.2: **Overall Framework.** During test time, we aim to adapt multiple source models in a manner such that it optimally blends the sources with suitable weights based on the current test distribution. Additionally, we update the parameters of only one model that exhibits the strongest correlation with the test distribution.

### 3.2.3 Learning the combination weights

For an unlabeled target sample $x_i^{(t)}$ that arrives at time-stamp $t$, we denote its pseudo-label, as predicted by source $j$, as $\hat{y}_{ij}^{(t)} = \mathrm{f}_S^{j(t)}(x_i^{(t)})$, where $\mathrm{f}_S^{j(t)}$ is the state of source $j$ at time-stamp $t$. Now we linearly combine these pseudo-labels by source combination weights $\mathbf{w} = [\mathrm{w}_1\ \mathrm{w}_2 \dots \mathrm{w}_N]^\top \in \mathbb{R}^N$ to get weighted pseudo-label $\hat{y}_i^{(t)} = \sum_{j=1}^{N} \mathrm{w}_j \hat{y}_{ij}^{(t)}$. Using these weighted pseudo-labels for all the samples in the $t$-th batch we calculate the expected Shannon entropy as,

$$\mathcal{L}_w^{(t)}(\mathbf{w}) = -\mathbb{E}_{\mathcal{D}_T^{(t)}} \sum_{c=1}^{K} \hat{y}_{ic}^{(t)} \log(\hat{y}_{ic}^{(t)}) \tag{3.3}$$

Based on this loss we solve the following optimization:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathcal{L}_w^{(t)}(\mathbf{w}) \\ \text{subject to} \quad & \mathrm{w}_j \geq 0, \forall j \in \{1, 2, \ldots, N\}, \\ & \sum_{j=1}^{n} \mathrm{w}_j = 1 \end{aligned} \tag{3.4}$$

Suppose we get $\mathbf{w}^{\star(t)}$ to be the optimal combination weight vector by performing the optimization in (3.4). In such case, the optimal inference model for test batch $t$ can be expressed as follows:

$$\mathrm{f}_T^{(t)} = \sum_{j=1}^{N} \mathrm{w}_j^{\star(t)} \mathrm{f}_S^{j(t)} \tag{3.5}$$

Thus, by learning $\mathbf{w}$ in this step, we satisfy Eqn. (3.1).

**Model parameter update.** After obtaining $\mathbf{w}^{\star(t)}$, next we select the most relevant source model $k$ given by $k = \arg \max_{1 \leq j \leq N} \mathrm{w}_j^{\star(t)}$. This indicates that the distribution of the current test batch is most correlated with the source model $k$. We then adapt model $k$ to the test batch $t$ using any state-of-the-art single source method that adapts to dynamic target distributions. Specifically, we employ three distinct adaptation approaches: (i) TENT [182], (ii) CoTTA [188], and (iii) EaTA [126]. For a more in-depth discussion of these adaptation methods, please consult the Appendix-2.

**Optimization strategy for (3.4).** Solving the optimization problem in Eq. 3.4 is a prerequisite for inferring the current test batch. As inference speed is critical for test-time adaptation, it is desirable to learn the weights quickly. To achieve this, we design two strategies: (i) selecting an appropriate initialization for $\mathbf{w}$, and (ii) determining an optimal learning rate.

**(i) Good initialization:** Pre-trained models contain information about expected batch

mean and variance in their Batch Norm (BN) layers based on the data they were trained on. To leverage this information, we extract these stored values from each source model prior to adaptation. Specifically, we denote the expected batch mean and standard deviation for the $l$-th layer of the $j$-th source model as $\mu_l^j$ and $\sigma_l^j$, respectively.

During testing on the current batch $t$, we pass the data through each model and extract its mean and standard deviation from each BN layer. We denote these values as $\mu_l^{T(t)}$ and $\sigma_l^{T(t)}$, respectively. One useful metric for evaluating the degree of alignment between the test data and each source is the distance between their respective batch statistics. A smaller distance implies a stronger correlation between the test data and the corresponding source. Assuming that the batch-mean statistic per node of the BN layers to be a univariate Gaussian, we calculate the distance (KL divergence) between the $j$-th source (approximated as $\mathcal{N}(\mu_l^j, (\sigma_l^j)^2)$) and the $t$-th test batch (approximated as $\mathcal{N}(\mu_l^{T(t)}, (\sigma_l^{T(t)})^2)$) as follows (derivation in Appendix-2 subsection 3.6.14):

$$\theta_j^t = \sum_l \mathcal{D}_{KL}\left[\mathcal{N}\left(\mu_l^{T(t)}, (\sigma_l^{T(t)})^2\right), \mathcal{N}\left(\mu_l^j, (\sigma_l^j)^2\right)\right] =$$
$$\sum_{l=1}^{n_j}\sum_{m=1}^{d_j^l} \log\left(\frac{\sigma_{lm}^j}{\sigma_{lm}^{T(t)}}\right) + \frac{\left(\sigma_{lm}^{T(t)}\right)^2 + \left(\mu_{lm}^j - \mu_{lm}^{T(t)}\right)^2}{2\left(\sigma_{lm}^j\right)^2} - \frac{1}{2}$$

where subscript $lm$ denotes the $m$-th node of $l$-th layer. After obtaining the distances, we use a softmax function denoted by $\delta(\cdot)$ to normalize their negative values. The softmax function is defined as $\delta_j(a) = \frac{\exp(a_j)}{\sum_{i=1}^N \exp(a_i)}$, where $a \in \mathbb{R}^N$, and $j \in 1, 2, \ldots, N$. If $\theta^t = \left[\theta_1^t, \theta_2^t \ldots \theta_N^t\right]^\top \in \mathbb{R}^N$ is the vectorized form of the distances from all the sources, we set

$$\mathbf{w}_{init}^{(t)} = \delta(-\theta^t) \tag{3.6}$$

where $\mathbf{w}_{init}^{(t)}$ is the initialization for $\mathbf{w}$. As we shall see, this choice leads to a substantial performance boost compared to random initialization.

**(ii) Optimal step size:** Since we would like to ensure rapid convergence of optimization in Eqn. 3.4 , we select the optimal step size for gradient descent in the initial stage. Given an initialization $\mathbf{w}_{init}^{(t)}$ and a step size $\alpha^{(t)}$, we compute the second-order Taylor series approximation of the function $\mathcal{L}_w^{(t)}$ at the updated point after one gradient step. Next, we determine the best step size $\alpha_{best}^{(t)}$ by minimizing the approximation with respect to $\alpha^{(t)}$. This is essentially an approximate Newton's method (details in Appendix section 3.6.15) and has a closed-form solution given by

$$\alpha_{best}^{(t)} = \left[ \left( \nabla_{\mathbf{w}} \mathcal{L}_w^{(t)} \right)^\top \left( \nabla_{\mathbf{w}} \mathcal{L}_w^{(t)} \right) / \left( \nabla_{\mathbf{w}} \mathcal{L}_w^{(t)} \right)^\top \mathcal{H}_w \left( \nabla_{\mathbf{w}} \mathcal{L}_w^{(t)} \right) \right] \Bigg|_{\mathbf{w}^{init}}. \tag{3.7}$$

Here $\nabla_{\mathbf{w}} \mathcal{L}_w^{(t)}$ and $\mathcal{H}_w$ are the gradient and Hessian of $\mathcal{L}_w^{(t)}$ with respect to $\mathbf{w}$. Together with $\mathbf{w}_{init}^{(t)}$ and $\alpha_{best}^{(t)}$, optimization of ( 3.4) converges very quickly as demonstrated in the experiments (*in Table 3.5 of Appendix-2*). Please note that, *we calculate the Hessian for only n scalar parameters, with n representing the number of source models. Typically, in common application domains, addressing distribution shifts requires only a small number of source models, making the computational overhead of calculating hessian negligible.*

We provide a complete overview of CONTRAST in Algorithm 2 in the Appendix-2 (Subsection 3.5.1).

### 3.2.4 Theoretical insights regarding combination weights

**Theorem 3 (Convergence of Optimization 3.4.)** *The Optimization 3.4 converges according to the rule as follows:*

$$\frac{1}{(k+1)}\sum_{j=0}^{k}\|\nabla_{\aleph}\mathcal{L}_w(\mathbf{w^{(j)}})\|_2^2 \leq \frac{2(\mathcal{L}_w(w^{(0)})-\mathcal{L}_w(w^{\star}))}{\alpha_{best}^{(t)}(k+1)} \tag{3.8}$$

*where, $\nabla_{\aleph}$ represents the gradient of the objective function over the set of n-simplex $\aleph$ and $j$ represents the iteration number.*

**Proof.** Please refer to the Appendix-2 for the proof. ■

**Implication of Theorem 3** The theorem tells us that to make the optimization converge faster with fewer iterations (small $k$), it is crucial to start with a good initialization close to the best solution ($(\mathcal{L}(w^{(0)}) - \mathcal{L}(w^{\star}))$ should be small). By using Eqn. (3.6), we ensure this condition for quicker convergence. Also, please note that in Theorem 3, $j$ denotes the iteration number in the optimization process, and for simplicity, the batch number $t$ has been intentionally omitted from the notation.

### 3.2.5 Theoretical insights regarding model update

We now provide theoretical justification on how CONTRAST selects the best source model by optimally trading off model accuracy and domain mismatch. At time $t$, let $\mathrm{f}_S^{(t)}$ be the set of source models defined as $\left[\mathrm{f}_S^{1(t)}\ \mathrm{f}_S^{2(t)}\ \ldots\ \mathrm{f}_S^{N(t)}\right]$. CONTRAST aims to learn a combination of these models by optimizing weights $w$ on the target domain. For simplicity of exposition, we consider convex combinations $w \in \Delta$ where $\Delta$ is the $N$-dimensional simplex.

To learn $\mathrm{w} \in \Delta$, CONTRAST runs empirical risk minimization on the target task using a loss function $\ell(\cdot)$ with pseudo-labels generated by w-weighted source models. Let $\mathcal{L}(f)$ denote the target population/test risk of a model $f$ (with respect to ground-truth labels) and $\mathcal{L}_T^{\star(t)}$ represent the optimal population risk obtained by choosing the best possible $w \in \Delta$ (i.e. oracle risk). We introduce the functions: **(1)** $\Psi$ which returns the distance between two data distributions and **(2)** $\varphi$ which returns the distance between two label distributions. We note that, rather than problem-agnostic metrics like Wasserstein, our $\Psi, \varphi$ definitions are in terms of the loss landscape and source models $\mathrm{f}_S^{(t)}$, hence tighter. We have the following generalization bound at time step $t$. precise details in Appendix-2 Section.

**Theorem 4** *Consider the model* $\mathrm{f}_T^{(t)}$ *with combination weights* $\mathrm{w}^{\star(t)}$ *obtained via CONTRAST by minimizing the empirical risk over* $B$ *IID target examples per Eqn. 3.5. Let* $\hat{y}_{\mathrm{w}}^{(t)}$ *denote the pseudo-label variable of* w*-weighted source models and* $\mathcal{D}_{\mathrm{w}}^{(t)} = \sum_{i=1}^{N} \mathrm{w}_i^{(t)} \mathcal{D}_{S_i}^{(t)}$ *denote weighted source distribution. Under Lipschitz* $\ell$ *and bounded* $\mathrm{f}_S^{(t)}$*, with probability at least* $1 - 3e^{-\tau}$ *over the target batch, test risk obeys*

$$\underbrace{\mathcal{L}(\mathrm{f}_T^{(t)})}_{CONTRAST} - \underbrace{\mathcal{L}_T^{\star(t)}}_{Optimal} \leq \min_{\mathrm{w} \in \Delta} \{ \underbrace{\Psi(\mathcal{D}_T^{(t)}, \mathcal{D}_{\mathrm{w}}^{(t)})}_{\textbf{\textit{shift}}} + \underbrace{\varphi(\hat{y}_{\mathrm{w}}^{(t)}, y_{\mathrm{w}}^{(t)})}_{\textbf{\textit{quality}}} \} + \sqrt{\tilde{\mathcal{O}}((N+\tau)/B)}.$$

**Proof.** Please refer to the Appendix-2 (Subsection 3.5.2) for the proof. ■

**Discussion.** In a nutshell, this result shows how CONTRAST strikes a balance between: (1) choosing the domain that has the smallest **shift** from target, and (2) choosing a source model that has high-**quality** pseudo-labels on its own distribution (i.e. $\hat{y}_{\mathrm{w}}^{(t)}$ matches $y_{\mathrm{w}}^{(t)}$). From our analysis, it is evident that, rather than adapting the source models to the target

distribution, if we simply optimize the combination weights to optimize pseudo-labels for inference, the left side excess risk term ($\mathcal{L}(\mathrm{f}_T^{(t)}) - \mathcal{L}_T^{\star(t)}$) becomes upper bounded by a relatively modest value. This is because the **shift** and **quality** terms on the right-hand side are optimized with respect to w. We note that $\sqrt{N/B}$ is the generalization risk due to finite samples $B$ and search dimension $N$.

To further refine this, our immediate objective is to tighten the upper bound. This can be achieved by individually adapting each source model to the current test data, all the while maintaining the optimized w constant. Yet, such an approach is not ideal since our second goal is to preserve knowledge from the source during continual adaptation. To attain our desired goal, we must relax the upper bound, reducing our search over $\mathrm{w} \in \hat{\Delta}$. Here, $\hat{\Delta}$ is the discrete counterpart of the simplex $\Delta$. The elements of $\hat{\Delta}$ are one-hot vectors that have all but one entry zero. The elements of $\hat{\Delta}$ essentially represent discrete model selection. Examining the main terms on the right reveals that: (i) source-target distribution shift and (ii) divergence between ground-truth and pseudo-labels are all minimized when we select the source model with the highest correlation to target. This model, denoted by $\mathrm{f}_S^{\star(t)}$, essentially corresponds to the largest entry of $\mathrm{w}^{\star(t)}$ and presents the most stringent upper bound within the $\hat{\Delta}$ search space. Thus, to further minimize the right hand side, the second stage of CONTRAST adapts $\mathrm{f}_S^{\star(t)}$ with the current test data. Crucially, besides minimizing the target risk, this step helps avoids forgetting the source because $\mathrm{f}_S^{\star(t)}$ already does a good job at the target task. Thus, during optimization on target data, $\mathrm{f}_S^{\star(t)}$ will have small gradient and will not move much, resulting in smaller forgetting. Please refer to the Appendix-2 (Subsection 3.5.2) for more detailed discussion along with the proof.

## 3.3 Experiments

**Datasets.** We demonstrate the efficacy of our approach using both **static target distribution** and **dynamic target data distributions**. For static case, we employ the *Digits* and *Office-Home* datasets [179]. For the dynamic case, we utilize *CIFAR-100C* and *CIFAR-10C* [59]. Detailed descriptions of these datasets and additional experiments on Digits, Office-Home and CIFAR-10C along with results on segmentation task can be found in the Appendix-2.

**Baseline Methods.** As our problem setting is most closely related to test time adaptation, our baselines are some widely used state-of-the-art (SOTA) single source test time adaptation methods: we specifically compare our algorithm with Tent [182], CoTTA [188] and EaTA [126]. These methods deal with adaptation from small batches of streaming data and without the source data, which is our setting, and hence we compare against these as our baselines. To evaluate the adaptation performance, we follow the protocol similar to [6], where we apply each source model to the test data from a particular test domain individually, which yields X-Best and X-Worst where "X" is the name of the single source adaptation method, representing the highest and lowest performances among the source models adapted using method "X", respectively. For our algorithm, we extend all of the methods "X" in the multi source setting and call the multi-source counterpart of "X" as "X+CONTRAST".

**Implementation Details.** We use ResNet-18 [58] model for all our experiments. For solving the optimization of Eq. (3.4), we first initialize the combination weights using Eq. (3.6) and calculate the optimal learning rate using Eq. (3.7). After that, we use 5 iterations

to update the combination weights using SGD optimizer and the optimal learning rate. For all the experiments we use a batch size of 128, as used by Tent [182]. For more details on implementation and experimental setting see Appendix-2.

**Experiments on CIFAR-100C.** We conduct a thorough experiment on this dataset to investigate the performance of our model under dynamic test distribution. We consider 3 corruption noises out of 15 noises from CIFAR-100C, which are adversarial weather conditions namely *Snow*, *Fog* and *Frost*. We add these noises for severity level 5 to the original CIFAR-100 training set and train three source models, one for each noise. Along with these models, we also add the model trained on clean training set of CIFAR-100. During testing, we sequentially adapt the models across the 15 noisy domains, each with a severity of 5, from the CIFAR-100C dataset [188, 126]. We report the results for this experiment in Table. 3.1. Moreover, we also conduct an experiment on CIFAR10-C with the exact same experimental settings as with CIFAR100-C. ***CIFAR-10C results are in Table 3.4 of Appendix-2***.

From the table, we can draw two key observations:

(i) As anticipated, X+CONTRAST consistently outperforms X-Best across each test distribution, underscoring the validity of our algorithmic proposition. (ii) Given that the CoTTA and EaTA methods are tailored to mitigate forgetting, the average error post-adaptation across the 15 noises using these methods is significantly lower than that of Tent, which is not designed for this specific challenge. For instance, in Table 3.1, Tent-Best error is approximately 68.2%, while CoTTA and EaTA-Best are around 39.9% and 38.5%, respectively. However, when these adaptation methods are incorporated into our framework, the

Table 3.1: **Results on CIFAR-100C.** We take four source models trained on *Clear*, *Snow*, *Fog*, and *Frost*. We employ these models for adaptation on 15 sequential test domains. This table illustrates that even in the dynamic environment X+ CONTRAST performs better than X-Best, which is the direct consequence of optimal aggregation of source models as well as better preservation of source knowledge. (Results in error rate ↓ (in %))

| | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Worst | 97.7 | 96.5 | 98.2 | 68.8 | 78.1 | 66.1 | 65.1 | 53.6 | 59.3 | 62.0 | 55.8 | 95.4 | 61.9 | 71.5 | 75.2 | 73.7 |
| Source Best | 90.5 | 89.0 | 94.5 | 50.7 | 48.1 | 51.9 | 44.5 | 30.0 | 29.5 | 28.2 | 39.0 | 81.9 | 44.0 | 38.5 | 57.1 | 54.5 |
| Tent Worst | 55.9 | 55.6 | 71.2 | 58.0 | 75.5 | 78.2 | 83.3 | 89.2 | 92.4 | 93.7 | 95.4 | 96.7 | 96.5 | 96.6 | 96.7 | 82.3 |
| Tent Best | 45.6 | 43.8 | 59.1 | 48.5 | 59.1 | 59.1 | 60.4 | 65.6 | 66.1 | 76.7 | 75.3 | 89.8 | 89.0 | 91.3 | 94.2 | 68.2 |
| Tent + CONTRAST | **42.2** | **40.6** | **55.3** | **28.6** | **40.7** | **31.9** | **29.6** | **31.7** | **32.4** | **30.9** | **28.6** | **41.5** | **38.5** | **34.8** | **49.9** | **37.1** |
| EaTA Worst | 57.7 | 54.0 | 66.5 | 40.6 | 53.2 | 41.4 | 36.8 | 44.0 | 43.5 | 45.4 | 34.8 | 45.4 | 45.7 | 39.9 | 55.7 | 47.0 |
| EaTA Best | 48.1 | 44.7 | 57.9 | 37.1 | 44.1 | 38.7 | 34.9 | 33.7 | 31.9 | 31.6 | 33.2 | 37.2 | 40.0 | 34.7 | 50.3 | 39.9 |
| EaTA + CONTRAST | **43.3** | **40.7** | **54.3** | **27.5** | **39.4** | **30.4** | **27.5** | **29.2** | **29.1** | **28.3** | **25.9** | **31.3** | **33.4** | **29.0** | **43.1** | **34.2** |
| CoTTA Worst | 59.2 | 57.4 | 68.0 | 40.1 | 52.7 | 42.1 | 40.5 | 47.0 | 46.6 | 47.2 | 39.4 | 43.6 | 44.5 | 41.4 | 47.4 | 47.8 |
| CoTTA Best | 49.8 | 46.6 | 58.6 | 34.0 | 40.7 | 36.5 | 34.2 | 34.2 | 32.8 | 33.0 | 32.8 | 34.8 | 35.3 | 33.6 | 41.1 | 38.5 |
| CoTTA + CONTRAST | **44.6** | **43.8** | **57.2** | **27.8** | **37.6** | **30.6** | **28.0** | **29.3** | **29.3** | **28.2** | **26.6** | **30.0** | **32.5** | **29.7** | **41.4** | **34.4** |

final errors are remarkably close: 37.1% for Tent, 34.2% for EaTA, and 36.9% for CoTTA. This suggests that even though Tent is more lightweight and faster compared to the other methods and is not inherently designed to handle forgetting, its performance within our framework is on par with the results obtained when incorporating the other two methods designed to prevent forgetting. This shows the generalizability of our approach to various single-source methods.

**Analysis of Forgetting.** Here, we demonstrate the robustness of our method against catastrophic forgetting by evaluating the classification accuracy on the source test set after completing adaptation to each domain [126, 162, 18]. For CONTRAST, we use our ensembling method to adapt to the incoming domain. After adaptation, we infer each of the

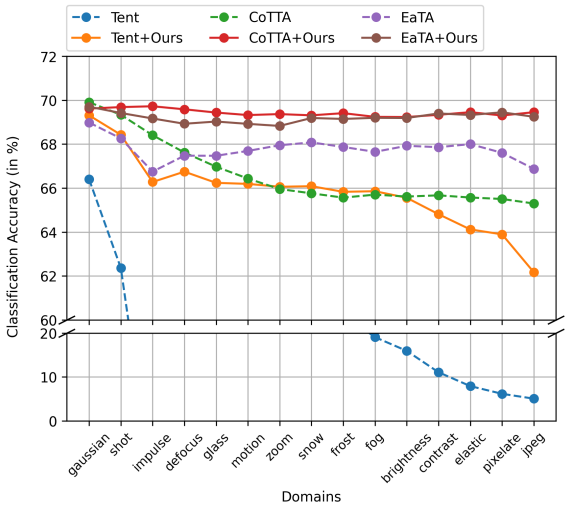



Figure 3.3: **Comparison with baselines in terms of source knowledge forgetting.** Maintaining the same setting as in Table 3.1, we demonstrate that by integrating single-source methods with CONTRAST, the source knowledge is better preserved during dynamic adaptation. Unlike all these single-source methods, our algorithm demonstrates virtually no forgetting throughout the entire adaptation process.

adapted source models on its corresponding source test set. For the baseline single-source methods, every model is adapted individually to the incoming domain, followed by inference on its corresponding source test set. The reported accuracy represents the average accuracy obtained from each of these single-source adapted models.

From Figure 3.3, we note that our method consistently maintains its source accuracy during the adaptation process across the 15 sequential noises. In contrast, the accuracy for each individual single-source method (X) declines on the source test set as the adaptation process progresses. Specifically, Tent, not being crafted to alleviate forgetting, experiences a sharp decline in accuracy. While CoTTA and EaTA exhibit forgetting, it occurs at a more gradual pace. Contrary to all of these single-source methods, our algorithm exhibits virtually no forgetting throughout the process.

**Ablation Study.** We conduct an ablation study in Table 3.5, 3.6 in the Appendix-2 to evaluate the impact of various initialization and learning rate strategies on the optimization process described in (3.4). Our findings demonstrate that the initialization and learning rate configurations generated by our method outperform other alternatives. Additionally, our experiments in Table 3.7, 3.8 and 3.9 in the Appendix-2 reveal that selectively updating the most correlated model parameters enhances performance compared to updating all model parameters, the least correlated ones, a selected subset of correlated models or even updating the models according to their combination weights. We report the comparison with MSDA in Table 3.10 and Model-Soups in Table 3.11. We also report the values of the combination weights learned by our method. **See Subection 3.6 of the Appendix-2 for detailed observations.**

## 3.4 Conclusions

We propose a novel framework called CONTRAST, that effectively combines multiple source models during test time with small batches of streaming data and without access to the source data. It achieves a test accuracy that is at least as good as the best individual source model. In addition, the design of CONTRAST offers the added benefit of naturally preventing the issue of catastrophic forgetting. To validate the effectiveness of our algorithm, we conduct experiments on a diverse range of benchmark datasets.

## 3.5 Appendix-2

## Appendix Overview:

- **Section 3.5.1**: Algorithm of CONTRAST
- **Section 3.5.2**: Proof of Theorem 1 and 2
- **Section 3.5.3**: Results on Digits
- **Section 3.5.4**: Results on Office-Home
- **Section 3.5.5**: Results on CIFAR-10C
- **Section 3.6**: Ablation Study
- **Section 3.6.6**: Implementation Details
- **Section 3.6.9**: Semantic Segmentation
- **Section 3.6.13**: Additional Discussion
- **Section 3.6.14**: KL divergence between two univariate Gaussians
- **Section 3.6.15**: Optimal step size in approximate Newton's method

### 3.5.1 Algorithm

### 3.5.2 Proof and discussion of Theorem 1 and 2

**Proof of Theroem 3.** The optimization (3.4) has a structure similar to a class of non convex problems as follows:

**Algorithm 2** Overview of CONTRAST

**Input:** Pre-trained source models $\{\mathrm{f}_S^j\}_{j=1}^N$, streaming sequential unlabeled test data $\{x_i^{(1)}\}_{i=1}^B \rightarrow \{x_i^{(2)}\}_{i=1}^B \rightarrow \ldots \{x_i^{(t)}\}_{i=1}^B \rightarrow \ldots$

**Output:** Optimal inference model for $t$-th test batch $\mathrm{f}_T^{(t)}$ $\forall t$

**Initialization:** Assign $\mathrm{f}_S^{j(1)} \leftarrow \mathrm{f}_S^j$ $\forall j$

**while** $t \geq 1$ **do**

  Set initial $\mathbf{w}_{init}^{(t)}$ using Eqn. (3.6)

  Set $\alpha_{best}^{(t)}$ using Eqn. (3.7)

  Solve optimization 3.4 to get $\mathbf{w}^{\star(t)}$

  Infer the test batch $t$ using inference model $\mathrm{f}_T^{(t)}$ using Eqn. (3.5)

  Find source index $k$ such that $k = \arg \max_{1 \leq j \leq N} \mathrm{w}_j^{\star(t)}$

  Update source model $\mathrm{f}_S^{k(t)}$ according to Model Parameter Update paragraph of Section 3.2.3 to get $\overline{\mathrm{f}_S^{k(t)}}$

  **for** $1 \leq j \leq N$ **do**

    **if** $j = k$ **then**

      Set $\mathrm{f}_S^{j(t+1)} \leftarrow \overline{\mathrm{f}_S^{j(t)}}$

    **else**

      Set $\mathrm{f}_S^{j(t+1)} \leftarrow \mathrm{f}_S^{j(t)}$

    **end if**

  **end for**

**end while**

$$\underset{x \in \chi}{\text{minimize}} \quad g(x) - h(x)) \tag{3.9}$$

where $\chi$ is a closed convex set, $g(x)$ is $M_g$ smooth and $h(x)$ is a continuous convex function. In such cases, the optimization converges as follows [78]:

$$\frac{1}{(k+1)} \sum_{j=0}^{k} \left( \nabla_\chi \| f(x^k) \|_2^2 \right) \leq \frac{2(f(x^0) - f^\star))}{\alpha(k+1)} \tag{3.10}$$

where, $f(x) = (g(x) - h(x))$. In our case $g(x) = c$, where $c$ is a constant (smooth and continous) and $h(x)$ is negative of the Shannon entropy, which is continous and convex. Also, $\chi$ is the n-simplex $\aleph$, which is a closed convex set. So, according to the proof derived in [78], we can conclude the bound in Theroem 3. ■

**Proof of Theorem 4.** We adapt the theorem from a corollary (corollary 1) in [130]. In this corollary the following result was derived:

$$\mathcal{L}(\mathrm{f}_{\hat{\alpha}}^{\tau}) \leq \min_{\alpha \in \Delta} (l_\star^\alpha(\mathcal{D}) + \mathrm{DM}_{\mathcal{D}'}^{\mathcal{D}}(\alpha) + 4\Gamma \mathcal{R}_{n_\tau}(\mathcal{F}_\alpha)) + \sqrt{\tilde{\mathcal{O}}((h_{eff} + t)/n_\nu)} + \delta$$

Here $\mathrm{f}^\tau$ in the $\mathrm{f}_{\hat{\alpha}}^{\tau}$ is the trained model on the training($\tau$) distribution $\mathcal{D}'$ and $\hat{\alpha}$ is a hyper-parameter that has been empirically optimized by fine tuning on the validation($\nu$) distribution $\mathcal{D}$. $\mathcal{L}$ is the expected risk over the distribution $\mathcal{D}$. DM measures the distribution mismatch via difference of sub-optimality gap using the training and validation distribution. $\mathcal{R}_{n_\tau}(\mathcal{F}_\alpha))$ is the Rademacher complexity of the function class $\mathcal{F}$ with $\alpha$ as the hyper-parameter. The corollary holds for probability of at least $1 - 3e^{-t}$ and $h_{eff}$ is the effective dimension of the hyper-parameter space. Also $n_\nu$ is the number of samples under the validation. The bound can be first of all easily extended to the source/target scenario instead of train/validation. In our scenario the source models jointly construct the function

class $\mathcal{F}_\alpha$ where, the hyper-parameter $\alpha$ is the combination weight w. Effective dimension for our case is exactly the number of source model $N$ and instead of $t$ we took $\tau$ as the probability variable. For the sake of simplicity we omitted $\delta > 0$ which is a positive constant along with the Rademacher complexity. Also $n_\nu = B$ in our setting since we have $B$ number of samples for the target/validation. Now there is a new term in our bound which is $\varphi$ which was not in the original corollary. This term is used to account for the mismatch between actual and pseudo-labels generated by the source. This is done due to the fact that we do empirical minimization of the entropy of the target pseudo-label since the problem is unsupervised and actual labels are not available. The left side of the inequality is derived using the test/target pseudo-label. Consequently, we can introduce an added distribution mismatch term. This term can be broken down into three components: mismatch from target pseudo to target ground truth (gt), from target gt to source gt, and from source gt to source pseudo label. Of these components, the first two can be readily integrated into the $\Psi(.)$ function , given that it measures the discrepancy between the weighted source and the target. The remaining third component is denoted by the $\varphi(.)$ function. This completes the proof. ■

### 3.5.3 Results on Digits

We report here the results of digit classification in Table 3.2. Each column of the table represents a test domain dataset. We train four source models on the rest of the digit datasets. For instance, in case of 'MM' column 'MM' is the test domain which is adapted using four source models trained on 'MT', 'UP', 'SV' and 'SY' respectively.

We calculate the test error of each incoming test batch and then report the num-

Table 3.2: **Results on Digits dataset.** We train the source models using four digit datasets to perform inference on the remaining dataset. The column abbreviations correspond to the datasets as follows: 'MM' for MNIST-M, 'MT' for MNIST, 'UP' for USPS, 'SV' for SVHN, and 'SY' for Synthetic Digits.. The table (reporting % error rate(↓)) shows that X+CONTRAST outperforms all of the baselines (X-Best) consistently .

| | MM | MT | UP | SV | SY | Avg. |
|---|---|---|---|---|---|---|
| Source Worst | 80.5 | 59.4 | 50.3 | 88.5 | 84.8 | 72.7 |
| Source Best | 47.7 | 2.2 | 16.8 | 18.3 | 6.7 | 18.3 |
| Tent Worst | 84.2 | 46.9 | 41.1 | 90.1 | 85.4 | 69.5 |
| Tent Best | 45.2 | 2.3 | 16.7 | 14.4 | 6.7 | 17.1 |
| Tent + CONTRAST | **37.5** | **1.9** | **11.2** | **14.2** | **6.7** | **14.3** |
| EaTA Worst | 80.1 | 48.4 | 42.6 | 88.0 | 83.1 | 68.4 |
| EaTA Best | 47.1 | 2.7 | 18.2 | 18.5 | 7.2 | 18.7 |
| EaTA + CONTRAST | **39.5** | **2.0** | **11.5** | **18.0** | **7.0** | **15.6** |
| CoTTA Worst | 80.0 | 48.3 | 42.8 | 87.9 | 82.9 | 68.4 |
| CoTTA Best | 47.0 | 2.8 | 18.6 | 18.5 | 7.2 | 18.8 |
| CoTTA + CONTRAST | **39.6** | **2.0** | **11.7** | **18.1** | **7.1** | **15.7** |

bers by averaging the error values over all the batches. The table shows that CONTRAST provides a significant reduction of test error compared to the best single source. This demonstrates that when presented with an incoming test batch, CONTRAST has the capability to effectively blend all available sources using optimal weights, resulting in superior performance compared to the best single source (on average 3% error reduction than the

best source). It is important to note that each test batch in this experiment is drawn from the same stationary distribution, which represents the distribution of the target domain. Another baseline exists that simply uses a naive ensemble of the source models, without any weight optimization. In situations where there's a significant performance gap between the best and worst source models adapted using single-source methods, a uniform ensemble of these models produces a predictor that trails considerably behind the best-adapted source, as noted by [6]. Referring to Table 3.2, when testing on the SVHN dataset, the error disparity between the best and worst adapted source models is approximately 70.7%—a substantial margin. Consequently, using a uniform ensemble in such a scenario results in an error rate of roughly 45.5% (experimentally found, not reported in the table). This is strikingly higher than our method's error rate of around 14.2%. *Given these findings, we deduce that uniform ensembling is not a reliable approach for model fusion. Thus, we exclude it from our experiment section's baseline.*

### 3.5.4 Results on Office-Home

Here, we put the results of the experiments on Office-Home. In the Table 3.3, the column refers to the target distribution. Three source models are trained on the rest of the distributions. It can be observed here that X+CONTRAST consistency yields better than best source performance.

### 3.5.5 Results on CIFAR-10C

Note that identical to the experiment on CIFAR100-C in the main chapter 3 the results on CIFAR10-C in Table 3.4 follow the same trend where X+CONTRAST outperforms

Table 3.3: **Results on Office-Home.** We train three source models using three domains in this dataset and use them for inference on the remaining domain under the TTA setting. Our results demonstrate that X+CONTRAST consistently outperforms all of the baselines (X) (% error).

| | Ar | Cl | Pr | Rw | Avg. |
|---|---|---|---|---|---|
| Source Worst | 61.4 | 64.9 | 46.2 | 43.9 | 54.1 |
| Source Best | 42.5 | 58.5 | 29.8 | 35.7 | 41.6 |
| Tent Worst | 57.7 | 60.4 | 46.5 | 42.1 | 51.7 |
| Tent Best | 41.4 | 54.3 | 27.9 | 36.0 | 39.9 |
| Tent + CONTRAST | **40.7** | **52.5** | **27.4** | **27.4** | **37.0** |
| EaTA Worst | 58.4 | 64.3 | 48.0 | 43.5 | 53.5 |
| EaTA Best | 42.1 | 57.8 | 30.3 | 35.9 | 41.5 |
| EaTA + CONTRAST | **40.1** | **53.3** | **28.3** | **28.0** | **37.4** |
| CoTTA Worst | 58.3 | 62.9 | 47.1 | 42.8 | 52.8 |
| CoTTA Best | 42.1 | 55.0 | 29.0 | 34.9 | 40.2 |
| CoTTA + CONTRAST | **40.6** | **53.3** | **28.3** | **29.0** | **37.8** |

the X-Best.

In the single-source scenario, one among the four source models achieves the X-Best (for example CoTTA-Best) accuracy for a specific domain. The determination of which individual model (from the four) will attain the best accuracy for that domain remains uncertain beforehand. Furthermore, the individual source model yielding the X-Best accuracy varies across different domains within CIFAR10-C. However, in our X+CONTRAST ap-

Table 3.4: **Results on CIFAR-10C.** We take four source models trained on *Clear*, *Snow*, *Fog* and *Frost*. We employ these models for adaptation on 15 sequential test domains. This table illustrates that even in the dynamic environment X+CONTRAST performs better than X, which is the direct consequence of better retaining source knowledge. (Results in error rate ↓ (in %))

| | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Worst | 84.7 | 81.1 | 89.1 | 42.6 | 55.6 | 36.2 | 32.2 | 30.6 | 39.2 | 28.7 | 18.5 | 76.4 | 26.9 | 50.0 | 32.7 | 48.3 |
| Source Best | 72.1 | 67.8 | 76.5 | 22.8 | 20.4 | 26.6 | 18.7 | 8.1 | 8.2 | 6.9 | 10.6 | 56.8 | 18.8 | 13.9 | 23.9 | 30.1 |
| Tent Worst | 26.6 | 22.7 | 36.1 | 20.0 | 34.9 | 28.8 | 28.7 | 32.8 | 34.4 | 36.1 | 30.3 | 38.2 | 44.8 | 41.7 | 46.8 | 33.5 |
| Tent Best | 19.3 | 17.6 | 27.9 | 14.5 | 21.1 | 17.6 | 13.5 | 14.3 | 12.6 | 14.4 | 12.4 | 17.0 | 19.0 | 14.3 | 20.4 | 17.1 |
| Tent + CONTRAST | **17.2** | **15.6** | **25.7** | **9.1** | **19.1** | **11.7** | **9.0** | **9.9** | **10.1** | **9.7** | **7.7** | **11.7** | **14.5** | **10.3** | **17.4** | **13.2** |
| EaTA Worst | 31.5 | 30.4 | 44.8 | 14.8 | 33.9 | 16.1 | 13.4 | 20.5 | 21.6 | 19.3 | 11.2 | 18.9 | 23.2 | 19.5 | 29.6 | 23.2 |
| EaTA Best | 21.9 | 20.8 | 33.9 | 10.5 | 19.6 | 14.3 | 10.6 | 8.6 | 9.0 | **7.5** | 8.5 | 10.3 | 16.1 | 11.4 | 24.0 | 15.1 |
| EaTA + CONTRAST | **18.0** | **17.3** | **29.4** | **8.3** | **18.2** | **10.0** | **7.5** | **8.0** | **8.4** | 7.9 | **6.4** | **9.1** | **13.1** | **10.0** | **18.1** | **12.6** |
| CoTTA Worst | 30.1 | 26.8 | 37.8 | 15.0 | 28.5 | 16.6 | 14.6 | 19.3 | 18.6 | 17.5 | 12.2 | 15.9 | 19.4 | 15.4 | 19.3 | 20.5 |
| CoTTA Best | 21.0 | 18.5 | 28.0 | 11.2 | **17.3** | 13.3 | 11.1 | 10.6 | 10.4 | 9.5 | 9.7 | 11.2 | 13.1 | 10.5 | 15.6 | 14.1 |
| CoTTA + CONTRAST | **18.4** | **17.0** | **28.0** | **8.4** | 17.7 | **10.7** | **7.9** | **9.1** | **8.4** | **8.5** | **6.8** | **8.3** | **12.1** | **9.3** | **15.3** | **12.4** |

proach, the need to deliberate over the selection of one out of the four source models is eliminated. X+CONTRAST reliably outperforms any single source X-model that might achieve the X-Best accuracy.

Individual TTA methods may have distinct advantages. For example, Tent offers several distinct advantages over CoTTA, including its lightweight nature and faster performance. Conversely, CoTTA presents certain benefits over Tent, such as increased resilience against forgetting. Consequently, the choice between TTA methods is dependent on the user's preferences, aligning with the specific task at hand. In this experiment, we have demonstrated that CONTRAST can be integrated with any TTA method of the user's choosing.

## 3.6 Ablation Study

### 3.6.1 Initialization and Learning Rate

Table 3.5: **Effect of initialization and step size choice.** Error rate on Office-Home under different choices of initialization and step sizes.

| | **Step size** | | | | | |
|---|---|---|---|---|---|---|
| **Initialization** | $1e-3$ | $1e-2$ | $1e-1$ | $1e0$ | $1e1$ | Ours |
| Random | 40.7 | 40.9 | 40.6 | 39.6 | 41.5 | 39.3 |
| Ours | 37.9 | 37.8 | 37.5 | 37.4 | 39.1 | **37.0** |

Table 3.5 presents the error rate results on the Office-Home dataset under the same experimental setting as Table 3.3 (Appendix) with Tent as the adaptation method, but with different initialization and learning rate choices for solving the optimization in (3.4). It is evident from the table that our chosen initialization and adaptive learning rate result in the highest accuracy gain.

We additionally show another ablation study in Table 3.6, where we initialize the combination weights based on the probability of source model predictions. More precisely, we set the initial weights inversely proportional to the entropy of the source model predictions. In simpler terms, a source model with low entropy receives a higher weight, while one with high entropy receives a lower weight.

In the presented table for CIFAR-100C, we note a 16.5% reduction in error resulting from our initialization method. We found that initializing the combination weights using the entropy of the test batch for various sources leads to somewhat uniform initializa-

Table 3.6: **Initialization based on Entropy.** The table shows the results of entropy based initialization. (Results in error-rate % ↓)

| Update Policy | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Entropy_init | 42.7 | 41.1 | 56.9 | 33.5 | 46.5 | 39.4 | 37.2 | 41.0 | 43.2 | 50.6 | 46.7 | 78.6 | 77.9 | 79.5 | 88.7 | 53.6 |
| Ours | 42.2 | 40.6 | 55.3 | 28.6 | 40.7 | 31.9 | 29.6 | 31.7 | 32.4 | 30.9 | 28.6 | 41.5 | 38.5 | 34.8 | 49.9 | 37.1 |

tion. However, when we initialize the combination weights using KL divergence, we achieve a highly effective and peaky prior, favoring the most correlated source model with relatively higher weightage. This clarifies why initializing with entropies fails to converge quickly to the optimum, resulting in significantly poorer outcomes compared to our method.

### 3.6.2 Model Update Policy

In Table 3.7 and 3.8, we demonstrate that by updating only the model with the highest correlation to the target domain, our method produces the lowest test accuracy. This is in comparison to scenarios where we either update all models or solely the least correlated one. This empirical observation directly supports our theoretical assertion from the theorem: updating the most correlated model is most effective in preventing forgetting, thereby resulting in the smallest test error during gradual adaptation. We also experiment with another model update policy where a subset of model is updated.

#### Subset of Model Update

In this approach, rather than focusing solely on the most correlated source model, we identify and update a subset of source models that exhibit higher correlation than the rest of the models. Specifically, we select models for updating based on their combina-

Table 3.7: **Choice of model update (MeTA+CoTTA).** In our experiments using CoTTA as the model update method on CIFAR100-C, we tested four scenarios: updating all models, updating only the least correlated model, updating subset of model, and updating only the most correlated model. Our results indicate that our model selection approach produces the most favorable outcome. (Results in error rate ↓ (in %))

| Update Policy | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| All Model Update | **44.0** | **42.5** | **54.5** | 30.1 | 38.9 | 33.4 | 31.7 | 32.7 | 32.1 | 32.6 | 30.2 | 32.8 | 34.5 | 32.0 | **40.2** | 36.2 |
| Least Corr. Update | 44.8 | 44.5 | 58.9 | 28.6 | 38.7 | 31.0 | 28.4 | 29.1 | **28.9** | 29.5 | 26.9 | 30.9 | 33.8 | 30.5 | 44.0 | 35.2 |
| Subset of Models Update | 44.5 | 43.3 | 57.1 | 28.1 | **37.5** | 30.6 | 28.4 | 29.9 | 29.9 | 28.8 | 26.8 | 30.2 | **32.4** | 30.2 | 40.4 | 34.5 |
| Most Corr. Update | 44.6 | 43.8 | 57.2 | **27.8** | 37.6 | **30.6** | **28.0** | **29.3** | 29.3 | **28.2** | **26.6** | **30.0** | 32.5 | **29.7** | 41.4 | **34.4** |

Table 3.8: **Choice of model update (CONTRAST+Tent).** In our experiments using Tent as the model update method on CIFAR100-C, we tested four scenarios: updating all models, updating only the least correlated model, updating subset of model, and updating only the most correlated model. Our results indicate that our model selection approach produces the most favorable outcome. (Results in error rate ↓ (in %))

| Update Policy | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| All Model Update | **41.6** | 40.9 | 57.8 | 47.1 | 60.2 | 60.3 | 62.1 | 68.6 | 73.2 | 80.9 | 82.1 | 92.4 | 91.2 | 92.5 | 94.9 | 69.7 |
| Least Corr. Update | 43.8 | 41.4 | 56.1 | 31.2 | 41.4 | 34.8 | 31.4 | 33.5 | 33.1 | 37.5 | 31.5 | 41.6 | 41.5 | 37.5 | 53.1 | 39.3 |
| Subset of Models Update | 43.0 | 41.1 | 56.4 | 33.0 | 47.8 | 38.7 | 37.5 | 41.4 | 45.3 | 51.1 | 46.4 | 83.6 | 81.0 | 60.1 | 92.4 | 53.3 |
| Most Corr. Update | 42.2 | **40.6** | **55.3** | **28.6** | **40.7** | **31.9** | **29.6** | **31.7** | **32.4** | **30.9** | **28.6** | **41.5** | **38.5** | **34.8** | **49.9** | **37.1** |

tion weights, choosing only those whose weights exceed $1/n$, with $n$ representing the total number of models. The intuition behind selecting this threshold $1/n$ for subset selection is grounded in the distance of the combination weight distribution with respect to the uniform distribution. A uniform combination weight implies that all models are equidistant w.r.t the test distribution and should be updated. However, if only one model weight surpasses

$1/n$, it signifies that only one model exhibits a high correlation with the overall model. Results are shown in Table 3.7 and 3.8. Several key observations can be extracted from here. Notably, when utilizing the Tent adaptation algorithm, updating a subset of models results in significantly poorer performance compared to updating only the most correlated model. Conversely, with the CoTTA adaptation algorithm, the performance decrement from updating a subset of models is relatively minor compared to updating the most correlated model. This discrepancy can be attributed to the varying degrees of resistance to forgetting exhibited by these adaptation algorithms. Updating multiple models tends to induce forgetting, leading to a decline in overall performance, especially when the adaptation algorithm is not highly resistant to forgetting. Despite the adaptation method's robustness to forgetting, it has been consistently observed that updating the most correlated model not only delivers superior performance but also offers computational advantages over updating a subset of models. This approach simplifies the update process and ensures more efficient use of computational resources.

**Model Update According to Weight**

Here, we update the model $j$ weighted by $w_j$. To do so, we need to properly devise an approach that updates models in measures according to their correlation with the test data. Drawing inspiration from recent studies that employ variable learning rates for single-source TTA, we devise a strategy to adjust the learning rate $\eta_j$ used in updating model $j$ based on their respective combination weights $w_j$. Specifically, we assigned the highest learning rate $\eta_{max} = 0.001$ (0.001 is the learning rate used for both Tent and CoTTA in our experiments) to the model with the greatest combination weight, while the lowest

learning rate $\eta_{min} = 0.0001$, (a tenfold reduction) was allocated to the model with the lowest combination weight. For the remaining models, we interpolated their learning rates proportionally between the highest and lowest rates, based on their respective combination weights following the formula: $\eta_j = [\left(\frac{w_j - w_{min}}{w_{max} - w_{min}}\right) \times (\eta_{\max} - \eta_{\min})] + \eta_{\min}$. In the Table 3.9, we present the resulting error rates for CIFAR-100C dataset using both Tent and CoTTA.

Table 3.9: **Model Update according to Weight.** The table shows results of updating model according to their respective weights. (Results in error-rate % ↓)

| Update Policy | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tent | 41.7 | 39.7 | 53.0 | 33.9 | 43.9 | 36.8 | 34.6 | 37.8 | 39.3 | 41.0 | 36.8 | 56.1 | 49.5 | 41.4 | 60.1 | 43.0 |
| CONTRAST+Tent | 42.2 | 40.6 | 55.3 | 28.6 | 40.7 | 31.9 | 29.6 | 31.7 | 32.4 | 30.9 | 28.6 | 41.5 | 38.5 | 34.8 | 49.9 | 37.1 |
| CoTTA | 44.5 | 43.0 | 56.2 | 28.1 | 38.1 | 30.8 | 28.6 | 29.9 | 29.6 | 28.7 | 27.0 | 29.5 | 31.8 | 29.0 | 38.6 | 34.2 |
| CONTRAST+CoTTA | 44.6 | 43.8 | 57.2 | 27.8 | 37.6 | 30.6 | 28.0 | 29.3 | 29.3 | 28.2 | 26.6 | 30.0 | 32.5 | 29.7 | 41.4 | 34.4 |

Our investigation reveals that, in scenarios where the update algorithm exhibits limited robustness against forgetting, such as Tent, updating only the model with the highest combination weight proves more advantageous. This is because even marginal updates to uncorrelated models can lead to detrimental forgetting, resulting in poor performance. Conversely, when the update algorithm demonstrates resilience against forgetting (CoTTA), updating the most correlated model impacts performance the most. While updating uncorrelated models does not substantially enhance performance, it significantly increases computational costs. It should also be noted that we have found exactly same finding with our ablation study focused on updating subsets of models. Consequently, we assert that updating the single model with the highest combination weight yields optimal performance across all scenarios.

### 3.6.3 Combination Weight Visualization

To provide insight into the combination weight distribution, let's consider an example where the source models are trained on the clean, snow, frost, and fog domains using the training data. We then select one of these domains to collect the average weights over all the test data. When the test data is from the fog domain, the weight distribution appears as follows: [0.05, 0.08, 0.09, 0.78]. On the other hand, when the test domain is frost, we observe the following weight distribution: [0.07, 0.14, 0.69, 0.11]. These results clearly illustrate that the weight distribution accurately reflects the correlation between the source models and target domains.

### 3.6.4 Comparison with MSDA

Existing multi-source source-free methods are designed for offline settings where all the target data are available during adaptation. However, in our setting, data is received batch by batch during adaptation. Therefore, theoretically, these methods are expected to perform worse in our setup. Nevertheless, we compared CONTRAST with the seminal paper [6] on source-free multi-source Unsupervised Domain Adaptation (UDA), specifically the DECISION method, to demonstrate its effectiveness in an online adaptation setting. We keep the hyperparameters exactly the same as described in the DECISION and perform adaptation on each incoming batch of test data with the number of epochs specified in DECISION.

It is evident from Table 3.10 that DECISION performs notably poorly in the online setting, with an error rate almost 56% higher than CONTRAST. DECISION utilizes

Table 3.10: **Comparison with MSDA.** The table compares the performance of our method with MSDA approach DECISION. (Results in error-rate % ↓)

| Update Policy | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DECISION | 55.0 | 76.2 | 90.5 | 95.2 | 97.3 | 97.9 | 98.2 | 98.0 | 98.3 | 98.4 | 98.4 | 98.7 | 99.0 | 98.9 | 98.9 | 93.3 |
| CONTRAST+Tent | 42.2 | 40.6 | 55.3 | 28.6 | 40.7 | 31.9 | 29.6 | 31.7 | 32.4 | 30.9 | 28.6 | 41.5 | 38.5 | 34.8 | 49.9 | 37.1 |

clustering of the entire offline dataset based on the number of classes, a method not feasible to accurately implement in our setting with very small batch sizes. This highlights the necessity of a multi-source method specifically tailored for our setting.

### 3.6.5 Comparison with Model Soups

Model Soups [193] is a popular approach for utilizing a set of models by averaging their parameters to create a single model for inference on test data. For completeness, we compare our method against Model Soups.

Table 3.11: **Comparison with Model Soups.** The table compares the performance our method against model soups. (Results in error-rate % ↓)

| Update Policy | GN | SN | IN | DB | GB | MB | ZB | Snow | Frost | Fog | Bright | Contrast | Elastic | Pixel | JPEG | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model-Soups | 96.82 | 96.26 | 97.08 | 95.17 | 95.33 | 95.30 | 95.22 | 95.17 | 95.86 | 95.28 | 94.96 | 97.41 | 95.04 | 95.05 | 95.86 | 95.72 |
| CONTRAST+Tent | 42.2 | 40.6 | 55.3 | 28.6 | 40.7 | 31.9 | 29.6 | 31.7 | 32.4 | 30.9 | 28.6 | 41.5 | 38.5 | 34.8 | 49.9 | 37.1 |

As shown in Table 3.11, the performance of Model Soups is significantly worse compared to our method. Model Soups averages the parameters of models fine-tuned on the same data distribution. However, in our setting, we have models trained on different source domains, making the averaging of model parameters suboptimal.

### 3.6.6 Implementation Details

In this section, we provide a comprehensive overview of our experimental setup. We conducted two sets of experiments: one on a stationary target distribution, and the other on a dynamic target distribution that changes continuously. The reported results in the main chapter 3 are average of three runs with different seeds.

### 3.6.7 Stationary Target

#### Digit Classification

The digit classification task consists of five distinct domains from which we construct five different adaptation scenarios. Each scenario involves four source models, with the remaining domain treated as the target distribution. In total, we construct five adaptation scenarios for our study.

The ResNet-18 architecture was used for all models, with an image size of $64 \times 64$ and a batch size of 128 during testing. Mean accuracy over the entire test set is reported in Table 2 of the main chapter 3. For Tent we use a learning rate of 0.01 and for rest of the adaptation method a learning rate of 0.001 is used. We use Adam optimizer for all the adaptation methods. Model parameter update is performed using a single step of gradient descent.

#### Object Recognition

The object recognition task on the Office-Home dataset comprises of four distinct domains from which we construct four different adaptation scenarios, similar to the digit

classification setup. We use the same experimental settings and hyperparameters as the digit classification experiment, with the exception of the image size, which is set to $224 \times 224$ in this experiment. The results of this evaluation are reported in Table 3 of the main chapter 3.

### 3.6.8 Dynamic Target

#### CIFAR-10/100-C

In this experiment, we use four ResNet-18 source models trained on different variants of the CIFAR-10/100 dataset: 1) vanilla train set, 2) train set with added fog (severity = 5), 3) train set with added snow (severity = 5), and 4) train set with added frost (severity = 5). To evaluate the models, we use the test set of CIFAR-10/100C (severity = 5) and adapt to each of the domains in a continual manner. The images are resized to $224 \times 224$. For all the adaptation methods, a learning rate of 0.001 with Adam optimizer is used.

### 3.6.9 Semantic Segmentation

Our method is not just limited to image classification tasks and can be easily extended to other tasks like semantic segmentation (sem-seg). We assume access to a set of sem-seg source models $\{\mathrm{f}_S^j\}_{j=1}^N$, where each model classifies every pixel of an input image to some class. Specifically, $\mathrm{f}_S^j : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{H \times W \times K}$, where $K$ is the number of classes. In this case, the entropy in Eqn. 3 of the main chapter 3 will be modified as follows:

$$\mathcal{L}_w^{(t)}(\mathbf{w}) = -\mathbb{E}_{\mathcal{D}_T^{(t)}} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{K} \hat{y}_{ihwc}^{(t)} \log(\hat{y}_{ihwc}^{(t)}) \tag{3.11}$$

Table 3.12: **Result on Cityscape to ACDC:** In this experiment, we test our method on the test data from individual weather conditions (static test distribution) of ACDC. The source models are trained on the train set of Cityscape and its noisy variants. Our method clearly outperforms baseline adaptation method. (Results in % mIoU)

| **Method** | **Fog** | **Rain** | **Snow** | **Night** | **Avg.** |
|---|---|---|---|---|---|
| Tent-Best | 25.3 | 21.0 | 19.2 | 12.6 | 19.5 |
| CONTRAST | **27.7** | **22.8** | **21.1** | **14.0** | **21.4** |

Table 3.13: **Result on Cityscapes to ACDC for dynamic test distribution:** This table illustrates that over a prolonged cycle of repetitive test distributions, our model can retain performance better than baseline Tent. ((Results in % mIoU))

| Time | t | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Round | 1 | | | | 3 | | | | 5 | | | | All |
| Conditions | Rain | Snow | Fog | Night | Rain | Snow | Fog | Night | Rain | Snow | Fog | Night | Mean |
| Tent-Best | 20.1 | 21.3 | 22.3 | 11.3 | 18.5 | 17.2 | 19.5 | 8.4 | 15.8 | 14.5 | 17.5 | 6.8 | 16.1 |
| CONTRAST | 22.1 | **21.4** | **24.3** | **13.4** | **21.4** | **18.3** | **23.5** | **11.3** | **18.6** | **15.5** | **21.4** | **10.4** | **18.6** |

Where, $\hat{y}_{ihwc}^{(t)}$ is the weighted probability output corresponding to class $c$ for the pixel at location $(h, w)$ at time-stamp $t$. We modify Eqn. 3 in the main chapter 3, while keeping the rest of the framework the same.

### 3.6.10 Datasets

We use the following datasets in our experiments:

• **Cityscapes:** Cityscapes [25] is a large-scale dataset that has dense pixel-level annotations for 30 classes grouped into 8 categories (flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void). There are also fog and rain variants [155, 69] of the Cityscapes dataset, where the clean images of Cityscapes have been simulated to add fog and rainy weather conditions.

• **ACDC:** The Adverse Conditions Dataset [156] has images corresponding to fog, nighttime, rain, and snow weather conditions. Also, the corresponding pixel-level annotations are available. The number of classes is the same as the evaluation classes of the Cityscapes dataset.

### 3.6.11 Experimental setup

We use Deeplab v3+ [21] with a ResNet-18 encoder as the segmentation model for all the experiments. We resize the input images to a size of $512 \times 512$. Following the conventional evaluation protocol [25], we evaluate our model on 19 semantic labels without considering the void label.

We first experiment in a static target distribution setting. Specifically, we train three source models on clean, fog, and rain train splits of Cityscapes. We then evaluate the models on the test set of each of the weather conditions of ACDC dataset using CONTRAST and baseline Tent models. We use a batch size of 16 and report the mean accuracy over all the test batches. Again, we have updated the combination weights of CONTRAST with

SGD optimizer using 5 iterations. For updating the source model in CONTRAST that has the most correlation with the incoming test batch, we use the Adam optimizer with a learning rate of 0.001 and updated the batch-norm parameters with one iteration. The baseline Tent models are also updated with the same optimizer and learning rate. The results in Table. 3.12 clearly demonstrate that CONTRAST outperforms all the baselines on test data from each of the adverse weather conditions.

We also evaluate our method in a dynamic test distribution setting, where we have sequentially incoming test batches from the four weather condition test sets of ACDC dataset. The test sequence includes 5 batches of Rain, followed by 5 batches of Snow, 5 batches of Fog, and finally 5 batches of Night. This sequence is repeated (with the same test images) for a total of 5 rounds. We report the mean accuracy over the 5 batches and include the results for the 1st, 3rd, and 5th rounds in Table 3.13. We use the same hyperparameters as in the dynamic setting of previous experiments with the exception that the batch-size is 16.

### 3.6.12 Visualization

In Fig. 3.4, we present the input images along with the corresponding predicted masks of the baseline models and CONTRAST from the last round. The figure contains rows of input image samples from the four different weather conditions of the ACDC dataset, in the order of rain, snow, fog, and night. CONTRAST is compared with baseline adaptation method Tent, and as shown in Fig. 3.4, it is evident that CONTRAST provides better segmentation results compared to the baselines visually.

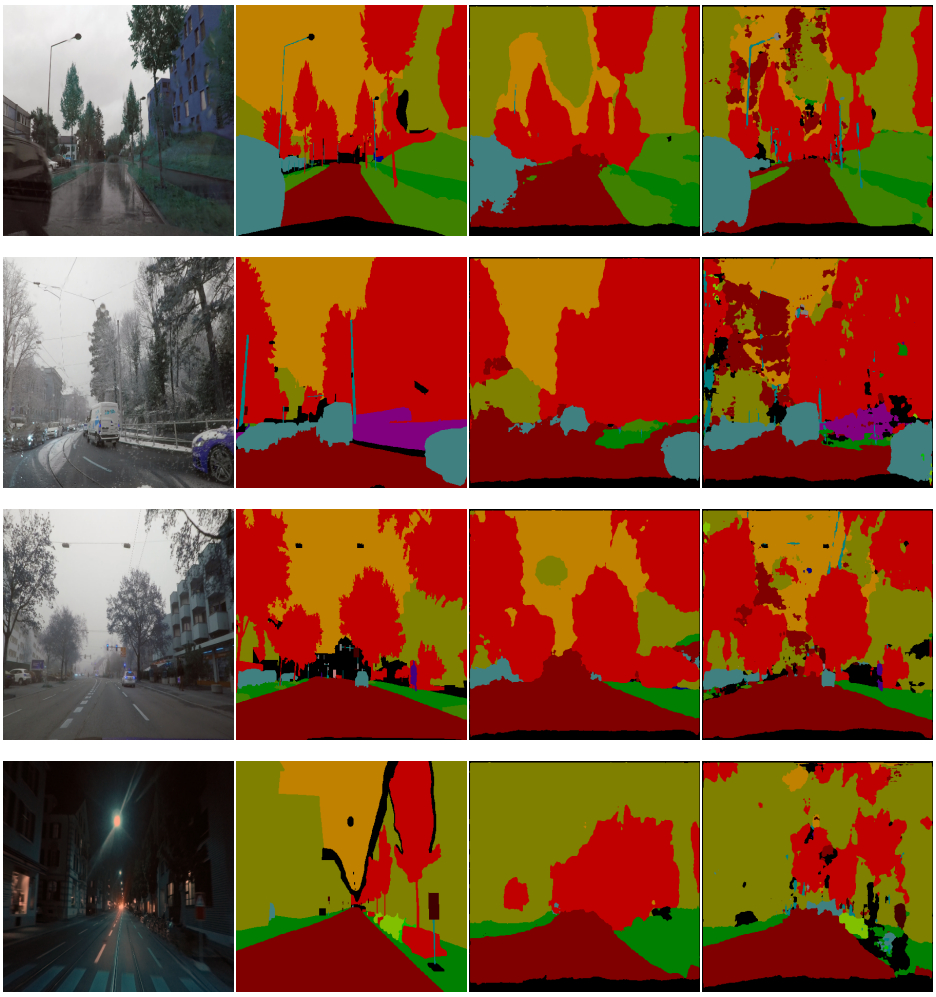



Figure 3.4: **Visual Comparison of CONTRAST with Baselines for Semantic Segmentation Task.** Each row in the figure corresponds to a different weather condition (rain, snow, fog, and night from top to bottom). It is evident that CONTRAST outperforms the baselines in terms of segmentation results.

### 3.6.13 Additional discussion

The $\varphi(.)$ function implies that trained sources should produce high-**quality** pseudo-labels within their own distribution. Essentially, this function evaluates the effectiveness

of the model's training. For instance, even if the **shift** between the source and target is minimal, a poorly trained source model might still under-perform on the target. Observe that both the **shift** and the **quality** terms are minimized when we broaden our search space over $\hat{\Delta}$. This allows us to select a model that exhibits the highest correlation with the test domain, thereby providing us with the most strict bound within the discrete simplex.

Examining the issue through the lens of the gradient provides another perspective. By updating the source model that is most correlated with the test data, its gradient will be smaller than those of other models. Over time, this ensures that the model's parameters remain closer to the original source parameters, thereby preventing catastrophic forgetting. let's examine a toy case mathematically of the most correlated source can give us least gradient.

Let us assume a binary classification task with linear regression where the final activation is sigmoid $\sigma(.)$ function. Now let's take the pseudo-label for a sample $x$ be $\hat{y}$, where $\hat{y} = \sigma(w^\top x)$. Then the entropy $h$ of $\hat{y}$ will be $h = -\hat{y}\log(\hat{y})$. Then we take the derivative of the objective $h$ w.r.t $w$ weight as follows:

$$h = -\hat{y}\log(\hat{y})$$
$$\Rightarrow \frac{\partial h}{\partial w} = (1 + \log(\hat{y}))\hat{y}(\hat{y} - 1)x$$

Now we can easily verify that if the source model is closest to the test domain, then the pseudo-label generated by the model has very small entropy which also means $\hat{y}$ is either close to 0 or close to 1. For both cases the derivative expression above goes close to zero which validate the claim of having smallest gradient for highest correlated source.

### 3.6.14 KL divergence between two univariate Gaussians

During the discussion of initialization of the combination weights in Section 3.5, we come up with $\theta_j^t$ which is calculated using the formula for KL divergence between two univariate Gaussians $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$. In this section, we provide the detailed derivation of this below:

From the definition of KL divergence, we know the distance between two distributions $p$ and $q$ is given by,

$$\begin{aligned} \mathcal{D}_{KL}(p, q) &= \int_{-\infty}^{+\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) \, dx \\ &= \int_{-\infty}^{+\infty} p(x) \log\left(p(x)\right) \, dx - \int_{-\infty}^{+\infty} p(x) \log\left(q(x)\right) \, dx \end{aligned} \tag{3.12}$$

Here in this problem $p$ and $q$ are univariate Gaussians and can be expressed as follows:

$$p(x) = \frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right), \qquad q(x) = \frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right).$$

Now we compute the second term in Eqn. (3.12) as follows:

$$\begin{aligned}
\int_{-\infty}^{+\infty} p(x)\log\left(q(x)\right)\,dx &= \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) - \int_{-\infty}^{+\infty} p(x)\frac{(x-\mu_2)^2}{2\sigma_2^2}\,dx \\
&= \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) - \frac{\int_{-\infty}^{+\infty} x^2 p(x)\,dx - 2\mu_2\int_{-\infty}^{+\infty} xp(x)\,dx + \mu_2^2}{2\sigma_2^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) - \frac{\mathbb{E}\left[X^2\right] - 2\mu_2\mathbb{E}\left[X\right] + \mu_2^2}{2\sigma_2^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) - \frac{\mathrm{Var}\left[X\right] + \left(\mathbb{E}\left[X\right]\right)^2 - 2\mu_2\mathbb{E}\left[X\right] + \mu_2^2}{2\sigma_2^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) - \frac{\sigma_1^2 + \mu_1^2 - 2\mu_2\mu_1 + \mu_2^2}{2\sigma_2^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) - \frac{\sigma_1^2 + (\mu_1-\mu_2)^2}{2\sigma_2^2}
\end{aligned} \tag{3.13}$$

In a similar manner we calculate the first term in Eqn. (3.12) as follows:

$$\begin{aligned}
\int_{-\infty}^{+\infty} p(x)\log\left(p(x)\right)\,dx &= \log\left(\frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}}\right) - \int_{-\infty}^{+\infty} p(x)\frac{(x-\mu_1)^2}{2\sigma_1^2}\,dx \\
&= \log\left(\frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}}\right) - \frac{\int_{-\infty}^{+\infty} x^2 p(x)\,dx - 2\mu_1\int_{-\infty}^{+\infty} xp(x)\,dx + \mu_1^2}{2\sigma_1^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}}\right) - \frac{\mathbb{E}\left[X^2\right] - 2\mu_1\mathbb{E}\left[X\right] + \mu_1^2}{2\sigma_2^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) - \frac{\mathrm{Var}\left[X\right] + \left(\mathbb{E}\left[X\right]\right)^2 - 2\mu_1\mathbb{E}\left[X\right] + \mu_1^2}{2\sigma_1^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}}\right) - \frac{\sigma_1^2 + \mu_1^2 - 2\mu_1^2 + \mu_1^2}{2\sigma_1^2} \\
&= \log\left(\frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}}\right) - \frac{1}{2}
\end{aligned} \tag{3.14}$$

Now combining Eqn. (3.14) and Eqn. (3.13), we get the final KL divergence as follows:

$$\begin{aligned}\mathcal{D}_{KL}(p,q) &= \log\left(\frac{1}{(2\pi\sigma_1^2)^{\frac{1}{2}}}\right) - \frac{1}{2} - \log\left(\frac{1}{(2\pi\sigma_2^2)^{\frac{1}{2}}}\right) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} \\ &= \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}\end{aligned} \tag{3.15}$$

### 3.6.15 Optimal step size in approximate Newton's method

In the main chapter 3, we compute the optimal combination weights by solving the optimization below:

$$\begin{aligned}\underset{\mathbf{w}}{\text{minimize}} \quad & \mathcal{L}_w^{(t)}(\mathbf{w}) \\ \text{subject to} \quad & \mathrm{w}_j \geq 0, \forall j \in \{1, 2, \ldots, N\}, \\ & \sum_{j=1}^{n} \mathrm{w}_j = 1\end{aligned} \tag{3.16}$$

To solve this problem, we begin by initializing $\mathbf{w}_{init}^{(t)}$ as $\delta(-\theta^t)$. Next, we determine the optimal step size based on the initial combination weights to minimize the loss $\mathcal{L}_w^{(t)}$ as much as possible. Specifically, we use a second-order Taylor expansion to approximate the loss at the updated point after taking a single step with a step size of $\alpha^{(t)}$. Thus, after one step of gradient descent, the updated point becomes:

$$\mathbf{w}_{init}^{(t)(1)} = \mathbf{w}_{init}^{(t)} - \alpha^{(t)} \left(\nabla_{\mathbf{w}} \mathcal{L}_w^{(t)}\right) \Big| \mathbf{w}^{init} \tag{3.17}$$

For notational simplicity let us first denote $\mathbf{w}_{init}^{(t)(1)} = \mathbf{w}^{(1)}$, $\mathbf{w}_{init}^{(t)} = \mathbf{w}^{(0)}$ and $\left(\nabla_{\mathbf{w}} \mathcal{L}_w^{(t)}\right) \Big| \mathbf{w}^{init} = \nabla_{\mathbf{w}^{(0)}} \mathcal{L}_w^{(t)}$. We also denote the hessian of $\mathcal{L}_w^{(t)}$ at $\mathbf{w}^{(0)}$ as $\mathcal{H}_{\mathbf{w}^{(0)}}$. Now, we can write the taylor

series expansion of $\mathcal{L}_w^{(t)}$ at $\mathbf{w}^{(1)}$ as follows:

$$\begin{aligned}
&\mathcal{L}_w^{(t)}(\mathbf{w}^{(1)}) = \mathcal{L}_w^{(t)}(\mathbf{w}^{(0)} - \alpha^{(t)}\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}) \\
&= \mathcal{L}_w^{(t)}(\mathbf{w}^{(0)}) - \alpha^{(t)}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right) + \frac{\left(\alpha^{(t)}\right)^2}{2}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\mathcal{H}_{\mathbf{w}^{(0)}}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right) + \mathcal{O}((\alpha^{(t)})^3) \\
&\approx \mathcal{L}_w^{(t)}(\mathbf{w}^{(0)}) - \alpha^{(t)}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right) + \frac{\left(\alpha^{(t)}\right)^2}{2}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\mathcal{H}_{\mathbf{w}^{(0)}}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)
\end{aligned} \tag{3.18}$$

In order to minimize $\mathcal{L}_w^{(t)}(\mathbf{w}^{(1)})$ we differentiate Eqn. (3.18) with respect to $\alpha^{(t)}$ and set it zero to get $\alpha_{best}^{(t)}$. Specifically,

$$\begin{aligned}
&\frac{\partial\mathcal{L}_w^{(t)}(\mathbf{w}^{(1)})}{\partial\alpha^{(t)}}\bigg|_{\alpha^{(t)}=\alpha_{best}^{(t)}} = 0 \\
\implies &-\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right) + \alpha_{best}^{(t)}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\mathcal{H}_{\mathbf{w}^{(0)}}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right) = 0 \\
\implies &\alpha_{best}^{(t)} = \frac{\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)}{\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)^\top\mathcal{H}_{\mathbf{w}^{(0)}}\left(\nabla_{\mathbf{w}^{(0)}}\mathcal{L}_w^{(t)}\right)} = \frac{\left(\nabla_{\mathbf{w}}\mathcal{L}_w^{(t)}\right)^\top\left(\nabla_{\mathbf{w}}\mathcal{L}_w^{(t)}\right)}{\left(\nabla_{\mathbf{w}}\mathcal{L}_w^{(t)}\right)^\top\mathcal{H}_w\left(\nabla_{\mathbf{w}}\mathcal{L}_w^{(t)}\right)}\bigg|_{\mathbf{w}^{init}}
\end{aligned} \tag{3.19}$$

This is the desired expression of $\alpha_{best}^{(t)}$ in Eqn. 3.7 in the main chapter 3.

Note that $\mathbf{w}^{(1)}$ does not lie within the simplex. To ensure that the updated $\mathbf{w}$ remains within the simplex, we project it onto the simplex after each gradient step. This can be done by applying the softmax operator ($\delta(.)$ in the main chapter 3), which will ensure that the updated weights are normalized and satisfy the constraints of the simplex.

# Chapter 4

# Source Free Cross Modal Transfer

## 4.1 Introduction

Depth sensors like Kinect and RealSense, LIDAR for measuring point clouds directly, or high resolution infra-red sensors such as from FLIR, allow for expanding the range of applications of computer vision compared to using only visible wavelengths. Sensing depth directly can provide an approximate three-dimensional picture of the scene and thus improve the performance of applications like autonomous navigation, while sensing in the infra-red wavelengths can allow for easier pedestrian detection or better object detection in adverse atmospheric conditions like rain, fog, and smoke. These are just a few examples.

Building computer vision applications using the now-straightforward supervised deep learning approach for modalities like depth and infrared needs large amounts of diverse labeled data. However, such large and diverse datasets do not exist for these modalities and the cost of building such datasets can be prohibitively high. In such cases, researchers have developed methods like knowledge distillation to transfer the knowledge from a model

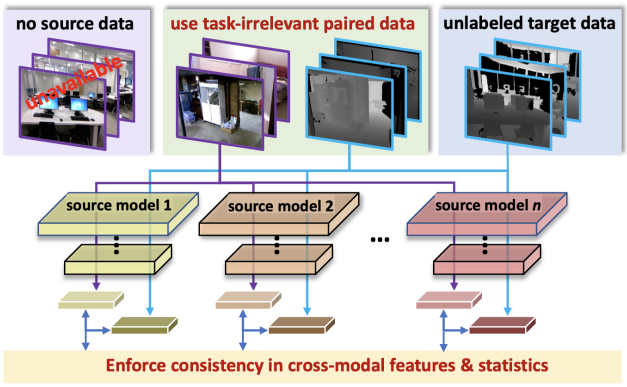



Figure 4.1: **SOCKET:** We describe the problem of single/multi-source cross-modality knowledge transfer using no data used to train the source models. To effectively perform knowledge transfer, we minimize the modality gap by enforcing consistency of cross modal features on **task-irrelevant** paired data in feature space, and by **matching the distributions** of the unlabeled task-relevant features and the source features

trained on a modality like RGB, where large amounts of labeled data are available, to the modality of interest like depth [57].

In contrast to prior work, we tackle a novel and challenging problem in the context of cross-modal knowledge transfer. We assume that we have access only to (a) the source models trained for the task of interest (TOI), and (b) unlabeled data in the target modality where we need to construct a model for the same TOI. The key aspect is that we assume we have **no access to any data in the source modality** for TOI. Such a problem setup is important in cases where memory and privacy considerations do not allow for sharing the training data from the source modality; only the trained models can be shared [4, 102, 6, 142]. We develop **SOCKET**: **SO**urce-free **C**ross-modal **K**nowledg**E** **T**ransfer as an effective solution to this problem for bridging the gap between the source and target

modalities. To this end, we show that employing an external dataset of source-target modality pairs, which are not relevant to TOI – which we call Task-Irrelevant (TI) data – can help in learning an effective target model by bringing the features of the two modalities closer. In addition to using TI data, we encourage matching the statistics of the features of the unlabeled target data – which are Task-Relevant (TR) by definition – with the statistics of the source data which are available to us from the normalization layers that are present in the trained source model.

We provide important empirical evidence showing that the modality-shift from a source modality like RGB to a target modality like depth can be much more challenging than a domain shift from one RGB dataset to another. This shows that the proposed framework is necessary to help minimize the modality gap, so as to make the knowledge transfer more effective. Based on the above ideas, we show that we can improve on existing state-of-the-art methods which were devised only for cross-domain setting in the same modality. We summarize our main contributions below:

1. We formulate a novel problem for knowledge transfer from a model trained for a source modality to a different target modality without any access to task-relevant source data and when the target data is unlabeled.

2. In order to bridge the gap between modalities, we propose a novel framework, SOCKET, for cross-modal knowledge transfer without access to source data (a) using an external task-irrelevant paired dataset, and (b) by matching the moments obtained from the normalization layers in the source models with the moments computed on the target.

3. Extensive experiments on multiple datasets – both for knowledge transfer from RGB

to depth, and from RGB to IR, and both for single-source and multi-source cases – show that SOCKET is useful in reducing the modality gap in the feature space and produces significantly better performance (improvement of as high as 12% for some cases) over the existing source-free domain adaptation baselines which do not account for the modality difference between the source and target modalities.

4. We also show empirically that, for the datasets of interest, the problem of knowledge transfer between modalities like RGB and depth is harder than domain shifts in the same modality such as sensor changes and viewpoint shifts, considered previously in literature.

## 4.2 Related work

**Cross-modal distillation methods.** Cross-modal knowledge distillation (CMKD) methods aim to learn representations for a modality which does not have a large amount of labeled data from a large labeled dataset of another modality [57]. These methods have been used for a variety of practical computer vision and learning tasks [172, 26, 45, 186]. Most of these works assume access to task-relevant paired data across modalities [57, 159, 45, 63]. A recent line of work relaxed this assumption in the context of domain generalization, where one does not have access to the Task-Relevant paired data on the target domain but has access to them for the source domain [230]. There also exist some works regarding domain translation across modalities for better classification of indoor scenes [40, 31, 8]. However these methods consider UDA across domains, where the target domain has unlabeled RGB-D pairs instead of a single modality. All of the above works either utilize the Task-Relevant

paired data for cross modal knowledge transfer [57], or consider cross modal paired data as a domain [230, 40]. There are also works in zero-shot domain adaptation that utilize external task-irrelevant paired data [138] but need access to the source data. Our work takes steps to allow for different source and target modalities, and can perform effective knowledge transfer without access to the TR paired data between source and target.

**Unsupervised domain adaptation methods without source data.** Most UDA methods that have been used for a wide variety of tasks [67, 177, 137, 65] need access to the source data while adapting to a new target domain [44, 139]. To combat the storage or privacy issue regarding the source data, a new line of work named Hypothesis Transfer Learning (HTL) [4, 142] has emerged recently, where one has access only to the trained source model instead of the source data [6, 102]. Here, people have explored adapting target domain data, which has limited labels [4] or no labels at all [102] in the presence of both single source [102, 212, 211] or multiple source models [6]. [102, 103] adapts a single source model to an unlabeled target domain via information maximization and an iterative self-supervised pseudo-label based cross entropy loss. [212] ensured that the adapted source model performs well, both on source and target domains, while [211] proposed a source free domain adaptation (SFDA) method by encouraging label consistency among local target features. [209] proposed to add an extra classifier for refinement of the source decision boundary, while [2] proposed a more robust adaptation method which works well in the presence of noisy pseudo-labels. The authors in [6] proposed fusion of multiple source models with appropriate weights so as to minimize the effect of negative transfer, which we refer to as multiple source free domain adaptation (MSFDA) in Table 4.1. Both these methods do

Table 4.1: We compare the proposed work SOCKET with existing problem settings in literature for knowledge transfer across different domains and modalities. The competitive settings described in this table are: (1) UDA (Unsupervised Domain Adaptation), DT (Domain Translation) [67, 177, 137, 65, 40, 31, 8] [$\mathcal{C}_1$], (2) MSDA (Multi-source domain adaptation) [139] [$\mathcal{C}_2$], (3) SFDA (Source free single source DA) [102, 212, 211, 209, 2, 103] [$\mathcal{C}_3$], (4) MSFDA (Source free multi-source DA) [6] [$\mathcal{C}_4$], (5) CMKD (Cross modal knowledge distillation) [57, 172, 26, 45] [$\mathcal{C}_5$], and (6) ZDDA (Zero shot DA) [138] [$\mathcal{C}_6$], respectively. We group citations into [$\mathcal{C}_1$] to [$\mathcal{C}_6$] based on problem settings. Only SOCKET allows cross-modal knowledge transfer from multiple sources without any access to relevant source training data for an unlabeled target dataset of a different modality

| Property \ Problem setting | UDA+DT [$\mathcal{C}_1$] | MSDA [$\mathcal{C}_2$] | SFDA [$\mathcal{C}_3$] | MSFDA [$\mathcal{C}_4$] | CMKD [$\mathcal{C}_5$] | ZDDA [$\mathcal{C}_6$] | SOCKET |
|---|---|---|---|---|---|---|---|
| Multiple sources | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ |
| No source data | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ |
| Unlabeled target data | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ |
| Different target modality | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Usage of Task-Irrelevant Data | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |

not work well in a regime where the unlabeled target set is from a different modality than the source. We solve this problem by modality gap reduction via feature matching of the task-irrelevant external data, as well as data statistics matching between the source and target modalities.

Table 4.1 summarizes the related work and compares them with SOCKET.

## 4.3 Problem setup and notation

We address the problem of source-data free cross-modality knowledge transfer by devising specialized loss functions that help reduce the gap between source and target modality features. We focus on the task of classification where both the source and target data belong to the same $N$ classes. Let us consider that we have $n$ source models of the same modality (*e.g.*, RGB). We denote the trained source classifiers as $\{\mathcal{F}_{S_k}^{m_S}\}_{k=1}^{n}$ , where $S_k$ denotes the $k$-th source model and $m_S$ represents the modality on which the source models were trained. The source models are denoted as $\mathcal{F}_{S_k}^{m_S}$ which are trained models that map images from the source modality distribution $\mathcal{X}_{S_k}^{m_S}$ to probability distribution over the classes. $\{x_{S_k}^i, y_{S_k}^i\}_{i=1}^{n_k} \sim \mathcal{X}_{S_k}^{m_S}$ are the data on which the $k$-th source model was trained, $n_k$ being the number of training data points corresponding to the $k$-th source. In our problem setting, at the time of knowledge transfer to the target modality, the source data are unavailable for all the sources.

We also have access to an unlabeled dataset in the target modality $\{x_T^i\}_{i=1}^{n_T} \sim \mathcal{X}_T^{m_T}$, where $n_T$ is the number of target samples. Note that the target modality, $m_T$, is different from the source modality. Traditional source free UDA methods try to mitigate domain

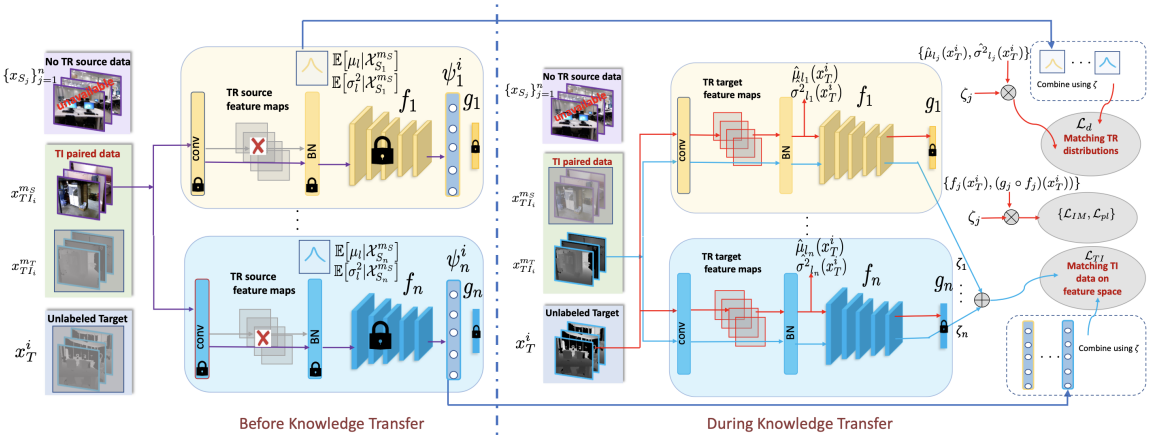



Figure 4.2: **SOCKET description:** Our framework can be split into two parts: (i) Before Knowledge Transfer (left): We freeze the source models and pass the task-irrelevant (TI) source data through the source feature encoders to extract the TI source features. As task-relevant (TR) source feature maps are not available, we extract the stored moments of its distribution from the BN layers. (ii) During Knowledge Transfer (right): We freeze only the classification layers and feed the TI and unlabeled TR target data through the models to get batch-wise TI target features and the TR target moments, respectively, which we match with pre-extracted source features and moments to jointly train all the feature encoders along with the mixing weights, $\zeta_k$'s. The final target model is the optimal linear combination of the updated source models

shift by adapting the source models to unlabeled target data that belong to the same modality [102, 6]. As we will show, applying these methods directly to the cross-modal setting results in poor performance. Hence, we propose to solve this problem using two novel losses as regularization terms which minimize the modality gap between source and target modalities. Our goal is to learn a target classifier $\mathcal{F}_T^{m_T}$, that adapts well on a target distribution obtained from a different sensor modality (*e.g.*, depth or NIR).

To train $\mathcal{F}_T^{m_T}$, we employ (a) methods that enable learning feature embeddings for the target modality that closely match with the source modality embeddings, which we group under modality-specific losses, since it bridges the gap between two different modalities; (b) modality-agnostic loss terms which operate only on the unlabeled target data and do not take into account shift in modality.

We split each of the source models into two blocks – *feature encoder* and *classifier*. For the $k$-th source model, we denote these blocks as $f_k$ and $g_k$, respectively. The function $f_k : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{\eta}$ maps the input image to an $\eta$ dimensional feature vector and $g_k : \mathbb{R}^{\eta} \rightarrow \mathbb{R}^{N}$ maps those features to the probability distributions over the $N$ classes, the maximum of which is treated as the classifier prediction. We can thus write $\mathcal{F}_{S_k}^{m_S} = g_k \circ f_k$ , where "$\circ$" is function composition. Since the classifier layer $g_k$ contains the information about unseen $k$-th source domain distribution, following the protocol of [6], we freeze all the $g_k$'s and train the target specific feature encoders by optimizing over all $f_k$'s.

## 4.4 Cross-Modal Feature Alignment

Traditional source free UDA methods [102, 6] use domain specific but modality-agnostic losses which do not help in reducing the feature distance between the source and target modalities. In order to train the target model, $\mathcal{F}_T^{m_T}$, with reduced modality-gap, we propose SOCKET, which uses *task-irrelevant feature matching* and *task-relevant distribution matching* which are described next.

### 4.4.1 Task-irrelevant feature matching

Capturing the mapping between two modalities effectively requires lots of paired data from both modalities [13]. For our task of interest, we do not have task relevant (TR) data on the source side. As a result, it is not possible to match the target modality with the source modality by using the data from task relevant classes directly. Hence, we propose to use **Task-Irrelevant (TI) paired data** from both modalities to reduce modality gap. TI data contain only classes that are completely **disjoint** from the TR classes and can be from any external dataset. For modalities like RGB-depth and RGB-IR, we can access a large amount of paired TI data that contain classes with no privacy concerns, which are available in public datasets or can be collected using multi-modal sensors. Moreover there are many real world applications where pairwise TI data can be collected and used beyond RGB-D or RGB-IR, such as autonomous driving, adpatation of LiDAR data, medical applications [91]. We denote paired TI data as $\{x_{TI_i}^{m_S}, x_{TI_i}^{m_T}\}_{i=1}^{n_{TI}}$, where $x_{TI_i}^{m_S}$ is the $i$-th TI data point from source modality and $x_{TI_i}^{m_T}$ is its paired counterpart from the target modality, $n_{TI}$ the total number of pairs. We compute our proposed loss $\mathcal{L}_{TI}$ using TI data as follows:

**Step 1:** We feed source modality images of the TI dataset through each of the source models to pre-compute features that are good representations of modality $m_S$. We denote the $i$-th TI source feature extracted from source $j$ as $\psi_j^i$:

$$\psi_j^i = f_j(x_{TI_i}^{m_S}). \tag{4.1}$$

**Step 2:** During the knowledge transfer phase, we feed the target modality images of the TI dataset which are encouraged to match the corresponding pre-extracted source modality features. We do so by minimizing $\mathcal{L}_{TI}$ defined below with respect to the parameters in the feature encoders for the target modality:

$$\mathcal{L}_{TI} = \sum_{i=1}^{n_{TI}} \sum_{j=1}^{n} \left\| \zeta_j(\psi_j^i - f_j(x_{TI_i}^{m_T})) \right\|^2. \tag{4.2}$$

### 4.4.2 Task-relevant distribution matching

In the task-irrelevant feature matching, we match the TI features of two modalities in the feature space. Even if this captures some class independent cross modal mapping between source and target modalities, it has no information about the *TR-class conditional cross modal mapping*. By this term we refer to the cross modal relationship between source and target, given the relevant classes. Assuming that the marginal distribution of the source features across the batches can be modeled as Gaussian, such feature statistics can be fully characterized by its mean and variance. We propose to match the feature statistics across the source and target, to reduce the modality gap further. It might seem as though some amount of source data would be required to estimate the batch-wise mean and and variance of its feature map, but the running average statistics stored in the conventional BatchNorm (BN) layers are good enough to serve our purpose. The BN layers normalize the feature

maps during the course of training to mitigate the covariate shifts [71, 222]. As a result it is able to capture the channel-wise feature statistics cumulatively over all the batches, which gives rise to a rough estimate of the expected mean and variance of the batch-wise feature map, at the end of training. Let us consider that the BN layer corresponding to the $l$-th convolution layer ($\mathcal{B}_l$) has $r_l$ nodes and there exist $b$ number of such layers per source model. Then we refer to the expected batch-wise mean and variance of the $l$-th convolution layer of the $k$-th source model as $\mathbb{E}\big[\mu_l|\mathcal{X}_{S_k}^{m_S}\big] \in \mathbb{R}^{r_l}$ and $\mathbb{E}\big[\sigma_l^2|\mathcal{X}_{S_k}^{m_S}\big] \in \mathbb{R}^{r_l}$. Prior to the start of the knowledge transfer phase, we pre-extract the information about the source feature statistics from all of the pre-trained source models. During the knowledge transfer phase, for each iteration we calculate the batch-wise mean and variance of the feature map of target data from all the source models, linearly combine them according to the weights $\zeta_i$ and minimize the distance of this weighted combination with the weighted combination of the pre-computed source feature statistics. We calculate this loss $\mathcal{L}_d$ given by

$$\mathcal{L}_d = \sum_{l=1}^{b} \left( \left\| \sum_{j=1}^{n} \zeta_j \mathbb{E}\big[\mu_l|\mathcal{X}_{S_j}^{m_S}\big] - \sum_{j=1}^{n} \zeta_j \hat{\mu}_{l_j} \right\| + \left\| \sum_{j=1}^{n} \zeta_j \mathbb{E}\big[\sigma_l^2|\mathcal{X}_{S_j}^{m_S}\big] - \sum_{j=1}^{n} \zeta_j \hat{\sigma^2}_{l_j} \right\| \right), \quad (4.3)$$

where $\mathbb{E}\big[\mu_l|\mathcal{X}_{S_j}^{m_S}\big]$ and $\mathbb{E}\big[\sigma_l^2|\mathcal{X}_{S_j}^{m_S}\big]$ are the running mean and variance of the batchnorm layer corresponding to the $l$-th convolution layer of source $j$, which we refer as $\mathcal{B}_l^j$, and $\hat{\mu}_{l_j} = \frac{1}{n_T}\sum_{k=1}^{n_T} \mathcal{B}_l^j(x_T^k)$ and $\hat{\sigma^2}_{l_j} = \frac{1}{n_T}\sum_{k=1}^{n_T} (\mathcal{B}_l^j(x_T^k) - \hat{\mu}_{l_j})^2$ denote the mean and variance of the target output from the same batchnorm layer. The losses $\mathcal{L}_{TI}$ and $\mathcal{L}_d$ minimize the modality gap between source and target. We name the combination of these two losses as *Modality Specific Loss* $\mathcal{L}_{ms} = \lambda_{TI}\mathcal{L}_{TI} + \lambda_d\mathcal{L}_d$, where $\lambda_{TI}$ and $\lambda_d$ are regularization hyper-parameters.

### 4.4.3 Overall optimization

The two proposed methods above help to reduce the modality gap between source and target without accessing task-relevant source data. In addition to them, we employ the unlabeled target data directly for knowledge transfer. Specifically, we perform *information maximization* along with minimization of a self-supervised *pseudo-label loss*, which have shown promising results in source-free UDA [102, 6] where the source and target modalities are the same.

**Information Maximization (IM)**: IM is essentially the task of performing maximization of the mutual information between distribution of the target data and its labels predicted by the source models. This mutual information is a combination of a conditional and a marginal entropy of the target label distribution.

Motivated by [6], we calculate the *conditional entropy* $\mathcal{L}_{ent}$ and the marginal entropy termed as *diversity* $\mathcal{L}_{div}$ as follows:

$$\mathcal{L}_{ent} = -\frac{1}{n_T}\Big[\sum_{i=1}^{n_T}(\mathcal{F}_T^{m_T}(x_T^i))\log(\mathcal{F}_T^{m_T}(x_T^i))\Big], \mathcal{L}_{div} = -\sum_{j=1}^{N}\bar{p}_j \log \bar{p}_j, \tag{4.4}$$

where $\mathcal{F}_T^{m_T}(x_T^i) = \sum_{k=1}^{n}\zeta_k \mathcal{F}_{S_k}^{m_S}(x_T^i)$, $\zeta_k$ is the weight assigned to the $k$-th source such that $\zeta_k \geq 0$ , $\sum_{k=1}^{n}\zeta_k = 1$ and $\bar{p} = \frac{1}{n_T}\sum_{i=1}^{n_T}\left[\mathcal{F}_T^{m_T}(x_T^i)\right] \in \mathbb{R}^N$ is the empirical label distribution. The *mutual information* is calculated as $\mathcal{L}_{IM} = \mathcal{L}_{div} - \mathcal{L}_{ent}$. Maximization of $\mathcal{L}_{IM}$ (or minimization of $-\mathcal{L}_{IM}$) ensures the target labels, as predicted by the sources, more confident and diverse in nature.

**Pseudo-label loss:** Maximizing $\mathcal{L}_{IM}$ helps to obtain labels that are more confident in prediction and globally diverse. However, that does not prevent mislabeling (*i.e.*, assigning wrong labels to the inputs), which leads to *confirmation bias* [170]. To alleviate this prob-

lem, we adopt a self supervised pseudo-label based cross entropy loss, inspired by [6, 102] (see the Appendix-3 for the exact details about computing the self-supervised pseudo-labels.) After calculating pseudo-labels we compute the *pseudo-label cross entropy* loss $\mathcal{L}_{pl}$ as follows:

$$\mathcal{L}_{pl} = -\frac{1}{n_T}\sum_{i=1}^{n_T}\sum_{k=1}^{K}\mathbf{1}\{\hat{y}_T^i = k\}\log\left[\mathcal{F}_T^{m_T}(x_T^i)\right]_k, \tag{4.5}$$

where $\hat{y}_T^i$ is the pseudo-label for the $i$-th target data point and $\mathbf{1}\{.\}$ is an indicator function that gives value 1 when the argument is true. Our final loss is the combination of the above two losses. We call this combination *modality agnostic loss* $\mathcal{L}_{ma}$, which is expressed as $\mathcal{L}_{ma} = -\mathcal{L}_{IM} + \lambda_{pl}\mathcal{L}_{pl}$.

We calculate the overall objective function as the sum of *modality agnostic* and *modality specific* losses and optimize Eq. (4.6) using Algorithm 3.

$$\underset{\{f_j\}_{j=1}^n,\,\zeta}{\text{minimize}} \quad \mathcal{L}_{ma} + \mathcal{L}_{ms} \quad \text{s.t.} \sum_{k=1}^{n}\zeta_k = 1, \zeta_k \geq 0 \tag{4.6}$$

## 4.5 Experiments

We first describe the datasets, baselines and experimental details we employ. Next, we show results of single and multi-source cross modal transfer which show the efficacy of our method. In Section 4.5.3 we demonstrate experimentally why source free cross modal is a much harder problem compared to cross domain knowledge transfer. We conclude this section by performing analysis on different hyperparameters.

**Algorithm 3** Algorithm to Solve Eq. 4.6

**Input:** $n$ source models trained on modality $m_S$ $\{\mathcal{F}_{S_k}^{m_S}\}_{k=1}^{n} = \{g_k \circ f_k\}_{k=1}^{n}$, unlabeled target data $\{x_T^i\}_{i=1}^{n_T}$ from modality $m_T$, TI cross-modal pairs $\{x_{TI_i}^{m_S}, x_{TI_i}^{m_T}\}_{i=1}^{n_{TI}}$, mixing weights $\{\zeta_k\}_{k=1}^{n}$, max number of epochs $E$, regularization parameters $\lambda_{TI}$, $\lambda_d$, number of batches $B$

**Output:** Optimal adapted feature encoders $\{f_k^\star\}_{k=1}^{n}$, mixing weights $\{\zeta_k^*\}_{k=1}^{n}$

**Initialization**: Freeze final classification layers $\{g_k\}_{k=1}^{n}$, set $\zeta_k = \frac{1}{n}$ for all $k$

Calculate $\{\psi_j^i\}_{j=1}^{n}$ $\forall i \in [1, 2, \ldots, n_{TI}]$ using Eq. (4.1)

Retrieve $\mathbb{E}[\mu_l|\mathcal{X}_{S_j}]$ and $\mathbb{E}[\sigma_l^2|\mathcal{X}_{S_j}]$ for all $j$ and $l$ as in Section 4.4.2

**Knowledge Transfer Phase**:

**for** *epoch* $= 1$ **to** $E$ **do**

  **for** *iteration* $= 1$ **to** $B$ **do**

    Sample a mini-batch of target data and feed it through each of the source models

    Calculate loss terms in Eq. (4.2), (4.3), (4.4), and (4.5)

    Compute overall objective from Eq. (4.6)

    Update parameters in $\{f_j\}_{j=1}^{n}$ and $\{\zeta_k\}_{k=1}^{n}$ by optimizing Eq. (4.6)

    Make $\zeta$ non-negative by setting $\zeta_k := 1/(1 + e^{-\zeta_k})$

    Normalize $\zeta$ by setting $\zeta_k := \zeta_k / \sum_{i=1}^{n} \zeta_i$

  **end for**

**end for**

Final target model $\mathcal{F}_T^{m_T} = \sum_{k=1}^{n} \zeta_k^\star (g_k \circ f_k^\star)$

### 4.5.1 Datasets, baselines and experimental details

**Datasets:** To show the efficacy of our method we extensively test on publicly available cross-modal datasets. We show results on two RGB-D (RGB and Depth) datasets – SUN RGB-D [163] and DIML RGB+D [23], and the RGB-NIR Scene (RGB and Near Infrared) dataset [14]. We summarize the statistics of the datasets in Table 4.2. In the Appendix-3, we provide examples from each dataset and the list of classes which we use as TI and TR data in our experiments.

1. SUN RGB-D: A scene understanding benchmark dataset which contains 10335 RGB-D image pairs of indoor scenes. The dataset has images acquired from four different sensors named *Kinect version1 (kv1)*, *Kinect version2 (kv2)*, *Intel RealSense* and *Asus Xtion*. We treat these four sensors as four different domains. Out of total 45 classes, 17 common classes are treated as TR classes and the remaining 28 classes as TI classes. To train four source models, one for each domain, we use the RGB images from the TR classes, specific to that particular domain. We treat the TR depth images from each of the domains as the target modality data.

2. DIML RGB+D: This dataset consists of more than 200 indoor/outdoor scenes. We use the smaller sample dataset instead of the full dataset, which has 1500/500 RGB-D pairs for training/testing distributed among 18 scene classes. We split the training pairs into RGB and depth, and treat those two as source and target, respectively. The synchronized RGB-D frames are captured using Kinect v2 and Zed stereo camera [81, 80, 82].

3. RGB-NIR Scene: This dataset consists of 477 images from 9 scene categories captured in RGB and Near-infrared (NIR). The images were captured using separate exposures from modified SLR cameras, using visible and NIR [14]. We perform single source knowledge transfer from RGB to NIR and vice versa for this dataset. For all the datasets, TR/TI split is done according to Table 4.2.

**Baseline Methods:** The problem statement we focus on in this chapter is new and has not been considered in literature before. As such, there is no direct baseline for our method. However, the closest related works are source free cross domain knowledge transfer methods that operates under both single and multi-source cases [102, 212, 211, 209, 2, 103, 6]. SHOT [102] and DECISION [6] are the best-known works on single source and multi-source SFDA and we compare against only these two methods. Unlike SOCKET, neither of these baselines employ strategies to overcome modality differences and use only the modality-agnostic loss $\mathcal{L}_{ma}$ for training the target models. Using scene classification as the task of interest, we will show that SOCKET outperforms these baselines for cross-modal knowledge transfer with no access to task-relevant source data. We provide details about the network architecture in the Appendix-3. We note that there a few more recent works [212, 211, 209, 2, 103] which have shown small improvements over SHOT, and are orthogonal to the ideas in this chapter. Incorporating these improvements for SOCKET as well can be interesting and consider this future work.

**Performing knowledge transfer:** Recall that we initialize the target models with the source weights and the classifier layers are frozen. The weights in the feature encoders and source mixing weight parameters ($\zeta_k$'s) in the case of multi-source are the optimiza-

Table 4.2: Datasets statistics

| | SUN-RGBD | RGB-NIR Scene | DIML |
|---|---|---|---|
| Number of domains | 4 | 1 | 1 |
| Domain names | kv1,kv2,Realsense,Xtion | N/A | N/A |
| # of TR images for source training | 1264,1234,238,2512 | 204 | 527 |
| # of TR unlabeled images | 1264,1234,238,2512 | 204 | 527 |
| Number of TI paired images | 1709 | 153 | 1088 |
| Number of TR & TI classes | 17 & 28 | 6 & 3 | 6 & 12 |
| Modalities | RGB-D | RGB-NIR | RGB-D |

tion parameters. The values of various parameters like the learning rate are given in the Appendix-3.

$\lambda_{pl}$ is set as 0.3 for all the experiments following [6]. For the regularization parameters $\lambda_{TI}$ and $\lambda_d$ of *modality specific* losses, we set them to be equal. We empirically choose those parameters in such a way so as to balance it with the *modality agnostic* losses such that no loss component overpowers the other by a large margin. Empirically we found that a range of $(0.1, 0.5)$ works best. All of the values in this range outperform the baselines and we report the best accuracies amongst those. For images from the modalities other than RGB, which are depth and NIR, we repeat the single-channel images into three-channel images, to be able to feed it through the feature encoders which are initialized from the source models trained on RGB images. We use a batch size of 32 for all of our experiments. We run our method 3 times for all experiments with 3 random seeds in PyTorch [135] and report the average accuracies over those.

Table 4.3: **Results on the SUN RGB-D dataset [163] for the task of single-source cross-modal knowledge transfer from RGB to depth modality without access to task relevant source data.** The rows represent RGB domains on which the source models are trained. The columns represent the knowledge transfer results on the depth domains for three methods – *Unadapted* shows results with unadapted source, SHOT[102] and SOCKET.

| Source RGB \ Target depth | Kinect v1 | | | Kinect v2 | | | Realsense | | | Xtion | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Unadapted | SHOT | SOCKET | Unadapted | SHOT | SOCKET | Unadapted | SHOT | SOCKET | Unadapted | SHOT | SOCKET |
| Kinect v1 | 14.8 | 16.7 | **25.3** | 14.6 | 20.3 | **23.6** | 9.0 | 11.9 | **13.4** | 7.1 | 15.3 | **18.1** |
| Kinect v2 | 4.0 | 12.8 | **13.6** | 17.0 | 29.4 | **35.2** | 10.8 | 19.3 | **22.8** | **10.6** | 7.0 | 8.3 |
| Realsense | 2.0 | 7.9 | **20.3** | 7.1 | 18.4 | **23.5** | 14.7 | 27.4 | **30.0** | 5.1 | 9.5 | **11.8** |
| Xtion | 0.7 | 9.5 | **14.2** | 6.0 | 20.2 | **24.2** | 9.0 | 21.8 | **23.5** | 8.1 | 13.2 | **22.2** |
| Average | 5.4 | 11.7 | **18.4** | 11.2 | 22.1 | **26.6** | 10.9 | 20.1 | **22.4** | 7.7 | 11.3 | **15.1** |

### 4.5.2 Main results

**Results on the SUN RGB-D dataset [163]:** Our method is general enough to deal with any number of sources and we demonstrate both single and multi-source knowledge transfer. In Table 4.3, we show single source RGB to depth results for all of the four domains. Treating the unlabeled depth data of each domain as target, we adapt these using source models trained on RGB data from each of the four domains. It is easily evident from Table 4.3, that for the target domains Kinect V1, Kinect V2, Realsense and Xtion, SOCKET consistently outperforms the baseline by a good margin of 6.7%, 4.5%, 2.3%, and 3.8%, respectively, thus proving the efficacy of SOCKET in a source-free cross modal setting. In some of the cases SOCKET outperforms the baseline by a very large margin, as high as 12.4% (Realsense-RGB to Kinect V1-depth). We show two-source RGB to depth

Table 4.4: **Results on the SUN RGB-D dataset [163] for the task of multiple cross-modal knowledge transfer from RGB to depth modality without access to task relevant source data.** The rows show the six combinations of two trained source models on RGB data from four different domains. The columns represent the knowledge transfer results on the domain specific depth data for *DECISION*[6], the current SOTA for multiple source adaptation without source data, and SOCKET

| Target depth / Source RGB | Kinect v1 | | Kinect v2 | | Realsense | | Xtion | |
|---|---|---|---|---|---|---|---|---|
| | DECISION | SOCKET | DECISION | SOCKET | DECISION | SOCKET | DECISION | SOCKET |
| Kinect v1 + Kinect v2 | 17.9 | **19.5** | 34.2 | **36.6** | 18.8 | **19.8** | 14.6 | **18.0** |
| Kinect v1 + Realsense | 12.6 | **18.0** | 23.3 | **26.8** | 24.3 | **24.7** | 10.9 | **12.2** |
| Kinect v1 + Xtion | 11.7 | **23.9** | 29.6 | **35.7** | 20.3 | **21.1** | 16.7 | **20.0** |
| Kinect v2 + Realsense | 7.4 | **11.7** | 22.7 | **33.1** | 28.4 | **29.4** | 6.9 | **9.1** |
| Kinect v2 + Xtion | 14.8 | **16.2** | 27.0 | **31.0** | **25.4** | 25.0 | 11.6 | **18.3** |
| Realsense + Xtion | 8.3 | **10.7** | 23.1 | **25.2** | 30.1 | **31.5** | 9.5 | **10.8** |
| Average | 12.1 | **16.6** | 26.7 | **31.4** | 24.6 | **25.3** | 11.7 | **14.7** |

adaptation results in Table 4.4. For four domains we get six two-source combinations, each of which is used for adaptation to depth data from all four domains. We see that in this case also, on average SOCKET outperforms the baseline for all four target domains by good margins. SOCKET shows good improvement for some individual cases like (Kinect v1 + Xtion)-RGB to Kinect v1 depth – improvement of 12.2% – and (Kinect v2 + Realsense)-RGB to Kinect v2 depth –improvement of 10.4%.

**Results on the DIML RGB+D dataset [23]:** We performed a single source adaptation experiment (Table 4.5) by restructuring the dataset according to Table 4.2. In Table 4.5, we use the TI data from both the DIML RGB+D as well as SUN RGB-D datasets in

Table 4.5: Classification accuracy (%) on DIML dataset with different TI data

| | Unadapted | SHOT | SOCKET | SOCKET |
|---|---|---|---|---|
| TI data | N/A | N/A | DIML RGB+D | SUN RGB-D |
| RGB→Depth | 26.9 | 41.4 | **46.1** | **53.2** |

Table 4.6: Results on RGB-NIR dataset [14] for the task of single-source cross-modal knowledge transfer from RGB to NIR and vice versa without task-relevant source data

| Setting \ Method | Unadapted | SHOT | SOCKET |
|---|---|---|---|
| RGB → NIR | 84.8 | 86.7 | **90.2** |
| NIR → RGB | 65.2 | 92.2 | **92.7** |

two separate columns, where the TI data of SUN RGB+D is the same that have been used for experiments related to the SUN RGB-D dataset. By doing so, we show that SOCKET can perform well even with TI data from a completely different dataset, and find that SOCKET has a gain of 4.7% and 11.8% over baseline for these two TI data settings, respectively.

**Results on the RGB-NIR scene dataset [14]:** We now show that SOCKET also outperforms baslines when the modalitiies are RGB and NIR using the RGB-NIR dataset. We follow the splits described in Table 4.2. We do experiments on both RGB to NIR and vice versa. The results are given in Table 4.6. For RGB to NIR transfer, SOCKET shows 3.5% improvement, while for NIR to RGB transfer, it shows 0.5% improvement over the competing method.

Table 4.7: **Cross modal vs cross domain knowledge transfer for SUN RGB-D dataset scene classification using SHOT[102]**: (1) The first columm shows the accuracies for RGB to depth transfer within the same domain. (2) The second column is generated by transferring knowledge from one RGB domain to other three RGB domains taking the average of the accuracies

| Source | Cross-Modal | Cross-Domain |
|---|---|---|
| Kinect v1 | 16.7 | 24.5 |
| Kinect v2 | 29.4 | 39.6 |
| Realsense | 27.4 | 29.7 |
| Xtion | 13.2 | 43.1 |
| Average | 21.7 | 34.2 |

### 4.5.3 Cross Modal vs Cross Domain

In order to show the importance of the novel problem we consider, we compare the single-source knowledge transfer results on the SUN RGB-D dataset for modality change vs domain shift in Table 4.7. We use SHOT [102] which is a source-free UDA method for this experiment. All the domain-specific source models are trained on RGB images. For domain shift, the targets are all the RGB images of the remaining 3 domains and we report the average over them. Domain shift involves changes in sensor configuration, viewpoints, etc. For modality change, the target data are depth images from the same domain. The scenes are the same as in the RGB source, except they are captured using the depth sensor. The table clearly shows that the accuracy drops by a large margin of 12.5% when we transfer

Table 4.8: **Ablation of contribution of our proposed novel loss components.** The first accuracy column (a) corresponds to single source adaptation from RGB to depth on *kv2* domain, whereas the second column (b) shows the multi-source adaptation result from *kv1+xtion* to *kv1* domain of SUN RGB-D dataset. We show the accuracy gain over using $\mathcal{L}_{ma}$ only inside the parentheses

| $\mathcal{L}_{ma}$ | $\mathcal{L}_{d}$ | $\mathcal{L}_{TI}$ | (a) accuracy (%) | (b) accuracy (%) |
|---|---|---|---|---|
| ✓ | | | 30.0 | 11.7 |
| ✓ | ✓ | | 31.6 (↑1.6) | 18.3 (↑6.6) |
| ✓ | | ✓ | 34.9 (↑4.9) | 22.6 (↑10.9) |
| ✓ | ✓ | ✓ | **36.3** (↑6.3) | **23.9** (↑12.2) |

knowledge across modalities instead of domains of the same modality. This shows that a cross-modal knowledge transfer is not the same as DA and a framework like SOCKET is necessary to reduce the modality gap.

### 4.5.4 Ablation and sensitivity analysis

**Contribution of loss components**: In Table 4.8, the first row has the result with just the *modality agnostic* loss $\mathcal{L}_{ma}$, whereas second and third row shows the individual effect of our proposed *modality specific* losses along with the $\mathcal{L}_{ma}$. For all cases, SOCKET outperforms the baseline and using both losses in conjunction with $\mathcal{L}_{ma}$ yields best results.

**Effect of number of TI images:** We randomly chose six classes from SUN RGB-D dataset as TI data. Table 4.9 clearly shows that increasing per class samples of TI data results in improving the scene-classification accuracy for RGB to depth transfer on the SUN

Table 4.9: **Left: Effect of number of TI data.** We perform knowledge transfer from Kinect v1 RGB to unlabeled depth data. We use six random TI classes and vary the number of TI images per class from 0 to 60 in steps of 20. **Right: Effect of regularization hyper-parameters**. We perform Kinect v1 and Kinect v2 RGB to Kinect v1 depth transfer with varying $(\lambda_{TI}, \lambda_d)$ and tabulate the accuracy of SOCKET

| Images per class | 60 | 40 | 20 | 0 |
|---|---|---|---|---|
| Accuracy (%) | 25.0 | 22.5 | 20.3 | 16.7 |

| $(\lambda_{TI}, \lambda_d)$ | 0.00 | 0.05 | 0.10 | 0.50 | 1.00 |
|---|---|---|---|---|---|
| Kinect v1 | 16.1 | 15.0 | 16.6 | 23.4 | 21.0 |
| Kinect v2 | 29.3 | 34.2 | 35.0 | 36.7 | 16.3 |

RGB-D dataset. In short, for a fixed number of TI classes, the more TI images per class, the better SOCKET performs.

**Effect of regularization parameters:** In Table 4.9, we observe the effect of test accuracy vs the regularization hyper-parameters for our novel losses proposed as a part of SOCKET. We keep $\lambda_{TI}$ and $\lambda_d$ equal to each other for values between 0 to 1. Using the value of 0 is the same as using SHOT. From the table, we see that as the value of the parameter increases the accuracy also increases up to a certain point, and then it starts decreasing.

## 4.6 Conclusion

We identify the novel and challenging problem of cross-modality knowledge transfer with no access to the task-relevant data from the source sensor modality, and only unlabeled data in the target. We propose our framework, SOCKET, which includes devising loss functions that help bridge the gap between the two modalities in the feature space.

Our results for both RGB-to-depth and RGB-to-NIR experiments show that SOCKET outperforms the baselines which cannot effectively handle modality shift.

## 4.7 Appendix-3

### 4.7.1 Dataset example images

In Figure 4.3 and Figure 4.4 we show some example samples from SUN RGB-D dataset, whereas in Figure 4.5 and Figure 4.6, samples from RGB-NIR scene dataset has been shown. For both datasets, some random samples of Task-Relevant (TR) and and Task-Irrelevant (TI) classes are shown. As DIML dataset has most of the classes overlapped with SUN RGB-D, we do not show examples for that dataset here. For the TR classes, source data are discraded after training the source models and we transfer knowledge from those models to the unlabeled data of target modality. For the TI classes, we have paired samples from both modalities. Note that for all the cases, TR and TI classes are completely disjoint.

### 4.7.2 Calculation of pseudo-labels

For these steps, we mainly follow [6, 102]. We first compute the cluster centroids of all the classes, followed by linearly combining the centroids using the current learned weight vector. We then take each of the weighted features and label it according to it's nearest neighbors from the set of $K$ weighted centroids. In the next step, we update the pseudo labels by repeating these steps. Below, we describe mathematically these steps in detail:

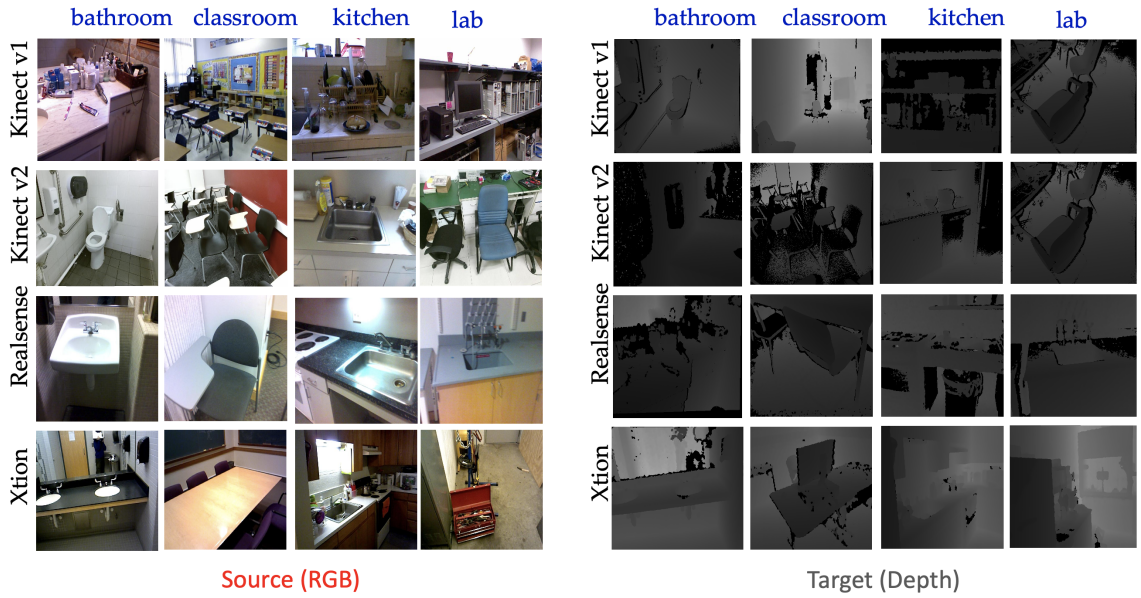



Figure 4.3: **SUN RGB-D TR samples.** We show some example images of the four domains of SUN RGB-D. Both modalities from 4 out of 17 TR classes are shown here. We discard the RGB source data after training four source models and we do not use any label information for the target depth data.

1. We first compute the cluster centroids of all the classes $k \in \{1, 2, \dots N\}$ induced by source $j \in \{1, 2, \dots, n\}$ for the 0-th iteration, by the following equation:

$$c_{k_j}^{(0)} = \frac{\sum_{x_T \in \mathcal{X}_T^{m_T}} \left[\tilde{\mathcal{F}}_{S_j}^{m_S}(x_T)\right]_k \tilde{f}_j(x_T)}{\sum_{x_T \in \mathcal{X}_T^{m_T}} \left[\tilde{\mathcal{F}}_{S_j}^{m_S}(x_T)\right]_k} \tag{4.7}$$

   where $\left[.\right]_k$ indicates the $k$-th element of the vector in argument , $\tilde{f}_j$ denotes the $j$-th source model's feature extractor and $\tilde{\mathcal{F}}_{S_j}^{m_S} = g_j \circ \tilde{f}_j$ represents the complete $j$-th source model from the last iteration.

2. In the next step, we linearly combine these centroids as well as the target features extracted from all the source models from last iteration, with the current learned


Figure 4.4: **SUN RGB-D TI samples.** We show some example images of the TI data from SUN RGB-D dataset. Six classes, each with paired example of RGB and depth are shown here. The TR and TI classes are completely disjoint.

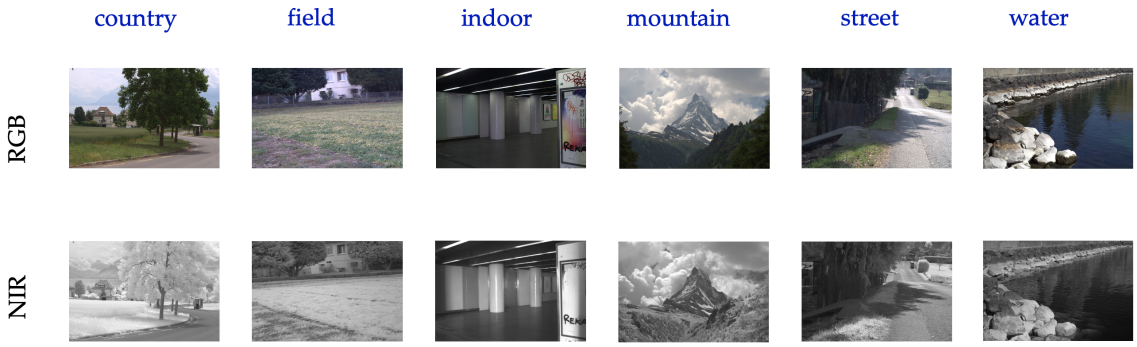



Figure 4.5: **RGB-NIR scene samples.** We show some example images of the of RGB-NIR scene dataset. Both modalities of all 6 TR classes are shown here. We discard the source data after training the source model and we do not use any label information for the target data.

weight vector $\zeta$ as follows:

$$c_k^{(0)} = \sum_{j=1}^{n} \zeta_j c_{k_j}^{(0)} \tag{4.8}$$

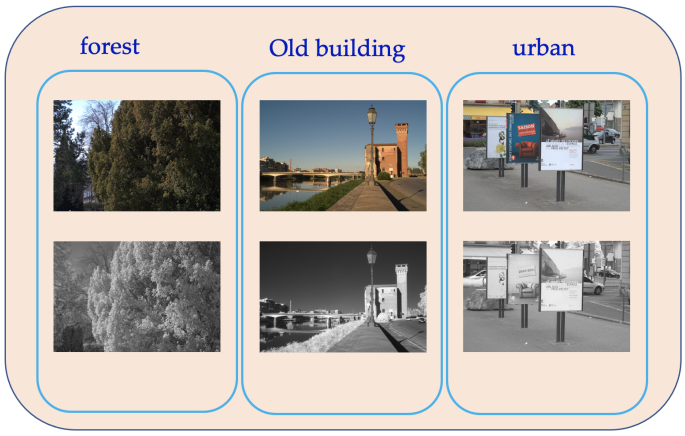



Figure 4.6: **RGB-NIR scene samples** We show some example images of the TI data from RGB-NIR scene dataset. Three classes, each with paired example of RGB and NIR are shown here. The TR and TI classes are completely disjoint.

$$\bar{x}_T = \sum_{j=1}^{n} \zeta_j \tilde{f}_j(x_T) \tag{4.9}$$

3. We take each of the weighted features and label it according to it's nearest neighbour from the set of $K$ weighted centroids, *i.e.*, for a particular target feature, if the nearest neighbour is $k$-th centroid, we assign class label $k$ for that particular feature. The assigned pseudo-label $\hat{y}_T^{i(0)}$ for the $i$-th target feature $\bar{x}_T^i$ at iteration 0 is calculated as:

$$\hat{y}_T^{i(0)} = \arg\min_k \|\bar{x}_T^i - c_k^{(0)}\|_2^2 \tag{4.10}$$

4. We update the pseudo-labels in the next iteration by repeating the steps as follows:

$$c_{k_j}^{(1)} = \frac{\sum_{x_T \in \mathcal{X}_T^{m_T}} \mathbf{1}\{\hat{y}_T^{(0)} = k\} \tilde{f}_j(x_T)}{\sum_{x_T \in \mathcal{X}_T^{m_T}} \mathbf{1}\{\hat{y}_T^{(0)} = k\}} \tag{4.11}$$

where, $\mathbf{1}\{.\}$ is an indicator function which takes value 1, when its argument is true.

$$c_k^{(1)} = \sum_{j=1}^{n} \zeta_j c_{k_j}^{(1)} \tag{4.12}$$

$$\hat{y}_T^{i(1)} = \arg\min_k \|\bar{x}_T^i - c_k^{(1)}\|_2^2 \tag{4.13}$$

Following the protocol of [6], we take $\hat{y}_T^{(1)}$ as the final pseudo-label $\hat{y}_T^i$, without further reiteration.

Finally the *pseudo-label cross entropy* loss $\mathcal{L}_{pl}$ is calculated as follows:

$$\mathcal{L}_{pl} = -\frac{1}{n_T} \sum_{i=1}^{n_T} \sum_{k=1}^{K} \mathbf{1}\{\hat{y}_T^i = k\} \log \left[\mathcal{F}_T^{m_T}(x_T^i)\right]_k. \tag{4.14}$$

### 4.7.3 More details about datasets

**SUN RGB-D**[163]: The 17 common scene classes shared among the four domains are *bathroom, classroom, computer room, conference room, corridor, discussion area, home office, idk, kitchen, lab, living room, office, office kitchen, printer room, reception room, rest space, study space.*

The 28 scene classes used as TI data are *basement, bedroom, book store, cafeteria, coffee room, dancing room, dinette, dining area, dining room, exhibition, furniture store, gym, home, study, hotel room, indoor balcony, study space, laundromat, lecture theatre, library, lobby, mail room, music room, office dining, play room, reception, recreation room, stairs, storage room.*

**DIML RGB+D**[23]: The 6 scene classes used as TR data are *bathroom, classroom, computer room, kitchen, corridor, living room.*

The 12 scene classes used as TI data are *bedroom, billiard hall, book store, cafe, church, hospital, laboratory, library, metting room, restaurant, store, warehouse.*

**RGB-NIR Scene**[14]: The 6 scene classes used as TR data are *country, field, indoor, mountain, street, water*. The 3 scene classes used as TI data are *forest, old building, urban*.

### 4.7.4 Effect of regularization parameters

For the single source adaptation results, we empirically observe that, $(\lambda_{TI}, \lambda_d) = (0.5, 0.5), (0.5, 0.5), (0.1, 0.1), (0.5, 0.5)$ yields best result for Kinect v1, Kinect v2, Realsense and Xtion as targets respectively. For the DIML RGB+D datset, the parameters are set to be $(0.5, 0.5)$, whereas for the RGB-NIR scene dataset, it is set as $(0.01, 0.05)$. Note that, for all of the cases this hyper-parameters are chosen to balance the two loss terms. Our method always performs better than the baseline in the range of values of the hyperparameters we tested and are close to the best accuracies reported in the thesis.

### 4.7.5 Network architectures

In our experiments, we take the Resnet50 [58] model pretrained on ImageNet as the backbone architecture for training the source models, the same way as [206, 139, 102]. Following the architectures used in [43, 6], we replace the last fully connected (FC) layer with a bottleneck layer containing 256 units, within which we add a Batch Normalization [72] (BN) layer at the end of the FC layer. A task specific FC layer with weight normalization [157] is added at the end of the bottleneck layer.

### 4.7.6 Training source models

For training the source models, we resize all the source images to $224 \times 224$. Moreover, to increase model robustness, we use smooth labels instead of one-hot encodings

[168, 121] during this procedure. We set the maximum number of training epochs to 20 for all of the sources, irrespective of the datasets. We utilize stochastic gradient descent with a momentum 0.9 and weight decay $10^{-3}$. The learning rates are set to $10^{-3}$ for the feature encoders ($f_k$'s) and $10^{-2}$ for the added bottleneck layer. During adaptation and knowledge transfer to the target modality, a learning scheduler setting similar to [43, 102] $\theta = \theta_0(1 + 10p)^{-\frac{3}{4}}$ is used, where $\theta$ and $\theta_0$ represent the current and initial learning rates and $p$ is a real number between 0 to 1 which captures the training progress. $\theta_0$ is set to be $10^{-3}$ for the feature encoders ($f_k$'s) and $10^{-2}$ for the added bottleneck layers along with the source mixing weight parameters ($\zeta_k$'s). The maximum number of epochs during target adaptation is set to be 15.

### 4.7.7 Knowledge transfer details

During adaptation and knowledge transfer to the target modality, a learning scheduler setting similar to [43, 102] $\theta = \theta_0(1 + 10p)^{-\frac{3}{4}}$ is used, where $\theta$ and $\theta_0$ represents the current and initial learning rates and $p$ is real number between 0 to 1 which captures the training progress. $\theta_0$ is set to be $10^{-3}$ for the feature encoders ($f_k$'s) and $10^{-2}$ for the added bottleneck layers along with the source mixing weight parameters ($\zeta_k$'s). The learning rate decreases exponentially during the course of training. The maximum number of epochs during target adaptation is set to be 15.

### 4.7.8 Modification of our algorithm in presence of TI *unpaired* data

In this section, we explore the scenario of inaccessibility of pairwise cross-modal data for TI classes. In practical scenario, one might not be able to acquire cross modal

paired data. In this case we show that adversarial matching between two cross modal distributions works reasonably well. Inspired from [177], we propose the following loss function in order to align the two cross modal data distributions which are unpaired. For this purpose, we incorporate a discriminator $\mathcal{D}$ in our framework.

Our adversarial loss has two components: (1) *True Discriminator loss* $\mathcal{L}_{TD}$ and (2) *Adversarial Discriminator loss* $\mathcal{L}_{AD}$. The first loss tries to distinguish between source and target, while the second loss is a proxy for the generator part of the well known usual adversarial loss component, which tries to fool the discriminator in such a way, so that it fails to distinguish between source and target domain. The generator is irrelevant in our framework since we are not generating any new samples, rather as a proxy of the generator we use the same discriminator as an adversary in the second loss. In short, the first loss tries to correctly classify the source and target samples, while the second loss tries to do the opposite. Now, we describe the losses mathematically below:

$$\mathcal{L}_{TD} = -\frac{1}{n_{TI}} \sum_{i=1}^{n_{TI}} \left[ \log \mathcal{D}\Big( \sum_{j=1}^{n} \zeta_j \psi_j^i \Big) + \log \Big( 1 - \mathcal{D}\Big( \sum_{j=1}^{n} \zeta_j f_j(x_{TI_i}^{m_T}) \Big) \Big) \right] \tag{4.15}$$

$$\mathcal{L}_{AD} = -\frac{1}{n_{TI}} \sum_{i=1}^{n_{TI}} \left[ \log \mathcal{D}\Big( \sum_{j=1}^{n} \zeta_j f_j(x_{TI_i}^{m_T}) \Big) \right] \tag{4.16}$$

Note that, $\mathcal{L}_{TD}$ is essentially a cross entropy loss computed with source TI labels as 1 and target TI labels as 0, while $\mathcal{L}_{AD}$ is also a cross entropy loss but computed with target TI labels as 1. So, clearly $\mathcal{L}_{AD}$ will try to oppose the loss $\mathcal{L}_{TD}$, so that the source and target features are indistinguishable. So our overall adversarial loss $\mathcal{L}_{adv}$ is calculated as follows:

$$\mathcal{L}_{adv} = \mathcal{L}_{TD} + \lambda_{AD}\mathcal{L}_{AD} \tag{4.17}$$

where $\lambda_{AD}$ is a regularization parameter to balance the two adversarial loss components. In the absence of TI paired data, the overall new objective function $\mathcal{L}_{tot}$ will be

$$\mathcal{L}_{tot} = \mathcal{L}_{ma} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{d}\mathcal{L}_{d} \tag{4.18}$$

To show the effectiveness of this loss, we conduct a small experiment in table 4.10. We transfer knowledge from the kv2 RGB model to unlabeled kv2 depth data. Due to time constraint we just run this algorithm with one random seed. $\lambda_{AD}$ is set to be 10 to give slightly more importance to $\mathcal{L}_{AD}$ compare to $\mathcal{L}_{TD}$, since our ultimate goal is to learn a feature embedding that can not distinguish between source and target. Clearly we see that our new adversarial loss has an increment of almost 2.9% when used with $\mathcal{L}_{ma}$. Though this gain is not as high compare to the case of having paired TI data (see table 8 in main chapter 4), it is still significant and has great potential. This result is intuitively expected and show that even if with unpaired TI data, we can reduce the modality gap in the absence of TR source data. We hypothesize that for the unpaired TI data case, it is possible to reach a certain extent of the level of performance when using paired TI data, by using relatively more amount of unpaired data. We will explore it in detail for the future work.

### 4.7.9 Future work, limitations and potential negative impact

Further studies are required to better understand the effect of amount of TI data and the diversity present in the data on the knowledge transfer results, which will require access to larger and more diverse datasets. Another interesting avenue for future direction

Table 4.10: **Effect of our proposed adversarial loss component.** The accuracy column corresponds to single source adaptation from RGB to depth on *kv2* domain of SUN RGB-D dataset. We show the accuracy gain over using $\mathcal{L}_{ma}$ only inside the parentheses

| $\mathcal{L}_{ma}$ | $\mathcal{L}_{d}$ | $\mathcal{L}_{adv}$ | (a) accuracy (%) |
|---|---|---|---|
| ✓ | | | 31.0 |
| ✓ | | ✓ | 33.9 (↑2.9) |
| ✓ | ✓ | ✓ | **34.2** (↑3.2) |

is applying these ideas to other modalities like point clouds, medical imaging, etc. The work in this chapter is a general method for improving knowledge transfer from a source modality to a target modality with unlabeled data. The impact of this line of research is to make it easier to train networks for modalities and tasks where large amounts of data and labeled data are not available. This may lead to a wider deployment of deep learning for such modalities. For example in applications like person re-identification, one might have access to the source models trained on private IR labeled data, which they can use to adapt RGB unlabeled data using our method, in order to match people across cameras. Thus, these algorithms can of course be good or bad for society depending on the particular application in which these ideas are employed, the bias in the datasets being used etc. This is also in true in general for other source-free DA methods [102, 6].

# Chapter 5

# Camera Insertion in a Re-Id Network

## 5.1 Introduction

Person re-identification (re-id), which addresses the problem of matching people across different cameras, has attracted intense attention in recent years [233, 50, 151]. Much progress has been made in developing a variety of methods to learn features [104, 116, 117] or distance metrics by exploiting unlabeled and/or manually labeled data. Recently, deep learning methods have also shown significant performance improvement on person re-id [3, 99, 167, 171, 213, 234]. However, with the notable exception of [133, 134], most of these works have not yet considered the dynamic nature of a camera network, where new cameras can be introduced at any time to cover a certain related area that is not well-covered by the existing network of cameras. To build a more scalable person re-identification system,

it is very essential to consider the problem of how to on-board new cameras into an existing network with little additional effort.

Let us consider $K$ number of cameras in a network for which we have learned $\binom{K}{2}$ number of optimal pairwise matching metrics, one for each camera pair (see Figure 5.1 for an illustrative example). However, during an operational phase of the system, new camera(s) may be temporarily introduced to collect additional information, which ideally should be integrated with minimal effort. Given newly introduced camera(s), the traditional re-id methods aim to re-learn the pairwise matching metrics using a costly training phase. This is impractical in many situations where the newly added camera(s) need to be operational soon after they are added. In this case, we cannot afford to wait a long time to obtain significant amount of labeled data for learning pairwise metrics, thus, we only have limited labeled data of persons that appear in the entire camera network after addition of the new camera(s).

Recently published works [133, 134] attempt to address the problem of on-boarding new cameras to a network by utilizing old data that were collected in the original camera network, combined with newly collected data in the expanded network, and source metrics to learn new pairwise metrics. They also assume the same set of people in all camera views, including the new camera (i.e., before and after on-boarding new cameras) for measuring the view similarity. However, this is unrealistic in many surveillance scenarios as source camera data may have been lost or not accessible due to privacy concerns. Additionally, new people may appear after the target camera is installed who may or may not have appeared in existing cameras. Motivated by this observation, we pose an important question: *How*

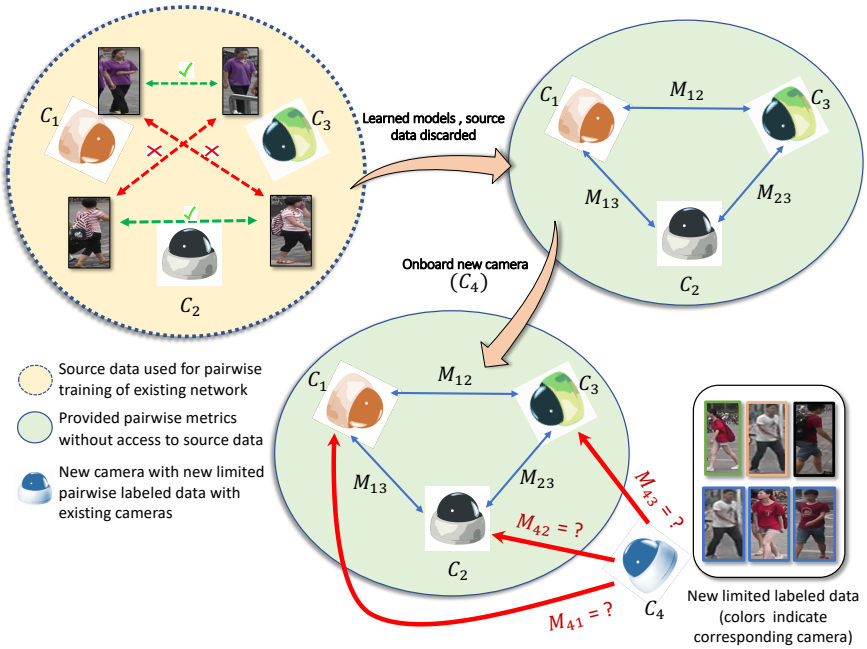



Figure 5.1: Consider a three camera ($C_1$, $C_2$ and $C_3$) network, where we have only three pairwise distance metrics ($M_{12}$, $M_{23}$ and $M_{13}$) available for matching persons, and no access to the labeled data due to privacy concerns. A new camera, $C_4$, needs to be added into the system quickly, thus, allowing us to have only very limited labeled data across the new camera and the existing ones. Our goal in this chapter is to learn the pairwise distance metrics ($M_{41}$, $M_{42}$ and $M_{43}$) between the newly inserted camera(s) and the existing cameras, using the learned source metrics from the existing network and a small amount of labeled data available after installing the new camera(s).

*can we swiftly on-board new camera(s) in an existing re-id framework (i) without having access to the source camera data that the original network was trained on, and (ii) relying upon only a small amount of labeled data during the transient phase, i.e., after adding the new camera(s).*

Transfer learning, which focuses on transferring knowledge from a source to a target domain, has recently been very successful in various computer vision problems [112, 227, 166, 223, 127]. However, knowledge transfer in our system is challenging, because of limited labeled data and absence of source camera data while on-boarding new cameras. To solve these problems, we develop an efficient model adaptation approach using *hypothesis transfer learning* that aims to transfer the knowledge using only source models (i.e., learned metrics) and limited labeled data, but without using any original source camera data. *Only a few labeled identities that are seen by the target camera, and one or more of the source cameras, are needed for effective transfer of source knowledge to the newly introduced target cameras.* Henceforth, we will refer to this as *target data*. Furthermore, unlike [133, 134], which identify only one best source camera that aligns maximally with the target camera, our approach focuses on identifying an optimal weighted combination of multiple source models for transferring the knowledge.

Our approach works as follows. Given a set of pairwise source metrics and limited labeled target data after adding the new camera(s), we develop an efficient convex optimization formulation based on hypothesis transfer learning [92, 32] that minimizes the effect of negative transfer from any outlier source metric while transferring knowledge from source to the target cameras. More specifically, we learn the weights of different source metrics

and the optimal matching metric jointly by alternating minimization, where the weighted source metric is used as a biased regularizer that aids in learning the optimal target metric only using limited labeled data. The proposed method, essentially, learns which camera pairs in the existing source network best describe the environment that is covered by the new camera and one of the existing cameras. Note that our proposed approach can be easily extended to multiple additional cameras being introduced at a time in the network or added sequentially one after another.

### 5.1.1 Contributions

We address the problem of swiftly on-boarding new camera(s) into an existing person re-identification network without having access to the source camera data, and relying upon only a small amount of labeled target data in the transient phase, i.e., after adding the new cameras. Towards solving the problem, we make the following contributions.

- We propose a robust and efficient multiple metric hypothesis transfer learning algorithm to efficiently adapt a newly introduced camera to an existing person re-id framework without having access to the source data.
- We theoretically analyse the properties of our algorithm and show that it minimizes the risk of negative transfer and performs closely to fully supervised case even when a small amount of labeled data is available.
- We perform rigorous experiments on multiple benchmark datasets to show the effectiveness of our proposed approach over existing alternatives.

## 5.2 Related Work

**Person Re-identification.** Most of the methods in person re-id are based on supervised learning. These methods apply extensive training using lots of manually labeled training data, and can be broadly classified in two categories: (i) *Distance metric learning based* [54, 192, 84, 104, 215, 225] (ii) *Deep learning based* [3, 201, 143, 235, 234, 184, 213]. *Distance metric learning based* methods tend to learn distance metrics for camera pairs using pairwise labeled data between those cameras, whereas end-to-end *Deep learning based* methods tend to learn robust feature representations of the persons, taking into consideration all the labeled data across all the cameras at once. To overcome the problem of manual labeling, several unsupervised [225, 187, 112, 226, 208, 107] and semi-supervised [197, 38, 203, 194] methods have been developed over the past decade. However, these methods do not consider the case where new cameras are added to an existing network. The most recent approach in this direction [133, 134] has considered unsupervised domain adaptation of the target camera by making a strong assumption of accessibility of the source data. None of these methods have considered the fact of not having access to the source data in the dynamic camera network setting. This is relevant, as source camera data might have been deleted after a while due to privacy concerns. **Hypothesis Transfer Learning.** Hypothesis transfer learning [207, 113, 128, 92, 32] is a type of transfer learning that uses only the learned classifiers from a source domain to efficiently learn a classifier in the target domain, which contains only limited labeled data. This approach is practically appealing as it does not assume any relationship between source and target distribution, nor the availability of source data, which may be non accessible [92]. Most of the literature has dealt with simple

linear classifiers for transferring knowledge [92, 189]. One recent work [142] has addressed the problem of transferring the knowledge of a source metric, which is a positive semi-definite matrix, with some provable guarantees. However, it has been analyzed for only a single source metric and the weight of the metric is calculated by minimizing a cost function using sub-gradient descent from the generalization bound separately, which is a highly non-convex non-differential function. In [189], the method has addressed transfer of multiple linear classifiers in an SVM framework, where the corresponding weights are calculated jointly with the target classifiers in a single optimization. Unlike these approaches, our approach addresses the case of transfer from multiple source metrics by jointly optimizing for target metric, as well as the source weights to reduce the risk of negative transfer.

## 5.3 Methodology

Let us consider a camera network with $K$ cameras for which we have learned a total $N = \binom{K}{2}$ pairwise metrics using extensive labeled data. We wish to install some new camera(s) in the system that need to be operational soon after they are added, i.e., without collecting and labeling lots of new training data. We do not have access to the old source camera data, rather, we only have the pairwise source distance metrics. Moreover, we also have access to only a limited amount of labeled data across the target and different source cameras, which is collected after installing the new cameras. Using the source metrics and the limited pairwise source-target labeled data, we propose to solve a constrained convex optimization problem (Eq. 5.1) that aims to transfer knowledge from the source metrics to the target efficiently while minimizing the risk of negative transfer.

**Formulation.** Suppose we have access to the optimal distance metric $M_{ab} \in \mathbb{R}^{d\times d}$ for the $a$ and $b$-th camera pair of an existing re-id network, where $d$ is the dimension of the feature representation of the person images and $a, b \in \{1, 2 \dots K\}$. We also have limited pairwise labeled data $\{(x_{ij}, y_{ij})\}_{i=1}^{C}$ between the target camera $\tau$ and the source camera $s$, where $x_{ij} = (x_i - x_j)$ is the feature difference between image $i$ in camera $\tau$ and image $j$ in camera $s$, $C = \binom{n_{\tau s}}{2}$, where $n_{\tau s}$ is the total number of ordered pair images across cameras $\tau$ and $s$, and $y_{ij} \in \{-1, 1\}$. $y_{ij} = 1$ if the persons $i$ and $j$ are the same person across the cameras, and $-1$ otherwise. Note that our approach does not need the presence of every person seen in the new target camera across all the source cameras; rather, it is enough for some people in the target camera to be seen in at least one of the source cameras, in order to compute the new distance metric across source-target pairs. Let $S$ and $D$ be defined as $S = \{(i, j) \mid y_{ij} = 1\}$ and $D = \{(i, j) \mid y_{ij} = -1\}$. Our main goal is to learn the optimal metric between target and each of the source cameras by using the information from all the pairwise source metrics $\{M_j\}_{j=1}^{N}$ and limited labeled data $\{(x_{ij}, y_{ij})\}_{i=1}^{C}$. In standard metric learning context, the distance between two feature vectors $x_i \in \mathbb{R}^d$ and $x_j \in \mathbb{R}^d$ with respect to a metric $M \in \mathbb{R}^{d\times d}$ is calculated by $\sqrt{(x_i - x_j)^\top M (x_i - x_j)}$.

Thus, we formulate the following optimization problem for calculating the optimal metric $M_{\tau s}$ between target camera $\tau$ and the $s$-th source camera, with $n_s$ and $n_d$ number of similar and dissimilar pairs, as follows:

$$\begin{aligned} \underset{M_{\tau s},\, \beta}{\text{minimize}} \quad & \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M_{\tau s} x_{ij} + \lambda \| M_{\tau s} - \sum_{j=1}^{N} \beta_j M_j \|_F^2 \\ \text{subject to} \quad & \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top M_{\tau s} x_{ij}) - b \geq 0,\ M_{\tau s} \succeq 0, \beta \geq 0, \|\beta\|_2 \leq 1 \end{aligned} \tag{5.1}$$

The above objective consists of two main terms. The first term is the normalized sum of distances of all similar pair of features between camera $\tau$ and $s$ with respect to the Mahalanobis metric $M_{\tau s}$, and the second term represents the Frobenius norm of the difference of $M_{\tau s}$ and weighted combination of source metrics squared. $\lambda$ is a regularization parameter to balance the two terms. Note that the second term in Eq. 5.1 is essentially related to hypothesis transfer learning [92, 32] where the hypotheses are the source metrics. The first constraint represents that the normalized sum of distances of all dissimilar pairs of features with respect to $M_{\tau s}$ is greater than a user defined threshold $b$, and the second constraints the distance metrics to always lie in the positive semi-definite cone. While the third constraint keeps all the elements of the source weight vector non-negative, the last constraint ensures that the weights should not deviate much from zero (through upper-bounding the $\ell$-2 norm by 1).

**Notation.** We use the following notations in the optimization steps.

(a) $\mathcal{C}_1 = \{M \in \mathbb{R}^{d\times d} \mid \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top M x_{ij}) - b \geq 0\}$

(b) $\mathcal{C}_2 = \{M \in \mathbb{R}^{d\times d} \mid M \succeq 0\}$

(c) $\mathcal{C}_3 = \{\beta \in \mathbb{R}^N \mid \beta \geq 0 \cap \|\beta\|_2 \leq 1\}$

**Optimization.** The proposed optimization problem (5.1) is jointly convex over $M_{\tau s}$ and $\beta$. To solve this optimization over large size matrices, we devise an iterative algorithm to efficiently solve (5.1) by alternatively solving for two sub-problems. For the sake of brevity, we denote $M_{\tau s}$ as $M$ in the subsequent steps. Specifically, in the first step, we fix the weight $\beta$ and take a gradient step with respect to $M$ in the descent direction with step size

$\alpha$ (Eq. 5.2). Then, we project the updated $M$ onto $\mathcal{C}_1$ and $\mathcal{C}_2$ in an alternating fashion until convergence (Eq. 5.3 and Eq. 5.4). In the next step, we fix the the updated $M$ and take a step with size $\gamma$ towards the direction of negative gradient with respect to $\beta$ (Eq. 5.6). In the last step, we simply project $\beta$ onto the set $\mathcal{C}_3$ (Eq. 5.7).

**Algorithm 4** Algorithm to Solve Eq. 5.1

**Input:** Source metric $\{M_j\}_{j=1}^{N}$, $\{(x_{ij}, y_{ij})\}_{i=1}^{C}$

**Output:** Optimal metric $M^\star$

**Initialization**: $M^k, \beta^k, k = 0$

**while** not converged **do**

  $M^{k+1} = M^k - \alpha \nabla_M f(M, \beta^k)|_{M=M^k}$ Eq. 5.2

  **while** not converged **do**

    $M^{k+1} = \Pi_{\mathcal{C}_1}(M^{k+1})$ Eq. 5.3

    $M^{k+1} = \Pi_{\mathcal{C}_2}(M^{k+1})$ Eq. 5.4

  **end while**

  $\beta^{k+1} = \beta^k - \gamma \nabla_\beta f(M^{k+1}, \beta)|_{\beta=\beta^k}$ Eq. 5.6

  $\beta^{k+1} = \Pi_{\mathcal{C}_3}(\beta^{k+1})$ Eq. 5.7

  $k = k + 1$

**end while**

Algorithm 4 summarizes the alternating minimization procedure to optimize (5.1). We briefly describe these steps below and refer the reader to the Appendix-4 for more mathematical details.

**Step 1: Gradient w.r.t $M$ with fixed $\beta$.** With $k$ being the iteration number and $M^k$,

$\beta^k$ being $M$ and $\beta$ in the $k$-th iteration, we compute the gradient of the objective function (5.1) with respect to $M$ by fixing $\beta = \beta^k$ at the $k$-th iteration as follows:

$$\nabla_M f(M, \beta^k)|_{M=M^k} = \Sigma_S + 2\lambda(M^k - \sum_{j=1}^{N} \beta_j^k M_j), \tag{5.2}$$

where $\Sigma_S = \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij} x_{ij}^\top$ and $f(M, \beta^k) = \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M x_{ij} + \lambda \|M - \sum_{j=1}^{N} \beta_j^k M_j\|_F^2$.

**Step 2: Projection of $M$ onto $\mathcal{C}_1$ and $\mathcal{C}_2$.** The projection of $M$ onto $\mathcal{C}_1$ (denoted as $\Pi_{\mathcal{C}_1}(M)$) can be computed by solving a constrained optimization as follows:

$$\begin{aligned} \Pi_{\mathcal{C}_1}(M) = &\arg\min_{\hat{M}} \quad \frac{1}{2}\|\hat{M} - M\|_F^2 \\ &\text{Subject to} \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top \hat{M} x_{ij}) - b \geq 0 \end{aligned}$$

By writing the Lagrange for the above constrained optimization and using KKT conditions with strong duality, the projection of $M$ onto $\mathcal{C}_1$ can be written as

$$\Pi_{\mathcal{C}_1}(M) = M + \max \left\{ 0, \frac{\left( b - \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij}^\top M x_{ij} \right)}{\|\Sigma_D\|_F^2} \right\} \Sigma_D, \tag{5.3}$$

where $\Sigma_D = \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij} x_{ij}^\top$. Similarly, using spectral value decomposition, the projection of $M$ onto $\mathcal{C}_2$ can be written as

$$\Pi_{\mathcal{C}_2}(M) = V\, diag(\begin{bmatrix} \hat{\lambda_1} & \hat{\lambda_2} \dots \hat{\lambda_n} \end{bmatrix}) V^\top, \tag{5.4}$$

where $V$ is the eigenvector matrix of $M$, $\lambda_i$ is the $i$-th eigenvalue of $M$ and $\hat{\lambda_j} = \max\{\lambda_j, 0\} \quad \forall \quad j \in \begin{bmatrix} 1 \dots d \end{bmatrix}$.

**Step 3: Gradient w.r.t $\beta$ with fixed $M$.** By fixing $M = M^{k+1}$ in the objective function, differentiating it w.r.t $\beta_i$, the $i$-th element of $\beta$ at the point $\beta = \beta^k$, we get

$$\begin{aligned}\nabla_{\beta_i}(f(M^{k+1},\beta))|_{\beta_i=\beta_i^k} = 2\lambda\beta_i^k \text{trace}(M_i^\top M_i) - \\ 2\lambda \text{trace}(M_i^\top (M^{k+1} - \sum_{j=1, j\neq i}^{N} \beta_j^k M_j))\end{aligned} \tag{5.5}$$

By denoting $\nabla_{\beta_i}(f(M^{k+1},\beta))|_{\beta_i=\beta_i^k}$ as $a_i^k$, we get

$$\nabla_{\beta}(f(M^{k+1},\beta))|_{\beta=\beta^k} = \begin{bmatrix} a_1^k & a_2^k & \dots & a_N^k \end{bmatrix}^\top \tag{5.6}$$

**Step 4: Projection of $\beta$ onto $\mathcal{C}_3$.** This step essentially projects a vector to the first quadrant of an $N$-dimensional unit norm hyper-sphere.

The closed form expression of the projection onto $\mathcal{C}_3$ is as follows:

$$\Pi_{\mathcal{C}_3}(\beta^{k+1}) = \max\left\{0, \frac{\beta^{k+1}}{\max\{1, \|\beta^{k+1}\|_2\}}\right\} \tag{5.7}$$

## 5.4 Discussion and Analysis

One of the key differences between our approach and existing methods is that the nature of our problem deals with the multiple metric setting within the hypothesis transfer learning framework. In this section, following [142], we theoretically analyze the properties of our Algorithm 4 for transferring knowledge from multiple metrics.

Let $\mathcal{T}$ be a domain defined over the set $(\mathcal{X} \times \mathcal{Y})$ where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \in \{-1, 1\}$ denote the feature and label set, respectively, and has a probability distribution denoted by $\mathcal{D}_\mathcal{T}$. Let $T$ be the target domain defined by $\{(x_i, y_i)\}_{i=1}^n$ consisting of $n$ i.i.d samples, each

drawn from the distribution $\mathcal{D}_{\mathcal{T}}$. The optimization proposed in Eq.1 of [142] (page. 2) is defined as:

$$\underset{M \succeq 0}{\text{minimize}} \;\; L_T(M) + \lambda \|M - M_S\|_F^2 \tag{5.8}$$

Fixing the value of $\beta$ in our proposed optimization (5.1), we have an optimization problem equivalent to (5.8), where $M_S = \sum_{j=1}^{N} \beta_j M_j$ and

$$L_T(M) = \frac{1}{n_s} \sum_{(i,j) \in S} x_{ij}^\top M x_{ij} + \mu^\star \big(b - \frac{1}{n_d} \sum_{(i,j) \in D} x_{ij}^\top M x_{ij}\big) \tag{5.9}$$

Note that $\mu^\star$ in Eq. 5.9 is the optimal dual variable for the inequality constraint optimization (5.1) with the weight vector fixed. Clearly, the expression is linear, hence convex in $M$, and has a finite Lipschitz constant $k$.

**Theorem 1** *For the convex and k-Lipschitz loss (shown in supp) defined in* (5.9) *the average bound can be expressed as*

$$\mathbb{E}_{T \sim \mathcal{D}_{\mathcal{T}}{}^n}[L_{\mathcal{D}_{\mathcal{T}}}(M^\star)] \leq L_{\mathcal{D}_{\mathcal{T}}}(\widehat{M_S}) + \frac{8k^2}{\lambda n}, \tag{5.10}$$

*where n is the number of target labeled examples, $M^\star$ is the optimal metric computed from Algorithm 4, $\widehat{M_S}$ is the average of all source metrics defined as $\frac{\sum_{j=1}^{N} M_j}{N}$, $\mathbb{E}_{T \sim \mathcal{D}_{\mathcal{T}}{}^n}[L_{\mathcal{D}_{\mathcal{T}}}(M^\star)]$ is the expected loss by $M^\star$ computed over distribution $\mathcal{D}_{\mathcal{T}}$ and $L_{\mathcal{D}_{\mathcal{T}}}(\widehat{M_S})$ is the loss of average of source metrics computed over $\mathcal{D}_{\mathcal{T}}$.*

**Proof.** The proof is given in Appendix-4. ■

**Implication of Theorem** 1: Since we transfer knowledge from multiple source metrics, and do not know which is the most generalizable over the target distribution (i.e., the best source metric), the most sensible thing is to check for the average performance of using

each of the source metrics directly over the target test data. It is equivalently giving all the source metrics equal weights and not using any of the target data for training purpose. The bound in Theorem (5.9) shows that, on average, the metric learned form Algorithm 4 tends to do better than, or in worst case, at least equivalent to the average of source metrics with a fast convergence rate of $\mathcal{O}(\frac{1}{n})$ with limited number of target samples [142].

**Theorem 2** *With probability* $(1-\delta)$*, for any metric* $M$ *learned from Algorithm 4 we have,*

$$L_{\mathcal{D}_\mathcal{T}}(M) \leq L_T(M) + \mathcal{O}\big(\frac{1}{n}\big) + \left(\sqrt{\frac{L_T(\sum_{j=1}^{N}\beta_j M_j)}{\lambda}} + \|\sum_{j=1}^{N}\beta_j M_j\|_F\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \qquad (5.11)$$

*where* $L_{\mathcal{D}_\mathcal{T}}(M)$ *is the loss over the original target distribution (true risk),* $L_T(M)$ *is the loss over the existing target data (empirical risk), and* $n$ *is the number of target samples.*

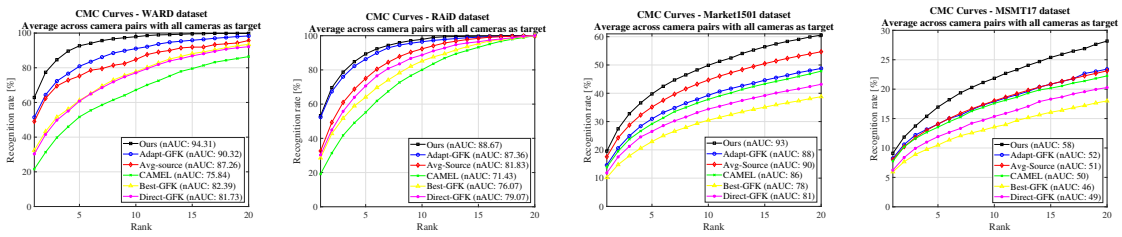
**Proof.** See the Appendix-4 for the proof. ■

**Implication of Theorem** 2: This bound shows that given only a small amount of labeled target data, our method performs closely to the fully supervised case. The right hand side of the inequality (5.11) consists of the term $\mathcal{O}\big(\frac{1}{n}\big) + \Phi(\beta)\mathcal{O}\big(\frac{1}{\sqrt{n}}\big)$. Since the optimal weight $\beta^\star$ from optimization (5.1) will be sparse due to the way $\beta$ is constrained, zero weights will automatically be assigned to the outlier metrics, i.e., outlier $M_j$s, resulting in zero values for the terms $\beta_k^\star L_T(M_j)$ corresponding to those indices $j$ and hence smaller value of $\Phi(\beta)$. As a result, the $\mathcal{O}\big(\frac{1}{\sqrt{n}}\big)$ term will be less dominant in (5.11) than $\mathcal{O}\big(\frac{1}{n}\big)$, due to smaller associated coeffiecient $\Phi(\beta^\star)$ and, hence, can be ignored. Thus, due to the faster decay rate of $\mathcal{O}\big(\frac{1}{n}\big)$, this implies that with very limited target data, the empirical risk will converge to the true risk. Furthermore, when $n$ is very large (the fully supervised case), $\mathcal{O}\big(\frac{1}{\sqrt{n}}\big)$ will be close to zero and cannot be altered by multiplication with any coefficient. This implies

that the source metrics will not have any effect on learning when there is enough labeled target data available and are only useful in the presence of limited data as in our application domain.

**Negative Transfer**: In optimization (5.1), we jointly estimate the optimal metric, as well as the weight vector, which determines which source to transfer from and with how much weight. If a source metric is not a good representative of the target distribution, for an optimal $\lambda$, the weight associated to this metric will automatically be set to zero or close to zero by optimization (5.1), due to the sparsity constraint of $\beta$.

Hence, our approach minimizes the risk of negative transfer.




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

Figure 5.2: CMC curves averaged over all target camera combinations, introduced one at a time. (a) WARD with 3 cameras, (b) RAiD with 4 cameras, (c) Market1501 with 6 cameras and (d) MSMT17 with 15 cameras. Best viewed in color.

## 5.5 Experiments

**Datasets**. We test the effectiveness of our method by experimenting on four publicly available person re-id datasets such as WARD [115], RAiD [27], Market1501 [232], and MSMT17 [191]. There are several other re-id datasets like ViPeR [52], PRID2011 [61]

and CUHK01 [98]; however, those do not apply in our case due to availability of only two cameras. RAiD and WARD are smaller datasets with 43 and 70 persons captured in 4 and 3 cameras, respectively, whereas Market1501 and MSMT17 are more recent and large datasets with 1,501 and 4,101 persons captured across 6 and 15 cameras, respectively.

**Feature Extraction and Matching.** We use Local Maximal Occurrence (LOMO) feature [104] of length $29,960$ in RAiD and WARD datasets. However, since LOMO usually performs poorly on large datasets [50], for Market1501 and MSMT17 we extract features from the last layer of an Imagenet [28] pre-trained ResNet50 network [58] (denoted as IDE features in our work). We follow standard PCA technique to reduce the feature dimension to 100, as in [84, 133].

**Performance Measures.** We provide standard Cumulative Matching Curves (CMC) and normalized Area Under Curve (nAUC), as is common in person re-id [104, 84, 27, 134]. While the former shows accumulated accuracy by considering the $k$-most similar matches within a ranked list, the latter is a measure of re-id accuracy, independent on the number of test samples. Due to the space constraint, we only report average CMC curves for most experiments and leave the full CMC curves in the Appendix-4.

**Experimental Settings.** For RAiD we follow the protocol in [104] and randomly split the persons into a training set of 21 persons and a test set of 20 persons, whereas for WARD, we randomly split the 70 persons into a set of 35 persons for training and rest 35 persons for testing. For both datasets, we perform 10 train/test splits and average accuracy across all splits. We use the standard training and testing splits for both Market1501 and MSMT17 datasets. During testing, we follow a multi-query approach by averaging all query features

of each id in the target camera and compare with all features in the source camera [232].

**Compared Methods.** We compare our approach with the following methods. (1) Two variants of Geodesic Flow Kernel (GFK) [49] such as Direct-GFK where the kernel between a source-target camera pair is directly used to evaluate the accuracy and Best-GFK where GFK between the best source camera and the target camera is used to evaluate accuracy between all source-target camera pairs as in [133, 134]. Both methods use the supervised dimensionality reduction method, Partial Least Squares (PLS), to project features into a low dimensional subspace [133, 134]. (2) State-of-the-art method for on-boarding new cameras [133, 134] that uses transitive inference over the learned GFK across the best source and target camera (Adapt-GFK). (3) Clustering-based Asymmetric MEtric Learning (CAMEL) method of [225], which projects features from source and target camera to a shared space using a learned projection matrix. For all compared methods, we use their publicly available code and perform evaluation in our setting.

### 5.5.1 On-boarding a Single New Camera

We consider one camera as newly introduced target camera and all the other as source cameras with all the possible combinations. In addition to the baselines described above, we compare against the accuracy of average of the source metrics (Avg-Source) by applying it directly over the target test set to prove the validity of Theorem 1. We also compute the GFK kernels in two settings; by considering only target data available after introducing the new cameras(Figure 5.2) and by considering the presence of both old source data and the new labeled data after camera installation as in [133, 134] (Figure 5.3).

**Implementation details.** We split training data into disjoint source and target data considering the fact that the persons that appear in the new camera after installation may or may not be seen before in the source cameras. That is, for Market1501 and MSMT17, we split the training data into 90% of persons that are only seen by the source cameras and 10% that are seen in both source cameras and the new target camera after the installation. Since there are much fewer persons in RAiD and WARD training set, we split the persons into 80% source and 20% target for those two datasets. For each dataset, we evaluate every source-target pair and average accuracy across all pairs. Furthermore, we average accuracy across all cameras as target. Note that the train and test set are kept disjoint in all our experiments.

**Results.** Figure 5.2 and 5.3 show the results. In all cases, our method outperforms all the compared methods. The most competitive methods are those of Adapt-GFK and Avg-Source that also use source metrics. For the remaining methods, we see the limitation of only using limited target data to compute the new metrics. For Market1501, we see that Avg-Source outperforms the Adapt-GFK baseline indicating the advantage of knowledge transfer from multiple source metric compared to one single best source metric as in [133, 134]. However, our approach still outperforms the Avg-Source baseline by a margin of 20.60%, 13.81%, 2.01% and 1.07% in Rank-1 accuracy on RAiD, WARD, Market1501 and MSMT17, respectively, validating our implications of Theorem 1. Furthermore, we observe that even without accessing the source training data that was used for training the network before adding a new camera, our method outperforms the GFK based methods that use all the source data in their computations (see Figure 5.3). To summarize, the experimental results

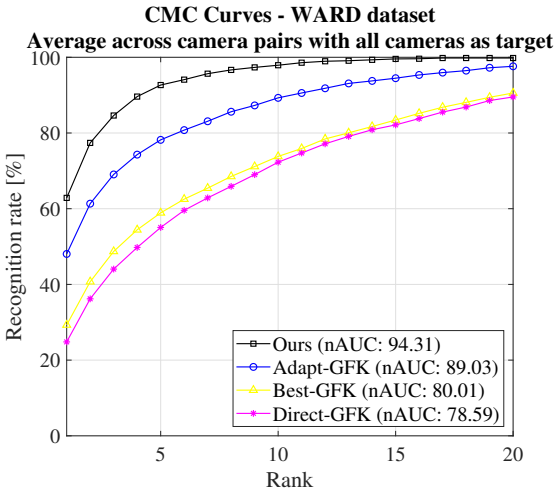


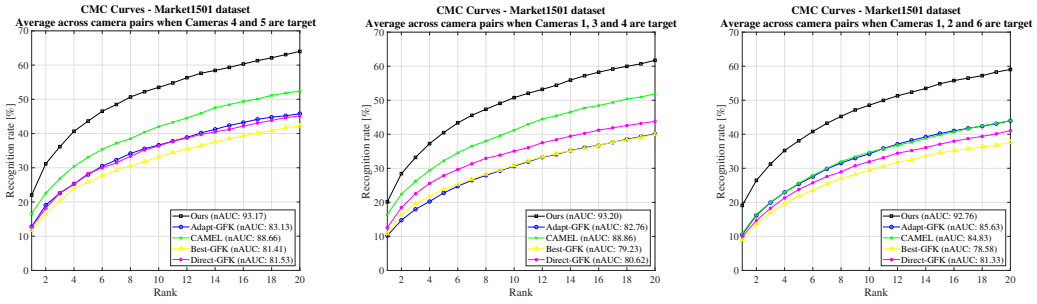

Figure 5.3: CMC curves averaged over all the target camera combinations, introduced one at a time, on the WARD dataset. Note that both old and new source data are used for calculation of GFK. Best viewed in color.

show that our method performs better on both small and large camera networks with limited supervision, as it is able to adapt multiple source metrics through reducing negative transfer by dynamically weighting the source metrics.

### 5.5.2 On-boarding Multiple New Cameras

We perform this experiment on Market1501 dataset using the same strategy as in Section 5.5.1 and compare our results with other methods while adding multiple target cameras to the network, either continuously or in parallel.

**Parallel On-boarding of Cameras:** We randomly select two or three cameras as target while keeping the remaining as source. All the new target cameras are tested against both source cameras and other target cameras. The results of adding two and three cameras in parallel (at the same time) are shown in Figure 5.4 (a) and (b), respectively. In




(a) (b) (c)

Figure 5.4: CMC curves averaged across target cameras on Market1501 dataset. (a) and (b) show results while adding two and three cameras in parallel, (c) show result while adding three cameras sequentially one after another. Best viewed in color.

both cases, our method outperforms all the compared methods with an increasing margin as rank increases. We outperform the most competitive CAMEL in Rank-1 accuracy by 5.45% and 3.73%, while adding two and three cameras respectively. Furthermore, our method better adapts source metrics since it has the capability of assigning zero weights to the metrics that do not generalize well over target data. Meanwhile, Adapt-GFK has a high probability of using the outlier source metrics in the presence of fewer available source metrics, which causes negative transfer. This has been shown in Figure 5.4 where GFK based methods are performing worse than CAMEL, which is computed just with limited supervision without using any source metrics.

**Sequential On-boarding of Cameras:** For this experiment, we randomly select three target cameras that are added sequentially. A target camera is tested against all source cameras and previously added target cameras. The results are shown in Figure 5.4 (c). Similar to parallel on-boarding, our methods outperforms compared methods by a large

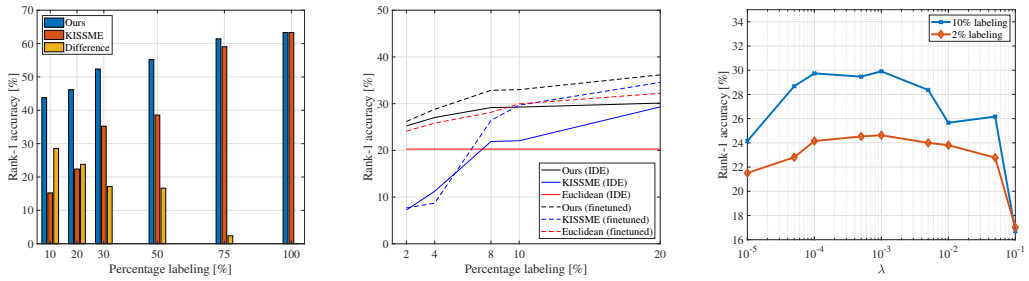
margin. In this setting, we outperform CAMEL by 8.22% in Rank-1 accuracy. Additionally, compared to all GFK-based methods, the Rank-1 margin is kept constant at 10% for both parallel and sequential on-boarding. These results show the scalability of our proposed method while adding multiple cameras to a network, irrespective of whether they are added in parallel or sequentially.

### 5.5.3 Different Labeled Data in New Cameras

We perform this experiment to show the implications of Theorem 2 by using different percentages of labeled target data (10%, 20%, 30%, 50%, 75% and 100%) in our method. We compare with a widely used KISS metric learning (KISSME) [84] algorithm and show the difference in Rank-1 accuracy as a function of labeled target data. Figure 5.5 (a) shows the results. At only 10% labeled data, the difference between our method and KISSME [84] is almost 30%; however, as we add more labeled data, the Rank-1 accuracy becomes equivalent for the two methods at 100% labeled data. This confirms the implications of Theorem 2, where we showed that with increasing labeled target data, the effect of source metrics in learning becomes negligible.

### 5.5.4 Finetuning with Deep Features

This section shows the strength of our method while comparing with CNN features extracted from a network trained on the source data (we train a ResNet50 model [58], pretrained on the Imagenet dataset). Without transfer learning, we have two options: (a) directly use the source model to extract features in the target and do matching based on Euclidean/KISSME metric (IDE), (b) finetune the source model using limited tar-




(a) (b) (c)

Figure 5.5: (a) Effect of different percentage of target labelling on WARD dataset for justifying Theorem 2, (b) Analysis of our method with deep features trained on source camera data in Market1501 dataset with 6th camera as target, (c) Sensitivity of $\lambda$ on the Rank-1 performance tested using deep features in Market1501 with 6th camera as target. Best viewed in color.

get data and then extract features to do matching using Euclidean/KISSME (finetuned). We compared these baselines with our method with different percentage of labeling on Market1501 dataset, where the pairwise metrics are computed using the source features extracted from the model without any finetuning. We use those source metrics along with the target features, extracted before (Ours(IDE)) and after finetuning the source model (Ours(finetuned)). Please see Appendix-4 for more details. Figure 5.5 (b) shows the results. Ours(IDE) outperforms Euclidean(IDE) by a margin of 10% on Market with 20% of labeled target data. The difference between Ours(finetuned) and Euclidean/KISSME (finetuned) is more noticeable with less labeled data and it becomes smaller with increase in labeled target data (Theorem 2). However Ours(finetuned) consistently outperforms all the other baselines for up to 20% labeling.

### 5.5.5 Parameter Sensitivity

We perform this experiment to study the effect of $\lambda$ in optimization (5.1) for a given percentage of labeled target data. Figure 5.5 (c) shows the Rank-1 accuracy of our proposed method for different values of $\lambda$. From optimization 5.1, when $\lambda \to \infty$ the left term can be neglected resulting in optimal $M$ and $\beta$ to be zero. However, when $\lambda \to 0$, the regularization term is neglected resulting in no transfer. We can see from Figure 5.5 (c) that there is an operating zone of $\lambda$ (e.g., in the range of $10^{-4}$ to $10^{-2}$), that is neither too high nor too low for useful transfer from source metrics.

## 5.6 Conclusions

We addressed a critically important problem in person re-identification which has received little attention thus far - how to quickly on-board new cameras into an existing camera network. We showed this can be addressed effectively using hypothesis transfer learning using only learned source metrics and a limited amount of labeled data collected after installing the new camera(s). We provided theoretical analysis to show that our approach minimizes the effect of negative transfer through finding an optimal weighted combination of multiple source metrics. We showed the effectiveness of our approach on four standard datasets, significantly outperforming several baseline methods.

## 5.7 Appendix-4

### 5.7.1 Dataset Descriptions

This section contains detailed descriptions of the datasets used in our experiemnts (see Figure 5.6 for sample images).

**WARD [115]** was collected from three outdoor cameras. The dataset contains 4,786 images of 70 different persons and includes variations in illumination.

**RAiD [27]** was collected from four cameras; two indoor and two outdoor. 6,920 images were captured of 43 different persons. However, two of these persons were only seen by two of the four cameras. As a result of having both indoor and outdoor cameras, the dataset includes large illumination and viewpoint variations.

**Market1501 [232]** was collected from six cameras and used a Deformable Part Model [39] to annotate images. This resulted in 32,668 images of 1,501 different persons, but also 2,793 "distractors" that are badly drawn bounding boxes. The dataset includes variations in both detection precision, resolution and viewpoint.

**MSMT17 [191]** is the largest person re-identification dataset to date, and contains images collected by no more than 15 cameras; 3 indoor and 12 outdoor. Data was collected over the course of four different days in a month, and Faster RCNN [149] was using for bounding box detection, resulting in 126,441 images of 4,101 different persons. Due to the diversity in data collection, this dataset contains large variations in illumination and viewpoint.

(a) WARD (b) RAiD (c) Market1501 (d) MSMT17

Figure 5.6: A total of 48 Sample images from the 4 datasets used in our experimentation. In each row 4 different persons are shown whereas for each column 3 different views of the same person from 3 different cameras are shown. We can see the that across cameras, the viewpoint of the same person is very diverse because of change in illumination condition or occlusion.

### 5.7.2 Detailed Description of the Optimization Steps

In this section we will rigorously discuss all the necessary derivations of the steps of our proposed algorithm that could not be shown in the main chapter 5 due to space constraint. We first present the notations that we will use throughout this section.

**Notations:**

- $\frac{1}{n_s} \sum_{(i,j)\in S} x_{ij} x_{ij}^\top = \Sigma_S$
- $\frac{1}{n_d} \sum_{(i,j)\in D} x_{ij} x_{ij}^\top = \Sigma_D$
- $\mathcal{C}_1 = \{M \mid \frac{1}{n_d} \sum_{(i,j)\in D} (x_{ij}^\top M x_{ij}) - b \geq 0\}$
- $\mathcal{C}_2 = \{M \mid M \succeq 0\}$

- $\mathcal{C}_3 = \{\beta \mid \|\beta\|_2 \leq 1\}$
- $\Pi_{\mathcal{C}}(X) = \underset{\hat{X}\in\mathcal{C}}{\text{minimize}} \ \ \frac{1}{2}\|\hat{X} - X\|_F^2$
- $f(M,\beta) = \frac{1}{n_s}\sum\limits_{(i,j)\in S} x_{ij}^\top M x_{ij} + \lambda\|M - \sum_{j=1}^N \beta_j M_j\|_F^2$

The proposed optimization problem in the main chapter 5 is defined below.

$$\begin{aligned} \underset{M,\beta}{\text{minimize}} \quad & \frac{1}{n_s}\sum_{(i,j)\in S} x_{ij}^\top M x_{ij} + \lambda\|M - \sum_{j=1}^N \beta_j M_j\|_F^2 \\ \text{subject to} \quad & \frac{1}{n_d}\sum_{(i,j)\in D} (x_{ij}^\top M x_{ij}) - b \geq 0,\ M \succeq 0, \\ & \beta \geq 0,\ \|\beta\|_2 \leq 1 \end{aligned} \tag{5.1}$$

**Step 1: Gradient w.r.t $M$ with fixed $\beta$.**

$$\begin{aligned} \nabla_M(f(M,\beta)) &= \frac{1}{n_s}\sum_{(i,j)\in S} x_{ij}x_{ij}^\top + 2\lambda(M - \sum_{j=1}^N \beta_j M_j) \\ &= \Sigma_S + 2\lambda(M - \sum_{j=1}^N \beta_j M_j) \end{aligned} \tag{5.2}$$

**Step 2: Projection of $M$ onto $\mathcal{C}_1$ and $\mathcal{C}_2$.**

This can be done by solving a constrained optimization problem.

$$\begin{aligned} \Pi_{\mathcal{C}_1}(M) = \arg\min_{\hat{M}} \ \ & \frac{1}{2}\|\hat{M} - M\|_F^2 \\ \text{Subject to} \ \ & \frac{1}{n_d}\sum_{(i,j)\in D} (x_{ij}^\top \hat{M} x_{ij}) - b \geq 0 \end{aligned}$$

We can write the lagrangian as follows,

$$\mathcal{L}(\hat{M},\psi) = \frac{1}{2}\|\hat{M} - M\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top \hat{M} x_{ij}) \tag{5.3}$$

The KKT conditions for this problem are:

1.

$$\nabla_{\hat{M}}\mathcal{L}(\hat{M},\psi)\big|_{\hat{M}=\hat{M}^\star}=0 \implies (\hat{M}^\star - M) - \frac{\psi}{n_d}\sum_{(i,j)\in D} x_{ij}x_{ij}^\top = 0$$

$$\implies (\hat{M}^\star - M) - \psi\Sigma_D = 0 \implies \hat{M}^\star = M + \psi\Sigma_D$$

2. $\psi^\star(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top \hat{M}^\star x_{ij}) \geq 0$

3. $\psi^\star \geq 0$

The optimization problem is convex, so strong duality should hold. So, we put the value of $\hat{M}^\star$ from KKT condition 1 in the equation (5.3) to get the dual objective function as follows,

$$\begin{aligned}
g(\psi) = \mathcal{L}(\hat{M}^\star,\psi) &= \frac{1}{2}\|M+\psi\Sigma_D - M\|_F^2 + \psi\big(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top(M+\psi\Sigma_D)x_{ij}\big) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} x_{ij}^\top\Sigma_D x_{ij} \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} \text{trace}(x_{ij}^\top\Sigma_D x_{ij}) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \frac{\psi^2}{n_d}\sum_{(i,j)\in D} \text{trace}(\Sigma_D x_{ij}x_{ij}^\top) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\text{trace}(\Sigma_D\frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}x_{ij}^\top) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\text{trace}(\Sigma_D^\top\Sigma_D) \\
&= \frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) - \psi^2\|\Sigma_D\|_F^2 \\
&= -\frac{1}{2}\psi^2\|\Sigma_D\|_F^2 + \psi(b - \frac{1}{n_d}\sum_{(i,j)\in D} x_{ij}^\top M x_{ij})
\end{aligned} \tag{5.4}$$

To get the optimal $\psi^\star$ we have to maximize $g(\psi)$.

$$
\begin{aligned}
& g'(\psi^\star) = 0 \\
\implies & -\psi^\star \|\Sigma_D\|_F^2 + (b - \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij}^\top M x_{ij}) = 0 \\
\implies & \psi^\star = \frac{(b - \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2}
\end{aligned}
$$

But also from KKT condition (3), we know $\psi \geq 0$. Combining with the last equation we get

$$
\psi^\star = \max \left\{ 0, \frac{(b - \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2} \right\} \tag{5.5}
$$

So, putting the value of $\psi^\star$, finally we can write the projection from KKT condition 1 as,

$$
\Pi_{C_1}(M) = M + \max \left\{ 0, \frac{(b - \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij}^\top M x_{ij})}{\|\Sigma_D\|_F^2} \right\} \Sigma_D \tag{5.6}
$$

projection onto $\mathcal{C}_2$ is standard, so we are not discussing it here. **Step 3: Gradient w.r.t $\beta$ with fixed $M$.**

$$
\begin{aligned}
f(M^{k+1}, \beta) &= \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M^{k+1} x_{ij} + \lambda \|M^{k+1} - \sum_{j=1}^{N} \beta_j M_j\|_F^2 \\
&= K + \lambda \|M^{k+1} - \sum_{j=1}^{N} \beta_j M_j\|_F^2 \\
&= K + \lambda \text{trace}\Big( (M^{k+1} - \sum_{j=1}^{N} \beta_j M_j)^\top (M^{k+1} - \sum_{j=1}^{N} \beta_j M_j) \Big) \\
&= K + \lambda \beta_i^2 \text{trace}(M_i^\top M_i) - 2\lambda \beta_i \text{trace}(M_i^\top (M^{k+1} - \sum_{j=1, j\neq i}^{N} \beta_j M_j))
\end{aligned} \tag{5.7}
$$

$K$ is term which is independent of $\beta$. Now differentiating equation (5.7) w.r.t $\beta_i$ we get ,

$$\nabla_{\beta_i} f(M^{k+1}, \beta) = 2\lambda\beta_i \text{trace}(M_i^\top M_i) - 2\lambda \text{trace}(M_i^\top (M^{k+1} - \sum_{j=1, j\neq i}^{N} \beta_j M_j)) = a_i \quad (5.8)$$

So, derivative of $f(M^{k+1}, \beta)$ w.r.t $\beta$ is given by,

$$\nabla_{\beta} f(M^{k+1}, \beta) = \begin{bmatrix} a_1 & a_2 & \dots & a_N \end{bmatrix}^\top \quad (5.9)$$

**Step 4: Projection of $\beta$ onto $\mathcal{C}_3$.**

$$\Pi_{\mathcal{C}_3}(\beta) = \max\left\{0, \frac{\beta}{\max\{1, \|\beta\|_2\}}\right\} \quad (5.10)$$

The intuition here is that, when the norm of $\beta$ is greater than 1 then $\max\{1, \|\beta\|_2\} = \|\beta\|_2$ which implies the normalization of $\beta$. Similarly when the norm of $\beta$ is lesser or equal to 1 then $\max\{1, \|\beta\|_2\} = 1$, which means keeping the $\beta$ as it is since it already lies in the unit norm ball. The maximum with 0 essentially denotes the projection of any vector within the unit norm ball to the first quadrant of that ball only.

### 5.7.3 Proof of the Theorems

As mentioned in the chapter 5 the optimization proposed by us can be written in the same format as [142]

$$\underset{M\succeq 0}{\text{minimize}} \;\; L_T(M) + \lambda\|M - M_S\|_F^2 \quad (5.11)$$

where $M_S = \sum_{j=1}^{N} \beta_j M_j$ and

$$L_T(M) = \frac{1}{n_s} \sum_{(i,j)\in S} x_{ij}^\top M x_{ij} + \mu^\star\big(b - \frac{1}{n_d} \sum_{(i,j)\in D} x_{ij}^\top M x_{ij}\big) \quad (5.12)$$

**Theorem 1** *For the convex and k-Lipschitz loss defined in* (5.12) *the average bound can be expressed as*

$$\mathbb{E}_{T\sim\mathcal{D}_{\mathcal{T}}{}^n}[L_{\mathcal{D}_{\mathcal{T}}}(M^\star)] \leq L_{\mathcal{D}_{\mathcal{T}}}(\widehat{M_S}) + \frac{8k^2}{\lambda n}, \tag{5.13}$$

*where $n$ is the number of target labeled example, $M^\star$ is the optimal metric computed from Algorithm 1, $\widehat{M_S}$ is the average of all source metrics defined as $\frac{\sum_{j=1}^{N} M_j}{N}$, $\mathbb{E}_{T\sim\mathcal{D}_{\mathcal{T}}{}^n}[L_{\mathcal{D}_{\mathcal{T}}}(M^\star)]$ is the expected loss by $M^\star$ computed over distribution $\mathcal{D}_{\mathcal{T}}$ and $L_{\mathcal{D}_{\mathcal{T}}}(\widehat{M_S})$ is the loss of average of source metrics computed over $\mathcal{D}_{\mathcal{T}}$.*

**Proof.** If there is a single source metric is available for transfer , the proof has been shown in [142]. In case of multiple metric for any fixed $\beta$, we can directly replace $M_S$ by $\sum_{j=1}^{N}\beta_j M_j$ in the **Theorem 2** in [142] to get,

$$\mathbb{E}_{T\sim\mathcal{D}_{\mathcal{T}}{}^n}[L_{\mathcal{D}_{\mathcal{T}}}(M^\star)] \leq L_{\mathcal{D}_{\mathcal{T}}}\big(\sum_{j=1}^{N}\beta_j M_j\big) + \frac{8k^2}{\lambda n} \tag{5.14}$$

which is true $\forall\beta \in \mathcal{C}_3$. Where,

$$\beta = \begin{bmatrix}\beta_1 & \beta_2 & \dots & \beta_N\end{bmatrix}^\top \in \mathbb{R}^N \tag{5.15}$$

Clearly without loss of generality we can write $\beta = \beta'$ where,

$$\beta' = \begin{bmatrix}\frac{1}{N} & \frac{1}{N} & \dots & \frac{1}{N}\end{bmatrix}^\top \in \mathcal{C}_3 \tag{5.16}$$

since, $\beta' \geq 0$ and $\|\beta'\|_2 = \frac{1}{\sqrt{N}} \leq 1$. So, plugging $\beta'$ in equation (5.14) we get equation (5.10), which completes the proof. ■

**Theorem 2** *With probability $(1-\delta)$, for any metric $M$ learned from Algorithm 1 we have,*

$$L_{\mathcal{D}_{\mathcal{T}}}(M) \leq L_T(M) + \mathcal{O}(\frac{1}{n}) + \left(\sqrt{\frac{L_T(\sum_{j=1}^{N}\beta_j M_j)}{\lambda}} + \|\sum_{j=1}^{N}\beta_j M_j\|_F\right)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \tag{5.17}$$

*where $L_{\mathcal{D}_\mathcal{T}}(M)$ is the loss over the original target distribution (true risk), $L_T(M)$ is the loss over the existing target data (empirical risk), and $n$ is the number of target samples.*

**Proof.** In [142], $L_T(M)$ is defined as,

$$L_T(M) = \frac{1}{n^2} \sum_{(z_i, z_j) \in T} l(M, z_i, z_j) \tag{5.18}$$

■ The authors in [142] have used a specific loss for analysis,

$$l(M, z_i, z_j) = [yy'((z_i - z_j)^\top M(z_i - z_j) - \gamma_{yy'})]_+ \tag{5.19}$$

For our case,

$$\begin{aligned} L_T(M) &= \frac{1}{n_s} \sum_{(i,j) \in S} z_{ij}^\top M z_{ij} + \mu^\star \big(b - \frac{1}{n_d} \sum_{(i,j) \in D} z_{ij}^\top M z_{ij}\big) \\ &= \frac{1}{(n_s + n_d)} \frac{(n_s + n_d)}{n_s} \sum_{(i,j) \in S} z_{ij}^\top M z_{ij} + \frac{\mu^\star b (n_s + n_d)}{(n_s + n_d)} - \frac{\mu^\star (n_s + n_d)}{n_d} . \frac{1}{(n_s + n_d)} \sum_{(i,j) \in D} z_{ij}^\top M z_{ij} \\ &= \frac{1}{n^2} \sum_{(i,j) \in T} (\zeta_{ij} (z_i - z_j)^\top M (z_i - z_j) + \gamma) \end{aligned} \tag{5.20}$$

In our case we took similar and dissimilar pairs in equal number. So, for our case $n_s = n_d = \frac{n^2}{2}$ which implies $(n_s + n_d) = n^2$. Also, $\zeta_{ij} = (1 + \frac{n_d}{n_s}) = 2$ if $(i, j) \in S$ and $\zeta_{ij} = -\mu^\star(1 + \frac{n_s}{n_d}) = -2\mu^\star$ if $(i, j) \in D$ are soft labels. Also $\gamma = \mu^\star b(n_s + n_d) = \mu^\star b n^2$. so for our case,

$$l(M, z_i, z_j) = (\zeta_{ij} (z_i - z_j)^\top M (z_i - z_j) + \gamma) \tag{5.21}$$

Also unlike [142] our source metric is defined as $M_S = \sum_{j=1}^{N} \beta_j M_j$. With the loss in equation (5.21) if we follow the exact same steps as in proof of the **Lemma 2** of [142] then we will end up with the fact that our proposed loss is $(\sigma, m)$ admissible with $m =$

$2(1+\mu^\star)\max_{x,x'}\|x-x'\|_2^2\left(\sqrt{\frac{L_T(\sum_{j=1}^N \beta_j M_j)}{\lambda}}+\|\sum_{j=1}^N \beta_j M_j\|_F\right)$ and $\sigma=0$. Now putting these values of $\sigma$ and $m$ in the equation of inequality of **Theorem 4** of [142] which is,

$$L_{\mathcal{D}_\mathcal{T}}(M) \leq L_T(M) + \mathcal{O}\big(\frac{1}{n}\big) + (4\sigma + 2m + c)\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}, \tag{5.22}$$

and ignoring $c$ and the constant factor which are not functions of source metrics or their weights we conclude our proof.

### 5.7.4 Finding lipschitz constant for our loss

**Goal:** Our goal is to show the $k$ in equation (5.10) has a finite value. According to the definition the loss $l(M, x, x')$ is $k$-lipschitz with respect to its first argument if for any pair of matrices $M$ and $M'$ and pair of samples $x$ and $x'$ we have the inequality as follows for a finite non-negative $k$ $(0 \leq k < \infty)$

$$|l(M, x, x') - l(M', x, x')| \leq k\|M - M'\|_F \tag{5.23}$$

**Lemma 5** *The loss defined in equation* (5.21) *is k-lipschitz with* $k = 2\max(1, \mu^\star)\max_{x,x'}\|x - x'\|_2^2$

**Proof.**

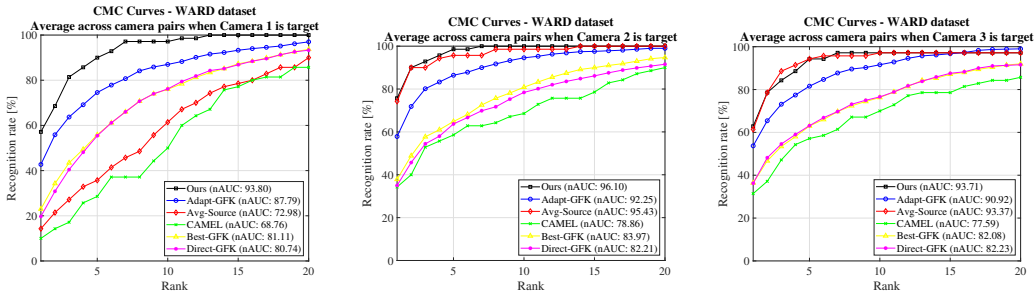
$$
\begin{aligned}
|l(M, x_i, x_j) - l(M', x_i, x_j)| &\leq |(\zeta_{ij}(x_i - x_j)^\top M(x_i - x_j) + \gamma) - (\zeta_{ij}(x_i - x_j)^\top M'(x_i - x_j) + \gamma)| \\
&\leq |\zeta_{ij}(x_i - x_j)^\top (M - M')(x_i - x_j)| \\
&\leq \max(|\zeta_{ij}|)\, |(x_i - x_j)^\top (M - M')(x_i - x_j)| \\
&\leq \max(2, 2\mu^\star)\, |(x_i - x_j)^\top (M - M')(x_i - x_j)| \\
&\leq 2\max(1, \mu^\star)\, \|x_i - x_j\|_2^2 \|M - M'\|_F \\
&\leq 2\max(1, \mu^\star) \max_{x,x'} \|x - x'\|_2^2 \|M - M'\|_F
\end{aligned}
\tag{5.24}
$$

Comparing this inequality with eq. (5.23) we get $k = 2\max(1, \mu^\star) \max_{x,x'} \|x - x'\|_2^2$, which is clearly non-negative and finite. ■

### 5.7.5 On-boarding a Single New Camera

This section covers the camera wise experimental results of on-boarding a single new camera (See Figure (5.7,5.9). We show for each dataset the camera wise CMC curves that are averaged to a single CMC curve in the main chapter 5. We also showed the comparison of GFK based methods in their original setting where source data is used during target adaptation in WARD dataset (See Figure 5.8).

Camera wise CMC curves for WARD dataset

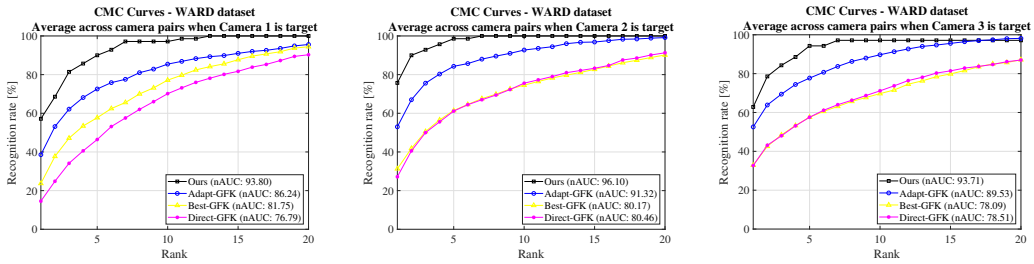



(a) (b) (c)

Figure 5.7: CMC curves for WARD[115] with 3 cameras. In this experiment each camera is shown as target while other two cameras served as source. The percentage label of new persons between the new target camera and the existing source cameras is taken to be 20% in this case. The most competitive method here is Adapt-GFK which is outperformed by our method in nAUC with margins 6%, 3.5% and 2.79% for camera 1,2 and 3 as target (plot a, b and c) respectively. In this case Adapt-GFK is calculated using the GFK matrix calculated by only using the limited labelled target data after the installation of new camera. Moreover for camera 1 as target (plot (a)) our method outperforms Adapt-GFK by a large rank-1 margin of almost 16%. Notable thing in this case is that there is only one source metric available for this dataset which is also handled by our multiple source metric transfer algorithm efficiently. Our method significantly outperform the semisupervised method CAMEL for all the plots which shows the strength of our method when a little target labeled data availabe. Also, our method outperforms Avg-Source for all the plots which is a proof of implication of Theorem 1.

Camera wise CMC curves for WARD dataset

(GFK computed for other relevant methods using old source data and new target data)




(a) (b) (c)

Figure 5.8: The setting in this case is exactly same as the setting of Figure 5.7. However this experiment is done only to compare our method with GFK methods in the original settings [133] where the assumption was of the availability of source data. In this case GFK is calculated using the old source data as well as new limited target data. Our method significantly outperforms all the GFK based methods in this case also. It proves that even if our method does not use source data, it still outperforms the doamin adaptation methods which uses source data.

### 5.7.6 On-boarding Multiple New Cameras

This section covers the camera wise experimental results of on-boarding multiple new cameras (See Figure (5.10,5.11,5.12). We show for each experiment the camera wise CMC curves that are averaged to a single CMC curve in the main chapter 5.

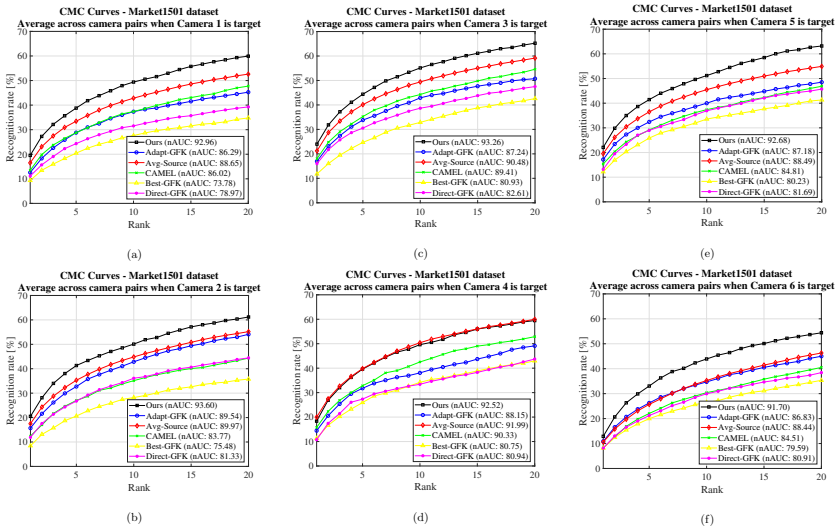



Figure 5.9: In this single camera insertion experiment Market1501 [232] dataset is used.

Camera wise CMC curves for Market1501 dataset: parallel addition of 2 cameras

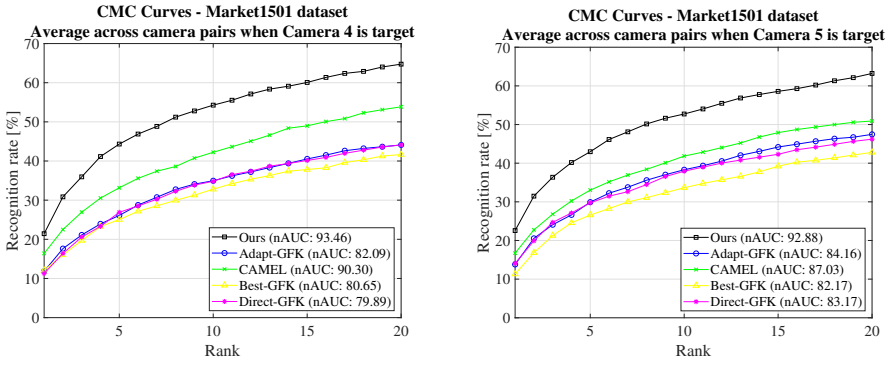
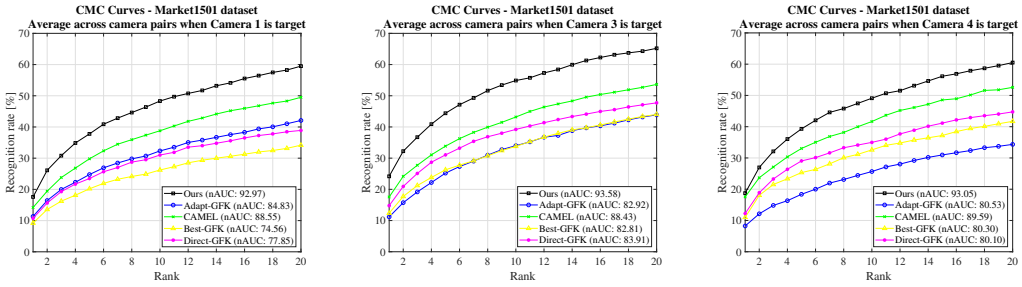



Figure 5.10: In this figure we used Market1501 dataset to show the effect of parallel on-boarding of multiple cameras (In this case 2 cameras). We effectively set camera 4 and 5 as target and compute 6 source metrics from the remaining cameras to transfer knowledge from. Accuracy is shown between camera 4 and camera (1,2,3,6) (plot(a)) and also between camera 5 and camera (1,2,3,6) (plot(b)) separately. We can see that our method significantly outperform other methods both in rank-1 and nAUC.

Camera wise CMC curves for Market1501 dataset: parallel addition of 3 cameras




(a) (b) (c)

Figure 5.11: In this figure we used Market1501 dataset to show the effect of parallel on-boarding of multiple cameras (In this case 3 cameras). We effectively set camera 1,3 and 4 as target and compute 3 source metrics from the remaining cameras to transfer knowledge from. Accuracy is shown between camera 1 and camera (2,5,6) (plot(a)),camera 3 and camera (2,5,6) (plot(b)) and also between camera 4 and camera (2,5,6) (plot(c)) separately. We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added in parallel. Best viewed in color.

### 5.7.7 Additional Experiments

**Effect of $\lambda = 0$:** When the existing pair-wise learned metrics are not considered (i.e., $\lambda = 0$), the rank-1 performance significantly drops from 62.86% to 27.14% on WARD. From that we conclude that a finite nonzero positive $\lambda$ is a very crucial factor in order for the algorihm to work.

**Initialization**: Since the proposed optimization is convex, initialization has very little effect on the performance. We tried 2 different initializations such as identity and random

positive semidefinite matrices with random weights within the first quadrant of unit-norm hypersphere, and found that both resulted minimal difference in rank-1 accuracy (RAiD: 51.25 vs 50.83 and WARD: 62.82 vs 62.38).

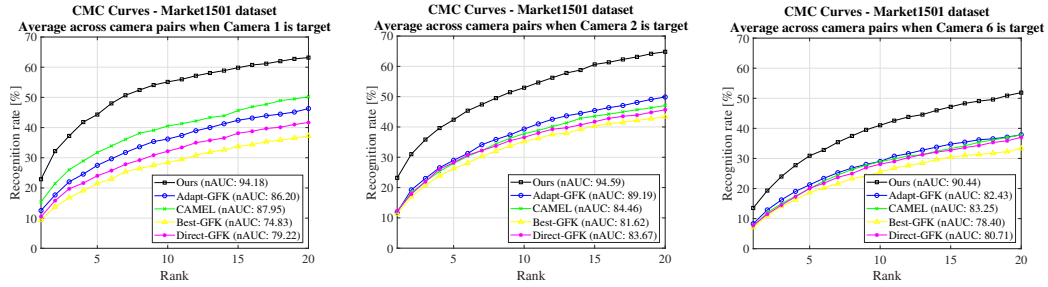
### 5.7.8 Finetuning with Deep Features

**Goal:** In this section our goal is to show the performance of our method (See Table **??** and Figure 5.13), if we have access to a deep model trained well using the source data.

**Implementation details:** This section covers the implementation details of finetuning deep features used in the experiments of Section 5.4 in the main chapter 5. First, we train a ResNet model [58], pretrained on the Imagenet dataset, using the source camera data. We remove the last classification layer and add two fully connected layers; one which embeds average pooled features to size 1024 and another which works as a classifier. We use the optimized source features to train the source metrics that will later be used to calculate new target metrics. Afterwards we fine-tune the model using the new target data and use the new optimized target features along with the source metrics in optimization 5.1. The model is trained for 50 epochs using SGD, with a base learning rate of 0.001, which is decreased by a factor 10 after 20 and 40 epochs. We use a batch size of 32 and perform traditional data augmentation, such as cropping and flipping. We use the optimized source features to train the source metrics that will later be used to calculate new target metrics. Afterwards, we fine-tune the model for 30 epochs using the new target data. We fine-tune with a batch size of 32 and a base learning rate is 0.0001 and decreased by a factor 10 after 20 epochs. The new optimized target features are used along with the source metrics in optimization. From Figure 5 (b) of the main chapter 5 and Figure 5.13 in here, we observe that when

we remove sixth camera in Market dataset, the accuracy of the test set between sixth and other cameras become very low as 20%, whereas in standard result for fully supervised deep model in Market dataset is around 80%. This drop in accuracy from 80 to 20% while removing $6^{th}$ camera in Market is due to two reasons. First, removing all the 151 person ids that appear in $6^{th}$ camera results in less labeled examples that leads to a less accurate deep model. Second, $6^{th}$ camera is the most uncorrelated with the other 5 cameras (see Fig. 7 in [232]). Figure 5(b) in main chapter 5 and Figure 5.13 in here clearly show that our approach works better than direct adaptation of the source model (even with finetuning) when feature distribution across source and target cameras are very different.

Camera wise CMC curves for Market1501 dataset: continuous addition of multiple cameras




(a) (b) (c)

Figure 5.12: In this figure we used Market1501 dataset to show the effect of sequential on-boarding of multiple cameras (In this case 3 cameras). Source cameras are camera 3,4 and 5 which has three source metrics between them. First camera 1 is added to the network and adapted. Accuracy for camera 1 as target is computed between camera 1 and camera (3,4,5) (plot(a)). Then camera 2 is added and adapted. For calculation of camera 2 adaptation accuracy we calculate matching score between camera 2 and camera (1,3,4,5) (plot(b)). In same fashion camera 6 is added afterwards and accuracy is calculated between camera 6 and camera (1,2,3,4,5) (plot(c)). We can see that our method significantly outperform other methods both in rank-1 and nAUC. This shows the effectiveness of our method for adaptation of multiple cameras in the network added sequentially.

CMC curves for Market1501 dataset with Camera 6 as target using deep learned features

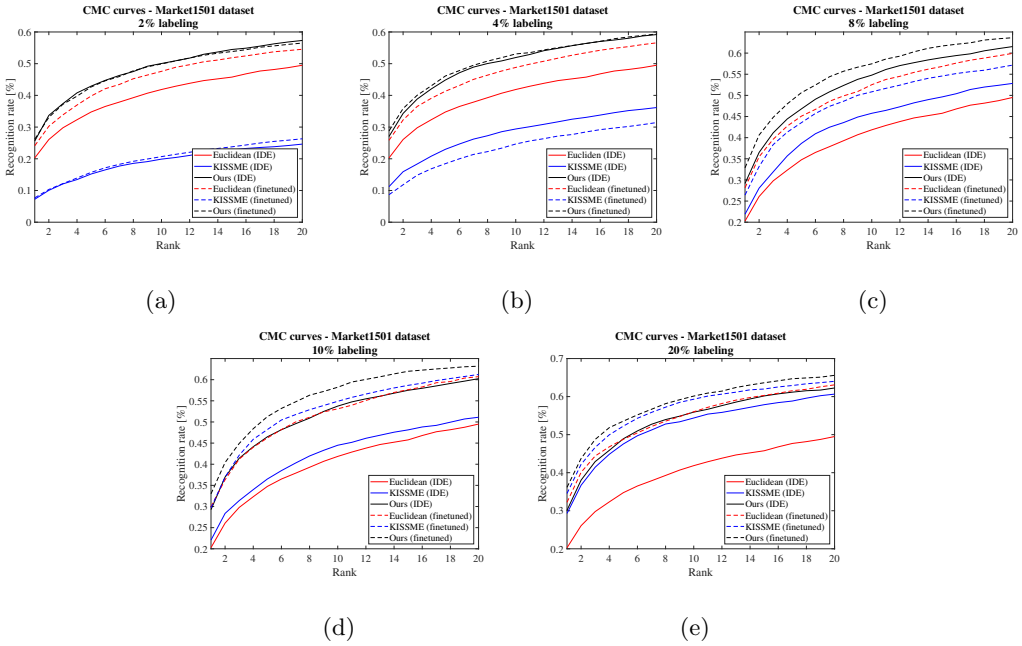



Figure 5.13: These plots show cmc curves for camera 6 of Market1501 dataset with different percentage labels in the target. We can clearly see that our method outperforms all the other (That is direct euclidean, direct metric learning and even fine tuning with target data). When the percentage label increase then our method with non-finetuned features merges with the direct fine tuning, whereas if we use our method with the finetuned features, it exceeds all the accuracy. This shows the strength of our method even in the presence of deep learned source model.

# Chapter 6

# Source Free Machine Unlearning

## 6.1 Introduction

Machine learning models have achieved significant success by training on large amounts of annotated data, much of which may include sensitive or private information [55]. With the introduction of data protection rules such as the General Data Protection Regulation (GDPR) [180], there is a growing need for algorithms that can delete (or forget) information learned from such sensitive datasets. Furthermore, privacy concerns may prompt individuals to request the removal of their data from the training set, invoking their "right to be forgotten" [114]. A straightforward solution to this issue would be to retrain the model from scratch using only the non-private subset of the original dataset. However, retraining is computationally inefficient and impractical (and impossible in the source-free setting we introduce in this chapter ). This highlights the necessity for efficient *Machine Unlearning* (MU) [48, 220] algorithms that enable modifications to the trained model parameters to forget specified data while maintaining performance on the remaining

data. Although several recent machine unlearning algorithms have demonstrated reasonable success on existing benchmarks [169, 89], nearly all current approaches [48, 220, 169, 89] assume the availability of the remaining data, either in full or in part. In practical settings, storing such large volumes of data is challenging due to storage costs and privacy issues. Consequently, these methods fail to address scenarios where the *model owner no longer has access to the original training data.* In these situations, ensuring accurate and efficient unlearning becomes markedly more difficult. Without the original data, it is challenging to verify that the specified information has been entirely removed from the model and that the model's performance on the remaining data remains unaffected. This limitation underscores the pressing need to develop robust unlearning techniques that can function effectively even when the original training dataset is inaccessible, i.e., the source-free setting.

A recent study has introduced a solution for this challenge, referring to the setting as "zero-shot machine unlearning" [24], which works by solely requiring access to the trained model weights and the data to be forgotten, without needing the original training dataset. However, a significant limitation of this technique is its inability to forget random instances encompassing diverse classes; it can only selectively forget particular data classes. This constraint could hinder its practicality in scenarios where users only want certain instances unlearned, as this method discards all the data pertaining to a user, rather than selectively removing certain examples. To tackle this limitation, another recent investigation [17] attempts zero-shot unlearning at the instance level instead of the class level. This approach enables the removal of requested data without requiring access to the complete training dataset. However, it suffers from scalability issues, as increasing the number of

instances results in a significant drop in performance on both test data and the remaining dataset, which is undesirable for effective and reliable machine unlearning. Additionally, these methods fall short in providing robust theoretical assurances regarding their respective performance.

Considering the aforementioned issues, we aim to design an unlearning algorithm that excels in such scenarios, where the original training data is unavailable, while providing robust theoretical guarantees. We term this as "source-free machine unlearning", in analogy to source-free domain adaptation methods [102]. (We believe this is a better term than zero-shot unlearning.) Inspired by [55], we study the unlearning mechanisms of $\ell_2$ regularized linear models with differentiable convex loss functions. Specifically, [55] define a Newton update step on the model parameters, which can be used to perform unlearning. This step is proven to be optimal for the quadratic loss function, and for strongly convex Lipschitz loss functions, the discrepancy between this step and optimal forgetting is bounded. Crucially, this Newton step requires the Hessian of the remaining data with respect to the trained model parameters. However, in our problem setup, we do not have access to the remaining data. Thus, we cannot compute this Hessian directly.

To tackle this issue, we introduce two algorithms to accurately estimate the Hessian of the remaining data, utilizing solely the data earmarked for removal and the trained model. The first algorithm is designed for any general convex loss function, while the second algorithm is tailored specifically for the quadratic loss function. By striving for a Hessian estimation closely aligned with the actual ground truth, our approach delivers robust theoretical assurances. This aspect is pivotal as it bolsters confidence in the machine

unlearning procedure for data removal, ensuring both precision and dependability. Our main contributions in this work can be summarized as following:

- To the best of our knowledge, this work proposes the first zero-shot unlearning method for linear classifiers that can effectively forget random instances of data from all classes while also providing robust theoretical guarantees regarding data removal and privacy.
- Since we cannot compute the Hessian directly from the remaining data, we propose two novel methods for estimating the Hessian from the remaining data for (i) for any general convex loss, and (ii) for the specialized case of quadratic mean squared error (MSE) loss. The first approach can approximately (i.e., with bounded error) unlearn for any convex loss functions, while the second is tailored specifically for quadratic loss and enables exact unlearning.
- We provide theoretical guarantees for our unlearning mechanism through extensive proofs and validate our claims with experiments and ablations on linear classifiers using multiple benchmark datasets.

## 6.2 Related works

**Machine unlearning.** Machine unlearning, introduced in [15], aims to efficiently remove the influence of certain training instances from a model's parameters. Unlearning approaches in the literature can be primarily categorised into exact and approximate unlearning methods. Exact unlearning methods aim to ensure that the data is completely unlearned from the model, akin to retraining from scratch. Recent approaches include

[12, 35], which split the data into multiple shards and train separate models on different non-overlapping combinations of these shards. However, it comes with substantial storage costs since multiple models must be maintained. In contrast, approximate unlearning methods estimate the influence of the unlearning instances and remove it through direct parameter updates. Some approximate methods focus on improving efficiency [196] or preserving performance [195], but they lack formal guarantees on data removal. A second group of approximate approaches [55, 124, 48, 47] provide theoretical guarantees on the statistical indistinguishability of unlearned and retrained models based on ideas similar to differential privacy [37]. All these methods require access to access to all, or a subset of, the training data. This assumption may not hold true in many practical settings; nevertheless, data privacy concerns may need to be addressed [46]. Recently, machine unlearning has attracted attention and achieved notable success in various applications, such as mitigating bias [22], erasing unwanted or copyrighted content [42, 96], and preventing malicious attacks [217] in recent large-scale generative models.

**Source-free unlearning.** A recent paper has proposed a method for unlearning which works by solely requiring access to the trained model weights and the data to be forgotten, without needing the original training dataset, referring to it as "zero-shot machine unlearning" [24]. They propose two approaches: error minimizing-maximizing noise and gated knowledge transfer. The first approach learns a set of noise matrices which maximize the error for the forget set, and a separate set of noise matrices which minimize the error as a proxy for the remaining data. The second approach uses knowledge distillation of the original model into a new model, gated by a filter that prevents the forget set knowledge from

being passed, and additionally, supplemented by a generator network for sample generation. A major limitation of these methods is their inability to forget specific instances of different classes; rather, they forget all the data pertaining to a class. However, such fine-grained forgetting scenarios are likely to occur in real-world applications, where the need for selective data removal or modification is prevalent. A recent work [17] proposes an adversarial sample generation strategy to extend zero-shot unlearning to the instance-wise case. However, this method struggles to scale beyond forgetting a few samples without significantly degrading model performance.

Critically, all existing zero-shot machine unlearning methods fail to provide any formal guarantees regarding the completeness or effectiveness of the forgetting process. In practical applications, where data privacy and compliance are paramount, such guarantees are essential to ensure that sensitive information is reliably removed from the model without compromising its overall performance. Additionally, a core contribution of our work is to provide such guarantees when the source data is no longer available.

## 6.3 Preliminaries

The mathematical concept central to the ideal machine unlearning setting is known as *parameter indistinguishability* [55]. In this section, we provide a brief overview of this definition. Additionally, we present the preliminaries of the unlearning mechanism for linear classifiers under convex losses.

### 6.3.1 Parameter Indistinguishability

Consider a data distribution $\mathcal{D} \sim \{x_i, y_i\}_{i=1}^n$ representing a training set used to train a model with a randomized algorithm $\mathcal{A}$ resulting in the output hypothesis space $\mathcal{H}$. Suppose there is a desire to eliminate the influence of $x_i$ from $\mathcal{H}$ using an unlearning mechanism $\Xi$. The unlearning mechanism is said to achieve *parameter indistinguishability*, if $\Xi$ functions in a manner such that the outputs of $\Xi(\mathcal{A}(\mathcal{D}), \mathcal{D}, x_i)$ and $\mathcal{A}(\mathcal{D} \backslash x_i)$ are very close to each other.

The current trend in unlearning research emphasizes demonstrating the efficacy of designed mechanisms using this metric. In simpler terms, the unlearned model should closely mimic, in terms of output space, the model that has been retrained from scratch without the specific data. Further discussion on this aspect will be provided in detail in the experimental section. In our case, we explore linear classifiers with the randomized algorithm $\mathcal{A}$ being the supervised learning, using standard convex loss functions.

### 6.3.2 Unlearning of Linear Classifier

The empirical loss with respect to a linear classifier $w \in \mathbb{R}^d$ and a convex loss function $l : \mathbb{R}^d \to \mathbb{R}$ can be written as $\mathcal{L}(w) = \sum_{i=1}^n l(y_i, w^\top x_i) + \frac{\lambda n}{2}\|w\|_2^2$. Let $w^\star = \arg\min_w \mathcal{L}(w)$ be the optimal linear classifier trained on the distribution $\mathcal{D}$. To forget a subset of the training data $\mathcal{D}_f \subset \mathcal{D}$, the naive approach involves retraining the classifier over the distribution $\mathcal{D}_r = \mathcal{D} \backslash \mathcal{D}_f$. However, this approach is impractical and time-consuming. A more widely used alternative is to mitigate the influence of the forget dataset using the influence function [55, 190] on the optimal model parameters. Mathematically, this

unlearning mechanism can be expressed as:

$$\Xi(w^\star, \mathcal{D}, \mathcal{D}_f) = w_{uf} = w^\star + H_r^{-1}\nabla_f \tag{6.1}$$

Here, $w_{uf}$ represents the model parameters obtained by unlearning the forget dataset, $w^\star$ is the optimal model parameter obtained using the entire training data, $H_r$ is the Hessian of the remaining dataset, and $\nabla_f$ is the gradient of the forget dataset at the optimal point $w^\star$. The term $-H_r^{-1}\nabla_f$ corresponds to the influence of the forget dataset on the model parameters. This unlearning methods is theoretically grounded and the residual norm of the gradient of the unlearned model on the remaining training set $\mathcal{D}_r$ can be tightly upper bounded. However, the assumption of having access to $\mathcal{D}$ during unlearning is strong and we relax this problem where we just have access to $\mathcal{D}_f$. However, without $\mathcal{D}_r$, computing the Hessian $H_r$ as in Eqn. 6.1 is non-trivial. To solve this, we devise a method where we can approximate this $H_r$ using only $w^\star$ and $\mathcal{D}_f$, which is elaborated in the next section in detail.

## 6.4 Methodology

### 6.4.1 Method for general convex losses (Method-1)

Given a differentiable convex loss $\mathcal{L}$, we can write the Taylor approximation of this around the optimal classifier $w^\star$ as follows:

$$\mathcal{L}(w) = \mathcal{L}(w^\star) + \nabla(w^\star)^\top(w - w^\star) + \frac{1}{2}(w - w^\star)^\top H(w^\star)(w - w^\star) + \xi(w^\star) \tag{6.2}$$

$$\approx \mathcal{L}(w^\star) + \nabla(w^\star)^\top(w - w^\star) + \frac{1}{2}(w - w^\star)^\top H(w^\star)(w - w^\star) \tag{6.3}$$

where $\zeta(w^\star)$ is the approximation error corresponding to the higher order terms of the taylor expansion. Assuming that the training converges to the global optima $w^\star$, we can safely assume that $\nabla(w^\star) = 0$. Plugging this in Eqn. 6.3 we get the following:

$$\delta\mathcal{L} = \mathcal{L}(w) - \mathcal{L}(w^\star) \approx \frac{1}{2}(w - w^\star)^\top H(w^\star)(w - w^\star) = \frac{1}{2}(\delta w)^\top H(w^\star)(\delta w) \tag{6.4}$$

Now we know that $\delta\mathcal{L}(w) = \mathcal{L}(w) - \mathcal{L}(w^\star) = (\mathcal{L}_f(w) - \mathcal{L}_f(w^\star)) + (\mathcal{L}_r(w) - \mathcal{L}_r(w^\star)) = \delta\mathcal{L}_f(w) + \delta\mathcal{L}_r(w)$, where $\mathcal{L}_f$ and $\mathcal{L}_r$ are the loss components corresponding to the forget and remaining training set. So the quantity $\frac{1}{2}(\delta w)^\top H(w^\star)(\delta w) - \delta\mathcal{L}_f(w) \approx \delta\mathcal{L}_r(w)$. Since, we are considering smooth convex loss functions, $\delta\mathcal{L}_r(w_i) \leq L\|w_i - w^\star\|$ where $L$ is the Lipschitz constant corresponding to the loss. As a result $\delta\mathcal{L}_r(w)$ can be upper bounded by $L\|\delta w\| \to 0$, for small perturbations. With this observation we generate some ($m$ points) small perturbations around the optima $w_i = w^\star + (\delta w)_i$ and calculate the average to formulate the following objective function of the Hessian as follows:

$$\Psi(H) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}(\delta w)_i^\top H(w^\star)(\delta w)_i - \delta\mathcal{L}_f(w_i)\right)^2 = \frac{1}{m}\sum_{i=1}^{m}\left[\text{trace}(P_i H)\right]^2 \tag{6.5}$$

where $P_i = \frac{1}{2}(\delta w)_i(\delta w)_i^\top - \frac{\delta\mathcal{L}_f(w_i)}{d}I_d$, where $I_d \in \mathbb{R}^{d\times d}$ is an identity matrix of dimension $d$. Since $\Psi(H) \to 0$ at the optima, minimizing it should output the desired Hessian $H$ w.r.t to the whole training data. However we need a proper constraint for the hessian matrix, otherwise the unconstrained minimization will result in $H = 0$.

Clearly the $H$ is positive semi definite (PSD) for any convex losses and also can be written as the sum $H_f$ and $H_r$, where these two matrices are the hessian with respect to the forget and remaining data respectively. Also both $H_f$ and $H_r$ are PSD matrices.

As a consequence, clearly $H_r = H - H_f \succeq 0$. Based on this observation, we formulate the following optimization as a Semi Definite Program (SDP) as follows:

$$\begin{aligned} \underset{H \succeq 0}{\text{minimize}} \quad & \Psi(H) \\ \text{subject to} \quad & H - H_f \succeq 0 \end{aligned} \tag{6.6}$$

If we can solve the optimization 5.1, we can retrieve $H(w^\star)$ and subtracting $H_f(w^\star)$ from that will result in our desired $H_r(w^\star)$ for the unlearning operation.

**Lemma 6** *Consider choosing $\delta w = \epsilon v$ where each element $v(j)$ of $v \in \mathbb{R}^d$ is sampled from $\mathcal{N}(0,1)$. Assuming that the optimization 6.6 achieves zero loss and the solution of the optimization converges to $\hat{H}$, then the trace of the difference between the Hessian $H$ (the actual ground truth Hessian with respect to $\mathcal{D}$) and $\hat{H}$ can be upper bounded as:*

$$-\frac{2L}{\epsilon\sqrt{d}} \leq \text{trace}(\Delta H) = \text{trace}(H - \hat{H}) \leq \frac{2L\sqrt{d}}{\epsilon(1-\kappa)},$$

*and subsequently,*

$$\|\Delta H\|_F \leq \frac{2Ld}{\epsilon(1-\kappa)}$$

*with a minimum probability of $1 - de^{-\frac{m\kappa^2}{2}}$, where $0 < \kappa < 1$.*

**Proof.** See Appendix 6.7.1. ■

**Implications of Lemma 6:** The lemma establishes a bound on the trace between the actual and estimated Hessians, delineated by two quantities. Notably, for large values of $d$, the lower bound approaches 0, suggesting that with high probability $\Delta H$ is a positive semidefinite (PSD) matrix. Consequently, an upper bound can be placed on the Frobenius norm of the difference. This implies that, on average, each element of the difference matrix

is at most $\frac{2Ld}{d^2\epsilon(1-\kappa)} = \frac{2L}{d\epsilon(1-\kappa)}$. Hence, the upper bound of the difference decreases linearly with the increase in matrix size. Note that we don't make any assumptions regarding the linearity of the model to prove this lemma. The bound indicates that if we can estimate the Hessian for large dimensions $d$ (such as in deep models), then this method could be a promising avenue to explore. However, it's evident that storing and inverting such a large Hessian is impractical and computationally inefficient. Nevertheless, various methods exist to approximate this Hessian, such as diagonalization [218] or linearizing the deep model using Hessian vector products [47]. Combining these approximation techniques with our approach could pave the way for promising research directions in the future.

**Theorem 7** *Suppose that* $\forall(x_i, y_i) \in \mathcal{D}$, $w \in \mathbb{R}^d$: $\|\nabla\ell(w^\top x_i, y_i)\| \leq C$. *Suppose that the second derivative of* $\ell$ *is* $\gamma$*-lipschitz and* $\|x_i\|_2 \leq 1$ *for all* $(x_i, y_i) \in \mathcal{D}$, *and the result of optimization 6.6 is* $\hat{H}$. *Then:*

$$\|\nabla\mathcal{L}(w_{uf}, \mathcal{D}_r)\|_2 \leq \gamma(n - n_f)\|\hat{H}^{-1}\nabla_f\|_2^2 \leq \frac{4\gamma C^2 n_f^2(n - n_f)}{\left[\lambda(n - n_f) - \alpha\sqrt{d}\right]^2}$$

*with a minimum probability of* $1 - de^{-\frac{m\kappa^2}{2}}$ *for* $\alpha = \frac{2L}{\epsilon(1-\kappa)}$, *where* $0 < \kappa < 1$.

**Proof.** See Appendix 6.7.2 ■

**Implications of Theorem 7:** The leftmost term in the theorem's inequality essentially represents the norm of the gradient with respect to the unlearned model on the remaining data. Successful unlearning, as suggested by the parameter indistinguishability (see Subsection 6.3.2), should lead this norm to approach 0. Examining the upper bound, we observe that for a fixed size $d$ of the Hessian, it becomes tighter with an increase in the

number of remaining data, as the term is inversely proportional to the remaining data size. We validate this phenomenon through experimentation, where we observe a decline in unlearning performance as the number of samples to be forgotten increases. Additionally, the upper bound tends to 0 as we significantly increase the Hessian dimension $d$. This finding aligns with Lemma 6, which asserts that for large $d$, the estimated and true Hessians closely resemble each other, indicating effective unlearning.

### 6.4.2 Special Case: Method for quadratic loss function (Method-2)

Method-1 is general and can be applied to any differentiable convex loss functions. However, if the loss is quadratic (i.e., the randomized algorithm $\mathcal{A}$ uses the standard Mean Square Error (MSE) loss), the unlearning mechanism $\Xi$ in 6.3.2 ensures $\Xi(\mathcal{A}(\mathcal{D}), \mathcal{D}, x_i) = \mathcal{A}(\mathcal{D} \setminus x_i)$. Thus, quadratic loss is particularly significant. Even in the absence of training data during unlearning, we develop an algorithm that ensures that the above condition holds true specifically for quadratic loss. We now explain this specialized algorithm (Method-2) in detail.

Let us consider a $k$ class classification problem with input $X \in \mathbb{R}^{n \times d}$, where $n$ and $d$ represents the number of samples and feature dimension respectively. Let $W \in \mathbb{R}^{d \times k}$ be the classifier that maps each input of $x_i \in \mathbb{R}^d$ of $X$ into a one-hot label of size $k$. Let's denote the resultant label matrix as $Y \in \mathbb{R}^{n \times k}$. In this case under the quadratic loss we can write the objective function as:

$$\mathcal{L}_{MSE}(W) = \frac{1}{2}\|XW - Y\|_F^2 = \frac{1}{2}\text{trace}(W^\top X^\top XW - 2W^\top X^\top Y + Y^\top Y) \tag{6.7}$$

If $W^\star$ is the minimizer of $\mathcal{L}_{MSE}(W)$, the we can say that at the optima:

$$\nabla \mathcal{L}_{MSE}(W^\star) = X^\top (XW^\star - Y) = 0$$

Which implies $X^\top X W^\star = X^\top Y$. If we put this expression of $H$ in Eqn. 6.7, we get the following:

$$\mathcal{L}_{MSE}(W^\star) = \frac{1}{2}\text{trace}(W^{\star\top} X^\top X W - 2W^{\star\top} X^\top X W^\star + Y^\top Y) = \frac{1}{2}\text{trace}(Y^\top Y - W^{\star\top} H W^\star)$$

since the Hessian at optima can be written as $H = X^\top X$. It is easy to see that $\text{trace}(Y^\top Y) = n$, and the training loss is close to 0 at optima for an ideal scenario. Saying that, if we can solve an $H$, such that $trace(W^{\star\top} H W^\star) = n$, we will get the Hessian of the whole training data. We do so by minimizing the expression of $\mathcal{L}_{MSE}(W^\star)$. Equivalently we write the optimization as follows:

$$\begin{aligned} \underset{H \succeq 0}{\text{maximize}} \quad & \text{trace}(W^{\star\top} H W^\star) \\ \text{subject to} \quad & H \succeq H_f, \text{Tr}(H) \leq n \end{aligned} \tag{6.8}$$

This algorithm does not use any random perturbation and utilizes all the information encoded within the trained model parameters. Thus, with very high probability, this method can reconstruct the exact Hessian on the remaining data and is independent of the number of forget data, unlike Method-1, as we will demonstrate in the experiments. Hence, while we analyze Method-1 for general convex cases, we also explore this alternative specifically tailored for the case of quadratic loss.

## 6.5 Experiments

**Datasets.** To demonstrate the efficacy of our proposed algorithms for the source-free unlearning scenario, we use four standard benchmark classification datasets: CIFAR-10 [86], CIFAR-100 [86], Stanford Dogs [79], and CalTech-256 [53]. CIFAR-10 is a dataset consisting of 60,000 RGB images in 10 different classes, with 6,000 images per class. CIFAR-100 is similar to CIFAR-10, but with 100 classes containing 600 images each, providing a more granular classification challenge. Stanford Dogs contains 20,580 images of 120 different breeds of dogs, making it ideal for fine-grained classification tasks. CalTech-256 comprises 30,607 images across 256 object categories, offering a diverse set of images for comprehensive object recognition research.

**Implementation details.** Since we perform zero-shot unlearning for linear classifiers, we use a ResNet-18 [58] architecture pre-trained on ImageNet [28] as our feature extractor. Using these features, we train a linear classifier and then discard the data. During unlearning, we only use the trained linear model and the data to be forgotten. We randomly sample up to 10% of the training data as the forget data. All experiments were performed on a single NVIDIA-RTX 3090 GPU.

**Baseline metrics.** The main baseline for any Machine Unlearning (MU) methods is the parameter indistinguishability between the retrained and unlearned models. A successful unlearning algorithm should emulate the performance of a model that was never exposed to the forget data, having been trained solely on the remaining data. In this context, we evaluate the classification accuracy of the model on the following datasets: (i) test data, (ii) remaining training data, and (iii) forget data. Additionally, we assess the Membership

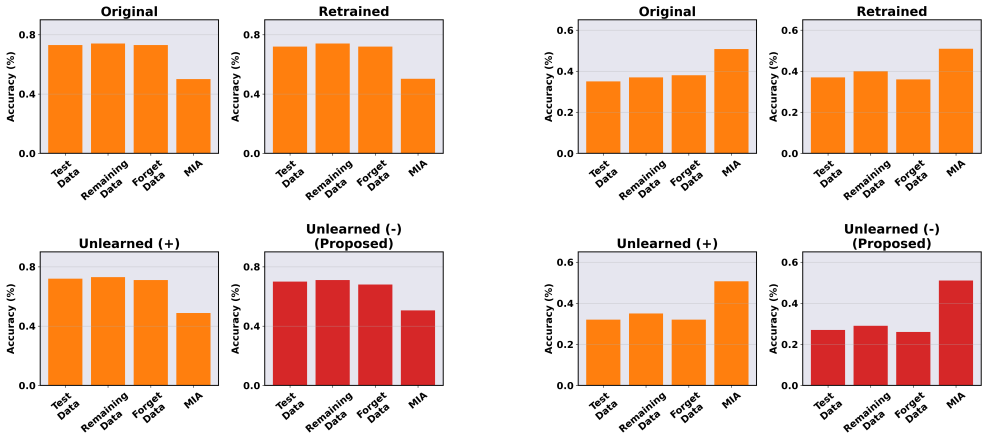
Inference Attack (MIA) score of the models, as proposed in the prior work [89]. This score indicates whether a sample was originally part of the training set. After forgetting certain samples, we check their MIA scores using the unlearned model. An MIA score close to 50% signifies successful unlearning, as it indicates that the unlearned model cannot distinguish whether the forget data came from the training distribution or the test distribution.

**Baseline models.** Unlearned models using our proposed algorithm are compared with three types of models: (a) A model trained with the whole training data (original), (b) A model retrained from scratch using the remaining training data (Retrained), and (c) An unlearned model that has been unlearned using the exact Hessian computed from the remaining data (Unlearned(+)). Since we estimate the remaining Hessian without accessing the remaining data, our primary objective is to closely mimic the performance of the model described in (c) using the baseline metrics. Since we do not need the training data during unlearning we refer the unlearned model using our proposed algorithm as (Unlearned(-)). We explore these model's performances using both **Method-1** and **Method-2**, for all the datasets.

### 6.5.1 Comparison of baseline metrics on different datasets

We compare the performance of Original, Retrained, Unlearned(+), and Unlearned(-) models on all four datasets (Fig. 6.1) by selecting 10% of the training samples as forget data. We use **Method-1** and quadratic loss as the convex loss function for all cases. The results are presented as bar plots for all these scenarios. As theoretically expected, the performance of Unlearned(+) closely mimics that of the Retrained model. As per the main results, in all cases, the performance of Unlearned(-) closely matches that of the

Unlearned(+) model, which aligns with our theoretical bounds.




CIFAR-10 CIFAR-100

Figure 6.1: Performance comparison of the proposed methods across different datasets: CIFAR-10 and CIFAR-100. We randomly select 10% of the entire training data as forget samples. Each figure illustrates the effectiveness of the optimization strategies in handling the forgetting of samples, as evidenced by the close performance of models *Unlearned(+)* and *Unlearned(-)*.

### 6.5.2 Effects of percentage of the forget data size

To investigate the influence of forget data size, we vary the proportion of randomly selected data for forgetting within the training set while maintaining consistency across all other factors. According to Theorem 7, it becomes apparent that the quantity of forgotten samples significantly influences the optimization process. As we can see in Table 6.1, increasing the number of forget samples negatively impacts performance. Specifically, when 5% of the training data is chosen randomly for forgetting, the disparity between the re-

| **Method** | | Test Data | Remaining Data | Forget Data | MIA |
|---|---|---|---|---|---|
| Retrained | | 73% | 75% | 73% | 50% |
| Unlearned (-) | 15% | 59% | 60% | 59% | 55.8% |
| Performance Gap | | **14%** | **15%** | **14%** | **5.8%** |
| Retrained | | 72% | 74% | 72% | 50% |
| Unlearned (-) | 10% | 70% | 71% | 68% | 51.4% |
| Performance Gap | | **2%** | **3%** | **4%** | **1.4%** |
| Retrained | | 73% | 74% | 73% | 49.4% |
| Unlearned (-) | 5% | 73% | 74% | 73% | 49.4% |
| Performance Gap | | **0%** | **0%** | **0%** | **0%** |

Table 6.1: The effect of the proportion of randomly selected data from the CIFAR-10 training dataset for forgetting. It's evident that as the number of forget data samples increases, the difference in performance between the Retrained and Unlearned(-) models also increases. Note that the second column is denoted to show the percentage of the selected forgetting data.

trained model with the remaining data and the model updated using our approach becomes negligible. However, with an increase in the percentage of forget data, the gap between these two models widens considerably. This result perfectly matches the bound we provide in Theorem 7.

| **Method** | | Test Data | Remaining Data | Forget Data | MIA |
|---|---|---|---|---|---|
| Retrained | | 72% | 74% | 72% | 50% |
| Unlearned (-) | 250 | 57% | 58% | 57% | 56.2% |
| Performance Gap | | **15%** | **16%** | **15%** | **6.2%** |
| Unlearned (-) | 500 | 70% | 71% | 68% | 51.4% |
| Performance Gap | | **2%** | **3%** | **4%** | **1.4%** |
| Unlearned (-) | 1000 | 72% | 74% | 71% | 49% |
| Performance Gap | | **0%** | **0%** | **1%** | **1%** |

Table 6.2: The effect of the number of perturbations randomly selected from Gaussian distribution for the CIFAR10 dataset. The second column is the number of perturbations used to approximate the hessian using our method. As we can see that increasing the number of perturbations positively influence the unlearning performance.

### 6.5.3 Effects of the number of perturbations

For this experiment, we conduct unlearning using Method-1 by varying the number of perturbations. In order to derive Lemma 6, we use the fact that $m$ is large. So clearly, according to Lemma 6, increasing the number of perturbations positively influences performance. This correlation is evident in our results in Table 6.2, where a higher number of perturbations consistently lead to improved outcomes.

### 6.5.4 Effects of the L2 regularization

The theoretical upper bound on the norm in Theorem 7 is clearly proportional to $\frac{1}{\lambda^2}$, with $\lambda$ representing the regularization parameter. Consequently, as demonstrated in Table 6.3, increasing the value of $\lambda$ leads to a reduction in the performance gap between our unlearned model and the retrained model.

| **Method** | | Test Data | Remaining Data | Forget Data | MIA |
|---|---|---|---|---|---|
| Retrained | | 72% | 74% | 72% | 50% |
| Unlearned (-) | 0 | 70% | 71% | 68% | 51.4% |
| Performance Gap | | **2%** | **3%** | **4%** | **1.4%** |
| Unlearned (-) | 0.0005 | 71% | 72% | 71% | 49.8% |
| Performance Gap | | **1%** | **2%** | **1%** | **0.2%** |
| Unlearned (-) | 0.001 | 72% | 73% | 72% | 50.9% |
| Performance Gap | | **0%** | **1%** | **0%** | **0.9%** |

Table 6.3: We demonstrate the impact of the regularization parameter $\lambda$ on our unlearning algorithm. It's evident that increasing the value of $\lambda$ leads to improved unlearning performance, consistent with our claim of Theorem 7.

### 6.5.5 Experiments on quadratic loss function using Method-2

With the proposed optimization specifically designed for quadratic loss (Method-2), we can effectively mitigate issues related to the number of forget samples (As compare to general Method-1). As shown in Table 6.4, the Method-1, applicable to all loss functions,

| | CIFAR10 | | | | CIFAR100 | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | Test Data | Remain Data | Forget Data | MIA | Test Data | Remain Data | Forget Data | MIA |
| Retrained | 73% | 75% | 73% | 48% | 34% | 38% | 34% | 50% |
| Unlearned (-)(Method-1) | 22% | 22% | 21% | 56% | 6% | 6% | 6% | 45% |
| Unlearned (-)(Method-2) | 69% | 71% | 7% | 50% | 32% | 35% | 31% | 50.5% |
| Performance Gap(Method-1) | **51%** | **53%** | **52%** | **8%** | **28%** | **32%** | **28%** | **5%** |
| Performance Gap(Method-2) | **4%** | **4%** | **3%** | **2%** | **2%** | **3%** | **3%** | **0.5%** |
| | **Stanford Dogs** | | | | **CalTech256** | | | |
| **Method** | Test Data | Remain Data | Forget Data | MIA | Test Data | Remain Data | Forget Data | MIA |
| Retrained | 25% | 76% | 22% | 45% | 38% | 80% | 38% | 50% |
| Unlearned (-)(Method-1) | 1% | 1% | 0% | 53.4% | 0% | 0% | 0% | 46% |
| Unlearned (-)(Method-2) | 20% | 73% | 21% | 50% | 32% | 74% | 33% | 51% |
| Performance Gap(Method-1) | **24%** | **75%** | **22%** | **8.4%** | **38%** | **80%** | **38%** | **4%** |
| Performance Gap(Method-2) | **5%** | **3%** | **1%** | **5%** | **6%** | **6%** | **5%** | **1%** |

Table 6.4: Performance comparison between **Method-1** and **Method-2** to illustrate that under quadratic loss, the performance of **Method-2** remains independent of the number of forget data samples, unlike **Method-1**, which is designed for any general convex loss function. We use 20% of the data as our forget data and observe a significant increase in the performance gap between the unlearned and retrained models for **Method-1**. However, the performance gap for **Method 2** remains considerably low even in the case of 20% forget data.

is significantly affected by the number of forget samples. In contrast, our second optimization (Method-2), tailored specifically for quadratic loss, delivers substantially better results regardless of the number of forget samples. For these experiments, 20% of the training dataset was randomly selected to be forgotten.

## 6.6 Conclusion

In this chapter , we introduce and evaluate two novel unlearning algorithms designed for linear classifiers, specifically targeting scenarios where the original training data is not available during the unlearning process. The first algorithm we present is a general-purpose method that is adaptable to a wide range of convex loss functions. This flexibility allows it to be applied in various contexts where different convex loss functions are used, making it a versatile tool for unlearning in diverse machine learning applications. The second algorithm is a more specialized solution, tailored specifically for the quadratic loss function. By focusing on this particular loss function, we are able to optimize the unlearning process for scenarios where quadratic loss is utilized, potentially enhancing performance and efficiency in these specific cases. We provide robust theoretical bounds for both algorithms, ensuring their reliability and effectiveness in unlearning tasks. These theoretical bounds offer a solid foundation for understanding the algorithms' behavior and performance guarantees. Furthermore, we explore the implications of these bounds and validate the practical effectiveness of our algorithms through extensive experimental evaluations. Our experimental results demonstrate that both algorithms perform significantly well, confirming their theoretical advantages and showcasing their potential in real-world applications.

## 6.7 Appendix-5

### 6.7.1 Proof for Lemma 6

**Proof.** Clearly from the definition,

$$\Psi(H) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}(\delta w)_i^\top H(w^\star)(\delta w)_i - \delta\mathcal{L}_f(w_i)\right)^2 = \frac{1}{m}\sum_{i=1}^{m}\left(\mathcal{L}_r(w_i) + \xi(w^\star)\right)^2$$

Using Cauchy-Schwarz inequality, we can lower bound $\Psi(H)$ as follows:

$$\left(\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}(\delta w)_i^\top H(w^\star)(\delta w)_i - \delta\mathcal{L}_f(w_i)\right)\right)^2 \leq \Psi(H) \approx \frac{1}{m}\left(\sum_{i=1}^{m}\mathcal{L}_r(w_i)^2 + 2\xi(w^\star)\sum_{i=1}^{m}\mathcal{L}_r(w_i)\right) \tag{6.9}$$

Defining the noise covariance matrix as $\Sigma = \frac{1}{m}\sum_{i=1}^{m}(\delta w)_i(\delta w)_i^\top = \epsilon^2\frac{1}{m}\sum_{i=1}^{m}v_i v_i^\top = \epsilon^2\Sigma_v$,

we can write:

$$\frac{1}{2m}\sum_{i=1}^{m}(\delta w)_i^\top H(w^\star)(\delta w)_i = \frac{\epsilon^2}{2}\text{trace}(\Sigma_v H)$$

Also we can upper bound $\delta\mathcal{L}_r(w_i) \leq L\|\delta w_i\|_2$. Since, we assume zero loss at optimal $\hat{H}$, we can say:

$$\frac{1}{2}(\delta w)_i^\top \hat{H}(w)(\delta w)_i \approx \delta\mathcal{L}_f(w_i)\ \forall i$$

As a result the leftmost term in Eqn.6.9 can be approximated by

$$\Psi(H) \approx \left(\frac{\epsilon^2}{2}\text{trace}(\Sigma_v H) - \frac{\epsilon^2}{2}\text{trace}(\Sigma_v\hat{H})\right)^2 = \left(\frac{\epsilon^2}{2}\text{trace}(\Sigma_v\Delta H)\right)^2$$

Also the rightmost term of Eqn.6.9 can be upper bounded by:

$$\frac{1}{m}\left(\sum_{i=1}^{m}\mathcal{L}_r(w_i)^2) + 2\xi(w^\star)\sum_{i=1}^{m}\mathcal{L}_r(w_i)\right) \leq L\left(L\mathbb{E}(\|\delta w_i\|_2^2) + 2\xi(w^\star)\mathbb{E}(\|\delta w_i\|_2)\right)$$

Since we choose $v_i(j) \sim \mathcal{N}(0,1)$ , $\|v\|_2^2$ follows a $\chi_d^2$ distribution with degrees of freedom $d$. Clearly $\mathbb{E}(\|v\|) = \sqrt{2}\frac{\Gamma(\frac{d+1}{2})}{\Gamma(\frac{d}{2})} \approx \sqrt{2}\sqrt{\frac{d}{2}} = \sqrt{d}$ and $\mathbb{E}(\|v\|^2) = d$ for sufficiently large $d$. Now the last inequality can be rewritten as:

$$\left(\frac{\epsilon^2}{2}\text{trace}(\Sigma_v \Delta H)\right)^2 \leq L\left(L\epsilon^2 d + 2\xi(w^\star)\epsilon\sqrt{d}\right)$$

Since $\xi(w^\star)\epsilon \to 0$, we neglect this term and the inequality becomes:

$$\left(\frac{\epsilon^2}{2}\text{trace}(\Sigma_v \Delta H)\right)^2 \leq L^2\epsilon^2 d$$

As a result we can write,

$$-\frac{2L\sqrt{d}}{\epsilon} \leq \text{trace}(\Sigma_v H) \leq \frac{2L\sqrt{d}}{\epsilon}$$

Now, we know that since $v$ is i.i.d random gaussian noise, then the covariance will converge to identity matrix with sufficiently large $m$. In other words, $\mu = \mathbb{E}(Y_i) \to I_d$, where $Y_i = v_i v_i^\top$ are sequence of i.i.d random PSD matrices. From definition $0 \preceq Y_i \preceq I_d$ and let's take $0 \leq \kappa \leq 1$. So, in this scenario we can apply matrix chernoff bounds in order to provide concentration bound as follows:

$$\Pr\left[\lambda_{\min}\left(\frac{1}{m}\sum_{i=1}^{m} Y_i\right) \leq (1-\kappa)\lambda_{\min}(\mu)\right] \leq d\exp\left(-\frac{m\kappa^2\lambda_{\min}(\mu)}{2}\right)$$

Here $\lambda_{min}(.)$ is the minimum eigenvalue operator. So, $\lambda_{min}(\mu) = \lambda_{\min}(I_d) = 1$. As a result we can say the following:

$$\Pr\left[\lambda_{\min}\left(\Sigma_v\right) \geq (1-\kappa)\right] \geq 1 - d\exp\left(-\frac{m\kappa^2}{2}\right)$$

Now since $\Sigma_v \succeq \lambda_{\min}(\Sigma_v) I_d$, we can conclude the following:

$$\mathrm{trace}(\Sigma_{\mathrm{v}}\mathrm{H}) \geq \mathrm{trace}(\lambda_{\min}(\Sigma_{\mathrm{v}})\mathrm{I_d}\mathrm{H}) \geq (1-\kappa)\mathrm{trace}(\mathrm{H})$$

So in conclusion, with a minimum probability of $1 - d\exp\left(-\frac{m\kappa^2}{2}\right)$, we have:

$$(1-\kappa)\mathrm{trace}(\Delta H) \leq \mathrm{trace}(\Sigma_v \Delta H) \leq \frac{2L\sqrt{d}}{\epsilon}$$

This implies,

$$\mathrm{trace}(\Delta H) \leq \frac{2L\sqrt{d}}{\epsilon(1-\kappa)}$$

Now for the lower bound, since we know that $\Sigma_v$ approaches $I_d$ in expectation and is a PSD matrix, we can say $\Sigma_v \preceq \lambda_{max}(\Sigma_v) I_d \preceq \mathrm{trace}(\Sigma_v) I_d$, which implies $\mathrm{trace}(\Sigma_v \Delta H) \leq \mathrm{trace}(\Sigma_v)\mathrm{trace}(\Delta H) \leq d\,\mathrm{trace}(\Delta H)$. Since, $-\frac{2L\sqrt{d}}{\epsilon} \leq \mathrm{trace}(\Sigma_v \Delta H)$, we can say $-\frac{2L\sqrt{d}}{\epsilon} \leq d\,\mathrm{trace}(\Delta H)$. This gives us our final inequality:

$$-\frac{2L}{\epsilon\sqrt{d}} \leq \mathrm{trace}(\Delta H) \leq \frac{2L\sqrt{d}}{\epsilon(1-\kappa)}$$

This concludes the first part of the proof.

We see that when $d$ is large enough, the lower bound is close to 0 and the matrix $\Delta H$ is PSD. Then by definition, $\|\Delta H\|_F \leq \sqrt{\mathrm{rank}(A)}\|A\|_2 \leq \sqrt{d}\lambda_{\max}(\Delta H) \leq \sqrt{d}\mathrm{trace}(\Delta H) \leq \frac{2Ld}{\epsilon(1-\kappa)}$. This concludes the second part and completes the proof. ■

### 6.7.2 Proof of Theorem 7

**Proof.** This proof is inspired by [55], and based on **Theorem 4** of the chapter. This bound says that upon forgetting $n_f$ samples from the dataset, if the resulting model

become $w_{uf}$ then the norm of the gradient with respect to this model on the remaining dataset can be upper bounded as follows:

$$\|\nabla\mathcal{L}(w_{uf}, \mathcal{D}_r)\|_2 \leq \gamma(n - n_f)\|H^{-1}\nabla_f\|_2^2$$

Now this $H$ is the actual hessian of the remaining data while our estimate is $\hat{H} = H - \Delta H$. From the definition of loss $\mathcal{L}$, we know that after removing $n_f$ samples the loss becomes $\lambda(n-n_f)$-strongly convex. As a result we get $\|H\|_2 \geq \lambda(n-n_f)$. Now we can apply triangle inequality and the upper bound from Lemma 6 as follows:

$$\lambda(n - n_f) \leq \|H\|_2 \leq \|\hat{H}\|_2 + \|\Delta H\|_2 \leq \frac{2L\sqrt{d}}{\epsilon(1-\kappa)} = \alpha\sqrt{d}$$

$$\implies \|\hat{H}\|_2 \geq \lambda(n - n_f) - \alpha\sqrt{d} \implies \|\hat{H}\|_2^{-1} \leq \frac{1}{(\lambda(n - n_f) - \alpha\sqrt{d})}$$

Also, from **Theorem 4** of [55], we know $\|\nabla_f\| \leq 2Cn_f$. So,

$$\|\hat{H}^{-1}\nabla_f\|_2^2 \leq \|\hat{H}^{-1}\|_2^2\|\nabla_f\|_2^2 \leq \frac{4C^2n_f^2}{(\lambda(n - n_f) - \alpha\sqrt{d})^2}$$

$$\implies \|\nabla\mathcal{L}(w_{uf}, \mathcal{D}_r)\|_2 \leq \frac{4\gamma C^2n_f^2(n - n_f)}{(\lambda(n - n_f) - \alpha\sqrt{d})^2}$$

Hence we conclude the proof of Theorem 7. ■

# Chapter 7

# Conclusions

In this dissertation, we introduce a series of methods enabling the domain-adaptive learning of deep models to address diverse distributional shifts and facilitate model unlearning in a source-free setting. These shifts encompass a spectrum of scenarios, including unlabeled or sparsely labeled target data, static or dynamic target distributions, or data from entirely different modalities. Our methods aim to achieve this adaptation with minimal or no supervision, ensuring rapid and effective generalization to new and challenging conditions. Additionally, we devise a source-free unlearning scenario enabling the model to efficiently discard specific data without reliance on the source data.

In Chapter 2, we develop a new UDA algorithm that can learn from and optimally combine multiple source models without requiring source data. We provide theoretical intuitions for our algorithm and verify its effectiveness in a variety of domain adaptation benchmarks. In Chapter 3, we extend the work of Chapter 2 by introducing a novel framework named CONTRAST, which efficiently combines multiple source models during test

time with small batches of streaming data, all without requiring access to the source data. It achieves a test accuracy that is at least as good as the best individual source model. Moreover, the design of CONTRAST naturally mitigates the issue of catastrophic forgetting. To validate the effectiveness of our algorithm, we conduct experiments across a diverse range of benchmark datasets for classification and semantic segmentation tasks. We demonstrate that CONTRAST seamlessly integrates with a variety of single-source methods.

In Chapters 4 and 5, we explore two applications of source free multi-source adaptation. In Chapter 4, we identify the novel and challenging problem of cross-modality knowledge transfer without access to task-relevant data from the source sensor modality, relying solely on unlabeled data in the target modality. To address this, we propose our framework, SOCKET, which includes the development of specialized loss functions to bridge the gap between the two modalities in the feature space. Our experiments, conducted for both RGB-to-depth and RGB-to-NIR scenarios, demonstrate that SOCKET consistently outperforms baseline methods that struggle to effectively handle modality shifts. Whereas, in Chapter 5, we tackle a critically important but under-explored challenge in person re-identification: rapidly integrating new cameras (models) into an established camera network. We demonstrated that this task can be effectively addressed through hypothesis transfer learning, leveraging learned source metrics (models) and a limited amount of labeled target data collected post-installation of the new camera(s). Our theoretical analysis highlights that our approach mitigates negative transfer effects by identifying an optimal weighted combination of multiple source metrics. Empirical results on four standard datasets showcase the effectiveness of our approach, significantly surpassing several baseline methods.

Finally in Chapter 6, we introduce and evaluate two novel unlearning algorithms tailored for linear classifiers, specifically addressing scenarios where the original training data is unavailable during the unlearning process. The first algorithm we present is a versatile method adaptable to a wide array of convex loss functions. Its flexibility allows it to be applied across various contexts employing different convex loss functions, rendering it a versatile tool for unlearning in diverse machine learning applications. The second algorithm is a specialized solution designed for the quadratic loss function. By focusing on this specific loss function, we optimize the unlearning process for scenarios utilizing quadratic loss, potentially enhancing performance and efficiency in such cases. We provide robust theoretical bounds for both algorithms, ensuring their reliability and effectiveness in unlearning tasks. These theoretical bounds establish a solid foundation for comprehending the algorithms' behavior and performance guarantees. Additionally, we explore the implications of these bounds and validate the practical effectiveness of our algorithms through extensive experimental evaluations. Our experimental results demonstrate that both algorithms perform significantly well, confirming their theoretical advantages and showcasing their potential in real-world applications.

While the methods presented in this thesis provide valuable insights into adaptation techniques, they represent only a fraction of the potential problems in this field. It is essential to acknowledge that there is still much to explore and discover. In conclusion, we briefly outline several logical extensions of this work that can pave the way for future research. These extensions hold promise for further advancements in the field and offer potential directions for future investigations.

**Unsupervised Validation Set.** In Chapter 2, I discussed Unsupervised Domain Adaptation (UDA) in a source-free setting. However, a fundamental question remains regarding the validation set in UDA. In a supervised learning setup, we can easily validate our algorithm using labeled data to determine hyperparameters such as the regularization constant or the number of epochs. However, in UDA, the concept of a validation set does not exist, necessitating a principled approach to optimize our algorithm. This represents a promising direction for future research.

**Scalable Multi-Source.** In Chapters 2 and 3, we explore the multi-source setup but experiment with a limited number of source models (up to six). This raises a fundamental question of scalability: will our approach work if we use hundreds or thousands of source models?

**General Cross-Modal Adaptation.** In Chapter 4, we explore cross-modal adaptation in the image domain using modalities such as depth, RGB, and IR. However, our algorithm needs to be generalized to work with other modalities, such as audio-video or audio-speech. Incorporating Vision Language Models (VLMs) could be a potential solution for achieving this generalization.

**Unlearning Large Deep Models.** In Chapter 6, we explored the algorithm for a linear classifier and conducted unlearning in a principled way by providing theoretical bounds. However, there is no source-free unlearning method for large deep models that provides a theoretical bound. Linearizing a deep model using Taylor approximation and then performing unlearning might be a potential direction. Nonetheless, our proposed method involves

an SDP solver, which may not be feasible when the Hessian is too large, as in deep models. Solving these large-scale matrices presents a challenging and potential direction for future research.

**Domain Adaptive Unlearning.** As the title of this thesis suggests, the main topic is domain adaptive learning and unlearning. While we primarily focus on adaptive learning, our exploration of unlearning considers scenarios where the data to be forgotten is from the same distribution as the training data. However, in practice, users might provide their data for unlearning, which could exhibit a domain shift from the original training data. This raises crucial questions about how to perform unlearning effectively under domain shift and the extent of theoretical guarantees we can provide based on the degree of domain shift. Addressing these questions in future work will further substantiate the thesis title.

# Bibliography


[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Peshal Agarwal, Danda Pani Paudel, Jan-Nico Zaech, and Luc Van Gool. Unsupervised robust domain adaptation without source data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2009–2018, 2022.

[3] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *CVPR*, pages 3908–3916, 2015.

[4] Sk Miraj Ahmed, Aske R Lejbolle, Rameswar Panda, and Amit K Roy-Chowdhury. Camera on-boarding for person re-identification using hypothesis transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12144–12153, 2020.

[5] Sk Miraj Ahmed, Suhas Lohit, Kuan-Chuan Peng, Michael J Jones, and Amit K Roy-Chowdhury. Cross-modal knowledge transfer without task-relevant source data. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 111–127. Springer, 2022.

[6] Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *CVPR*, 2021.

[7] N. N. Author. Suppressed for anonymity, 2021.

[8] Ali Ayub and Alan R Wagner. Centroid based concept learning for rgb-d indoor scene classification. *arXiv preprint arXiv:1911.00155*, 2019.

[9] Haoyue Bai, Fengwei Zhou, Lanqing Hong, Nanyang Ye, S-H Gary Chan, and Zhenguo Li. Nas-ood: Neural architecture search for out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8320–8329, 2021.

[10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[11] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. 2018.

[12] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

[13] John S Bridle, Anthony JR Heading, and David JC MacKay. Unsupervised classifiers, mutual information and'phantom targets. In *Advances in neural information processing systems*, pages 1096–1101, 1992.

[14] Matthew Brown and Sabine Süsstrunk. Multi-spectral sift for scene category recognition. In *CVPR 2011*, pages 177–184. IEEE, 2011.

[15] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

[16] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

[17] Sungmin Cha, Sungjun Cho, Dasol Hwang, Honglak Lee, Taesup Moon, and Moontae Lee. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11186–11194, 2024.

[18] Goirik Chakrabarty, Manogna Sreenivas, and Soma Biswas. Sata: Source anchoring and target alignment network for continual test time adaptation. *arXiv preprint arXiv:2304.10113*, 2023.

[19] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[20] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022.

[21] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[22] Ruizhe Chen, Jianfei Yang, Huimin Xiong, Jianhong Bai, Tianxiang Hu, Jin Hao, Yang Feng, Joey Tianyi Zhou, Jian Wu, and Zuozhu Liu. Fast model debias with machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Jaehoon Cho, Dongbo Min, Youngjung Kim, and Kwanghoon Sohn. Deep monocular depth estimation leveraging a large-scale outdoor stereo dataset. *Expert Systems with Applications*, 178:114877, 2021.

[24] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 2023.

[25] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[26] Rui Dai, Srijan Das, and François Bremond. Learning an augmented rgb representation with cross-modal knowledge distillation for action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13053–13064, 2021.

[27] Abir Das, Anirban Chakraborty, and Amit K Roy-Chowdhury. Consistent re-identification in a camera network. In *ECCV*, pages 330–345. Springer, 2014.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[29] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*, pages 994–1003, June 2018.

[30] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. Source-free domain adaptation via distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7212–7222, 2022.

[31] Dapeng Du, Limin Wang, Huiling Wang, Kai Zhao, and Gangshan Wu. Translate-to-recognize networks for rgb-d scene recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11836–11845, 2019.

[32] Simon S Du, Jayanth Koushik, Aarti Singh, and Barnabás Póczos. Hypothesis transfer learning via transformation functions. In *Advances in Neural Information Processing Systems*, pages 574–584, 2017.

[33] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive methods for real-world domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14340–14349, 2021.

[34] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2000.

[35] Yonatan Dukler, Benjamin Bowman, Alessandro Achille, Aditya Golatkar, Ashwin Swaminathan, and Stefano Soatto. Safe: Machine unlearning with shard graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17108–17118, 2023.

[36] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006.

[37] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[38] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(4):83, 2018.

[39] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[40] Andrea Ferreri, Silvia Bucci, and Tatiana Tommasi. Translate to adapt: Rgb-d scene recognition across domains. *arXiv preprint arXiv:2103.14672*, 2021.

[41] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pages 357–368. PMLR, 2017.

[42] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2426–2436, 2023.

[43] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[44] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[45] Nuno C Garcia, Sarah Adel Bargal, Vitaly Ablavsky, Pietro Morerio, Vittorio Murino, and Stan Sclaroff. Dmcl: Distillation multiple choice learning for multimodal action recognition. *arXiv preprint arXiv:1912.10982*, 2019.

[46] General Data Protection Regulation GDPR. General data protection regulation. *URL: https://gdpr-info. eu/[accessed 2020-11-21]*, 2018.

[47] Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 792–801, 2021.

[48] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

[49] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[50] Mengran Gou, Ziyan Wu, Angels Rates-Borras, Octavia Camps, Richard J Radke, et al. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):523–536, 2018.

[51] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[52] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, pages 262–275. Springer, 2008.

[53] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[54] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. In *CVPR*, pages 498–505. IEEE, 2009.

[55] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[56] Jiang Guo, Darsh J Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. *arXiv preprint arXiv:1809.02256*, 2018.

[57] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836, 2016.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[59] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[60] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[61] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Scandinavian conference on Image analysis*, pages 91–102. Springer, 2011.

[62] Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 867–874, 2014.

[63] Judy Hoffman, Saurabh Gupta, Jian Leong, Sergio Guadarrama, and Trevor Darrell. Cross-modal adaptation for rgb-d detection. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 5032–5039. IEEE, 2016.

[64] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems*, pages 8246–8256, 2018.

[65] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[66] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

[67] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 749–757, 2020.

[68] Minhao Hu, Tao Song, Yujun Gu, Xiangde Luo, Jieneng Chen, Yinan Chen, Ya Zhang, and Shaoting Zhang. Fully test-time adaptation for image segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*, pages 251–260. Springer, 2021.

[69] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 8022–8031, 2019.

[70] Xuefeng Hu, Gokhan Uzunbas, Sirius Chen, Rui Wang, Ashish Shah, Ram Nevatia, and Ser-Nam Lim. Mixnorm: Test-time adaptation through online normalization estimation. *arXiv preprint arXiv:2110.11478*, 2021.

[71] Sergey Ioffe and Christian Szegedy Batch Normalization. Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

[72] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[73] Masato Ishii and Masashi Sugiyama. Source-free domain adaptation via distributional alignment by matching batch normalization statistics. *arXiv preprint arXiv:2101.10842*, 2021.

[74] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

[75] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[76] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

[77] M. J. Kearns. *Computational Complexity of Machine Learning*. PhD thesis, Department of Computer Science, Harvard University, 1989.

[78] Koulik Khamaru and Martin Wainwright. Convergence guarantees for a class of non-convex and non-smooth optimization problems. In *International Conference on Machine Learning*, pages 2601–2610. PMLR, 2018.

[79] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[80] Sunok Kim, Dongbo Min, Bumsub Ham, Seungryong Kim, and Kwanghoon Sohn. Deep stereo confidence prediction for depth estimation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 992–996. IEEE, 2017.

[81] Youngjung Kim, Bumsub Ham, Changjae Oh, and Kwanghoon Sohn. Structure selective depth superresolution for rgb-d cameras. *IEEE Transactions on Image Processing*, 25(11):5227–5238, 2016.

[82] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE transactions on Image Processing*, 27(8):4131–4144, 2018.

[83] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9404–9413, 2019.

[84] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, pages 2288–2295. IEEE, 2012.

[85] Andreas Krause, Pietro Perona, and Ryan G Gomes. Discriminative clustering by regularized information maximization. In *Advances in neural information processing systems*, pages 775–783, 2010.

[86] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[87] Vikash Kumar, Rohit Lal, Himanshu Patil, and Anirban Chakraborty. Conmix for source-free single and multi-target domain adaptation. In *WACV*, 2023.

[88] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4544–4553, 2020.

[89] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024.

[90] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 615–625, 2021.

[91] Mohammed Kutbi, Kuan-Chuan Peng, and Ziyan Wu. Zero-shot deep domain adaptation with common representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[92] Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *ICML*, pages 942–950, 2013.

[93] P. Langley. Crafting papers on machine learning. In Pat Langley, editor, *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

[94] Bahram Lavi, Mehdi Fatan Serj, and Ihsan Ullah. Survey on deep learning techniques for person re-identification task. *arXiv preprint arXiv:1807.05284*, 2018.

[95] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[96] Guihong Li, Hsiang Hsu, Radu Marculescu, et al. Machine unlearning for image-to-image generative models. *arXiv preprint arXiv:2402.00351*, 2024.

[97] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

[98] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *ACCV*, pages 31–44. Springer, 2012.

[99] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *CVPR*, pages 2285–2294, 2018.

[100] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[101] Yitong Li, David E Carlson, et al. Extracting relationships by multi-domain matching. In *Advances in Neural Information Processing Systems*, pages 6798–6809, 2018.

[102] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *arXiv preprint arXiv:2002.08546*, 2020.

[103] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[104] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, pages 2197–2206, June 2015.

[105] Shengcai Liao and Stan Z Li. Efficient psd constrained asymmetric metric learning for person re-identification. In *ICCV*, pages 3685–3693, 2015.

[106] Chuang Lin, Sicheng Zhao, Lei Meng, and Tat-Seng Chua. Multi-source domain adaptation for visual sentiment classification. In *AAAI*, pages 2661–2668, 2020.

[107] Yutian Lin, Xuanyi Dong, Liang Zheng, Yan Yan, and Yi Yang. A bottom-up clustering approach to unsupervised person re-identification. In *AAAI*, volume 33, pages 8738–8745, 2019.

[108] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1215–1224, 2021.

[109] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[110] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[111] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2507–2516, 2019.

[112] Jianming Lv, Weihang Chen, Qing Li, and Can Yang. Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns. In *CVPR*, pages 7948–7956, 2018.

[113] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, pages 1041–1048, 2009.

[114] Alessandro Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[115] Niki Martinel and Christian Micheloni. Re-identify people in wide area camera network. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 31–36. IEEE, 2012.

[116] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptor for person re-identification. In *CVPR*, pages 1363–1372, 2016.

[117] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptors with application to person re-identification. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[118] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors. *Machine Learning: An Artificial Intelligence Approach, Vol. I.* Tioga, Palo Alto, CA, 1983.

[119] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14765–14775, 2022.

[120] T. M. Mitchell. The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA, 1980.

[121] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703, 2019.

[122] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik Metzen. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*, 2021.

[123] Sayak Nag, Dripta S Raychaudhuri, Sujoy Paul, and Amit K Roy-Chowdhury. Learning few-shot open-set classifiers using exemplar reconstruction. *arXiv preprint arXiv:2108.00340*, 2021.

[124] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pages 931–962. PMLR, 2021.

[125] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive Skills and Their Acquisition*, chapter 1, pages 1–51. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1981.

[126] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

[127] Hyeonwoo Noh, Taehoon Kim, Jonghwan Mun, and Bohyung Han. Transfer learning via unsupervised task discovery for visual question answering. In *CVPR*, pages 8385–8394, 2019.

[128] Francesco Orabona, Claudio Castellini, Barbara Caputo, Angelo Emanuele Fiorilla, and Giulio Sandini. Model adaptation with least-squares svm for adaptive hand prosthetics. In *2009 IEEE International Conference on Robotics and Automation*, pages 2897–2903. IEEE, 2009.

[129] Samet Oymak and Talha Cihad Gulcu. Statistical and algorithmic insights for semi-supervised learning with self-training. *arXiv preprint arXiv:2006.11006*, 2020.

[130] Samet Oymak, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural architecture search with train-validation split. In *International Conference on Machine Learning*, pages 8291–8301. PMLR, 2021.

[131] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton Van Den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, pages 1846–1855, 2015.

[132] Theodoros Panagiotakopoulos, Pier Luigi Dovesi, Linus Härenstam-Nielsen, and Matteo Poggi. Online domain adaptation for semantic segmentation in ever-changing conditions. In *European Conference on Computer Vision*, pages 128–146. Springer, 2022.

[133] Rameswar Panda, Amran Bhuiyan, Vittorio Murino, and Amit K Roy-Chowdhury. Unsupervised adaptive re-identification in open world dynamic camera networks. In *CVPR*, pages 7054–7063, 2017.

[134] Rameswar Panda, Amran Bhuiyan, Vittorio Murino, and Amit K Roy-Chowdhury. Adaptation of person re-identification models for on-boarding new camera (s). *Pattern Recognition*, 96:106991, 2019.

[135] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[136] Sujoy Paul, Ansh Khurana, and Gaurav Aggarwal. Unsupervised adaptation of semantic segmentation models without source data. *arXiv preprint arXiv:2112.02359*, 2021.

[137] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. *arXiv preprint arXiv:2007.15176*, 2020.

[138] Kuan-Chuan Peng, Ziyan Wu, and Jan Ernst. Zero-shot deep domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 764–781, 2018.

[139] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[140] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054*, 2019.

[141] Jorge Luis Rivero Perez, Bernardete Ribeiro, and Carlos Morell Perez. Mahalanobis distance metric learning algorithm for instance-based data stream classification. In *IJCNN*, pages 1857–1862. IEEE, 2016.

[142] Michaël Perrot and Amaury Habrard. A theoretical analysis of metric hypothesis transfer learning. In *International Conference on Machine Learning*, pages 1708–1717, 2015.

[143] Xuelin Qian, Yanwei Fu, Yu-Gang Jiang, Tao Xiang, and Xiangyang Xue. Multi-scale deep learning architectures for person re-identification. In *ICCV*, pages 5399–5408, 2017.

[144] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[145] Dripta S Raychaudhuri, Sujoy Paul, Jeroen Vanbaar, and Amit K Roy-Chowdhury. Cross-domain imitation from observations. In *ICML*, 2021.

[146] Dripta S Raychaudhuri and Amit K Roy-Chowdhury. Exploiting temporal coherence for self-supervised one-shot video re-identification. In *European Conference on Computer Vision*, pages 258–274. Springer, 2020.

[147] Dripta S. Raychaudhuri, Yumin Suh, Samuel Schulter, Xiang Yu, Masoud Faraki, Amit K. Roy-Chowdhury, and Manmohan Chandraker. Controllable dynamic multi-task architectures. *arXiv preprint arXiv:2203.14949*, 2022.

[148] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[149] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[150] Mohammad Rostami. Lifelong domain adaptation via consolidated internal distribution. *Advances in neural information processing systems*, 34:11172–11183, 2021.

[151] Amit K Roy-Chowdhury and Bi Song. Camera networks: The acquisition and analysis of videos over wide areas. *Synthesis Lectures on Computer Vision*, 3(1):1–133, 2012.

[152] Paolo Russo, Tatiana Tommasi, and Barbara Caputo. Towards multi-source adaptive semantic segmentation. In *International Conference on Image Analysis and Processing*, pages 292–301. Springer, 2019.

[153] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, pages 213–226. Springer, 2010.

[154] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[155] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *Proceedings of the european conference on computer vision (ECCV)*, pages 687–704, 2018.

[156] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021.

[157] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pages 901–909, 2016.

[158] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.

[159] Nawid Sayed, Biagio Brattoli, and Björn Ommer. Cross and learn: Cross-modal self-supervision. In *German Conference on Pattern Recognition*, pages 228–243. Springer, 2018.

[160] Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. Mm-tta: multi-modal test-time adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16928–16937, 2022.

[161] Shashank Singh, Ananya Uppal, Boyue Li, Chun-Liang Li, Manzil Zaheer, and Barnabás Póczos. Nonparametric density estimation under adversarial losses. In *Advances in Neural Information Processing Systems*, pages 10225–10236, 2018.

[162] Junha Song, Jungsoo Lee, In So Kweon, and Sungha Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11920–11929, 2023.

[163] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[164] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *arXiv preprint arXiv:1511.05547*, 2015.

[165] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2058–2065, 2016.

[166] Ruoqi Sun, Xinge Zhu, Chongruo Wu, Chen Huang, Jianping Shi, and Lizhuang Ma. Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection. In *CVPR*, pages 4360–4369, 2019.

[167] Yifan Sun, Qin Xu, Yali Li, Chi Zhang, Yikang Li, Shengjin Wang, and Jian Sun. Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification. In *CVPR*, pages 393–402, 2019.

[168] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[169] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[170] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[171] Chiat-Pin Tay, Sharmili Roy, and Kim-Hui Yap. Aanet: Attribute attention network for person re-identifications. In *CVPR*, pages 7134–7143, 2019.

[172] Fida Mohammad Thoker and Juergen Gall. Cross-modal knowledge distillation for action recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 6–10. IEEE, 2019.

[173] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt 'em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *CVPR*, 2019.

[174] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt'em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *CVPR*, 2019.

[175] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[176] Rosanna Turrisi, Rémi Flamary, Alain Rakotomamonjy, and Massimiliano Pontil. Multi-source domain adaptation via weighted joint distributions optimal transport, 2020.

[177] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[178] Jeya Maria Jose Valanarasu, Pengfei Guo, Vibashan VS, and Vishal M Patel. On-the-fly test-time adaptation for medical image segmentation. *arXiv preprint arXiv:2203.05574*, 2022.

[179] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.

[180] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[181] Riccardo Volpi, Diane Larlus, and Grégory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4443–4453, 2021.

[182] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[183] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. Exploring domain-invariant parameters for source free domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2022.

[184] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. In *AAAI*, volume 33, pages 8933–8940, 2019.

[185] Haotian Wang, Wenjing Yang, Zhipeng Lin, and Yue Yu. Tmda: Task-specific multi-source domain adaptation via clustering embedded adversarial training. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1372–1377. IEEE, 2019.

[186] Jianrong Wang, Ziyue Tang, Xuewei Li, Mei Yu, Qiang Fang, and Li Liu. Cross-modal knowledge distillation method for automatic cued speech recognition. *arXiv preprint arXiv:2106.13686*, 2021.

[187] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *CVPR*, pages 2275–2284, 2018.

[188] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[189] Yu-Xiong Wang and Martial Hebert. Learning by transferring from unsupervised universal sources. In *AAAI*, pages 2187–2193, 2016.

[190] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

[191] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, pages 79–88, 2018.

[192] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[193] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[194] Ancong Wu, Wei-Shi Zheng, Xiaowei Guo, and Jian-Huang Lai. Distilled person re-identification: Towards a more scalable system. In *CVPR*, pages 1187–1196, 2019.

[195] Ga Wu, Masoud Hashemi, and Christopher Srinivasa. Puma: Performance unchanged model augmentation for training data removal. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 8675–8682, 2022.

[196] Yinjun Wu, Edgar Dobriban, and Susan Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning*, pages 10355–10366. PMLR, 2020.

[197] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning. In *CVPR*, pages 5177–5186, 2018.

[198] Zuxuan Wu, Xin Wang, Joseph E. Gonzalez, Tom Goldstein, and Larry S. Davis. Ace: Adapting to changing environments for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[199] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental adversarial domain adaptation for continually changing environments. In *2018 IEEE International conference on robotics and automation (ICRA)*, pages 4489–4495. IEEE, 2018.

[200] Haifeng Xia, Handong Zhao, and Zhengming Ding. Adaptive adversarial network for source-free domain adaptation. In *ICCV*, 2021.

[201] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, pages 1249–1258, 2016.

[202] Zehao Xiao, Xiantong Zhen, Ling Shao, and Cees GM Snoek. Learning to generalize across domains on single test samples. *arXiv preprint arXiv:2202.08045*, 2022.

[203] Xiaomeng Xin, Jinjun Wang, Ruji Xie, Sanping Zhou, Wenli Huang, and Nanning Zheng. Semi-supervised person re-identification using multi-view clustering. *Pattern Recognition*, 88:285–297, 2019.

[204] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003.

[205] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

[206] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1426–1435, 2019.

[207] Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM MM*, pages 188–197. ACM, 2007.

[208] Qize Yang, Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. Patch-based discriminative feature learning for unsupervised person re-identification. In *CVPR*, pages 3633–3642, 2019.

[209] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Casting a bait for offline and online source-free domain adaptation. *arXiv preprint arXiv:2010.12427*, 2020.

[210] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 1(2):5, 2020.

[211] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *arXiv preprint arXiv:2110.04202*, 2021.

[212] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8978–8987, 2021.

[213] Wenjie Yang, Houjing Huang, Zhang Zhang, Xiaotang Chen, Kaiqi Huang, and Shu Zhang. Towards rich feature discovery with class activation maps augmentation for person re-identification. In *CVPR*, pages 1389–1398, 2019.

[214] Xun Yang, Meng Wang, Richang Hong, Qi Tian, and Yong Rui. Enhancing person re-identification in a self-trained subspace. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(3):27, 2017.

[215] Xun Yang, Meng Wang, and Dacheng Tao. Person re-identification with metric learning using privileged information. *IEEE Transactions on Image Processing*, 27(2):791–805, 2017.

[216] Yang Yang, Shengcai Liao, Zhen Lei, and Stan Z Li. Large scale similarity learning using similar pairs for person verification. In *AAAI*, pages 3655–3661, 2016.

[217] Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023.

[218] Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10665–10673, 2021.

[219] Hai Ye, Qizhe Xie, and Hwee Tou Ng. Multi-source test-time adaptation as dueling bandits for extractive question answering. *arXiv preprint arXiv:2306.06779*, 2023.

[220] Jingwen Ye, Yifang Fu, Jie Song, Xingyi Yang, Songhua Liu, Xin Jin, Mingli Song, and Xinchao Wang. Learning with recoverable forgetting. In *European Conference on Computer Vision*, pages 87–103. Springer, 2022.

[221] Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *WACV*, 2021.

[222] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[223] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *CVPR*, pages 5704–5713, 2019.

[224] Fuming You, Jingjing Li, and Zhou Zhao. Test-time batch statistics calibration for covariate shift. *arXiv preprint arXiv:2110.04065*, 2021.

[225] Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. Cross-view asymmetric metric learning for unsupervised person re-identification. In *ICCV*, pages 994–1002, 2017.

[226] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. Unsupervised person re-identification by soft multilabel learning. In *CVPR*, pages 2148–2157, 2019.

[227] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722, 2018.

[228] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, 2019.

[229] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pages 8559–8570, 2018.

[230] Long Zhao, Xi Peng, Yuxiao Chen, Mubbasir Kapadia, and Dimitris N Metaxas. Knowledge as priors: Cross-modal knowledge generalization for datasets without superior knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6528–6537, 2020.

[231] Sicheng Zhao, Bo Li, Pengfei Xu, and Kurt Keutzer. Multi-source domain adaptation in the deep learning era: A systematic survey. *arXiv preprint arXiv:2002.12169*, 2020.

[232] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, pages 1116–1124, 2015.

[233] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

[234] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *CVPR*, pages 2138–2147, 2019.

[235] Sanping Zhou, Jinjun Wang, Jiayun Wang, Yihong Gong, and Nanning Zheng. Point to set similarity based deep feature learning for person re-identification. In *CVPR*, pages 3741–3750, 2017.

[236] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[237] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5989–5996, 2019.