Unsupervised Domain Adaptation and Contrastive Learning for insufficiently labeled data

A

Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In partial fulfillment

of the requirement for the degree of

DOCTOR OF PHILOSOPHY

by

Nazanin Moradinasab

May 2024

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Nazanin Moradinasab

This Dissertation has been read and approved by the examing committee:

Advisor: Donald E. Brown

Advisor:

Committee Member: Tariq Iqbal

Committee Member: Afsaneh Doryab

Committee Member: Gary K. Owens

Committee Member: Michael Porter

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

May 2024

Abstract

Recently, modern deep learning-based approaches have become popular over traditional methods in many real-world applications. However, the success of these approaches relies on two factors: (1) access to the massive amount of labeled data for training and (2) independent and identically distributed (i.i.d) assumption of training and test datasets. In many applications, collecting a large amount of high-quality labeled data is expensive and financially demanding, especially for tasks like semantic segmentation and multivariate time series classification. The majority of practical datasets are only partially labeled or possess limited labeled instances.

The main goal of this dissertation is to develop robust deep learning models for situations where the target dataset is labeled insufficiently. To achieve this goal, we developed four innovative approaches, two of which are the Universal representation learning and Label-efficient Contrastive learning-based models. These models are designed for time series classification and semantic segmentation tasks where the datasets are insufficiently labeled. A distinctive feature of our methods is the introduction of a cluster-level Supervised Contrastive (SupCon) approach in addition to the instance-level SupCon. This addition aims to mitigate the negative impact caused by intra-class variances and inter-class similarities during the training process. By incorporating both instance and cluster-level contrastive learning, our approach seeks to enhance the model's ability to discern meaningful patterns and representations, particularly in scenarios where labeled data is scarce. The third approach focuses on self-training Domain Adaptation (DA) techniques to improve the generalization ability of the deep models on the unlabeled or scarce-labeled target tasks by training the model on both label-scarce target and label-rich source data. The prevalent self-training approach involves retraining the dense discriminative classifier of $p(class|pixelfeature)$ using the pseudo-labels from the target domain. While many

methods focus on mitigating the issue of noisy pseudo-labels, they often overlook the underlying data distribution $p(pixel feature|class)$ in both the source and target domains. To address this limitation, we designed the multi-prototype Gaussian-Mixture-based (ProtoGMM) model, which incorporates the Gaussian mixture model into contrastive losses to perform guided contrastive learning. This novel approach involves estimating the underlying multi-prototype source distribution by utilizing the Gaussian Mixture model on the feature space of the source samples. The components of the GMM model act as representative prototypes, effectively adapting to the multimodal data density and capturing within-class variations. To achieve increased intra-class semantic similarity, decreased inter-class similarity, and domain alignment between the source and target domains, we employed multi-prototype contrastive learning between source distribution and target samples. The fourth developed approach is the Generalized Gaussian mixture-based (GenGMM) Domain Adaptation Model which was designed for the Generalized Domain Adaptation (GDA) task. While significant efforts have been devoted to improving unsupervised domain adaptation for this task, it's crucial to note that many promising domain adaptation models rely on a strong assumption: the source data is entirely and accurately labeled, while the target data is unlabeled. In real-world scenarios, however, we often encounter partially or noisy labeled data in source and target domains, referred to as the Generalized Domain Adaptation (GDA) setting. In such cases, we leveraged weak or unlabeled data from both domains to narrow the gap between them, leading to more effective adaptation. To facilitate this, we introduce the GenGMM Domain Adaptation Model, which harnesses the underlying data distribution in both domains to refine noisy weak and pseudo labels.

All developed approaches compared to the current state-of-the-art (SOTA) approaches across different well-known benchmarks including, 1) The UEA multivariate time series classification archive, 2) The cardiopulmonary exercise testing (CPET) dataset, 3) The immunofluorescent images, and 4) The benchmarks of urban scenes including GTA5 to Cityscapes, Synthia to Cityscapes, and Cityscapes to Dark Zurich. The results demonstrate that our framework yields substantial improvements when compared to existing approaches.

Acknowledgements

First and foremost, I want to express my deepest appreciation to my advisor, Professor Donald E. Brown. His unwavering patience, insightful suggestions, and invaluable support have been instrumental throughout my years of study and the research process. Professor Brown never ceased to encourage and propel me forward. His constructive feedback not only sharpened my thinking but also elevated the quality of my work. I extend a heartfelt thank you for his enduring support.

I also extend sincere gratitude to my Ph.D. committee members for their valuable advice and insightful comments on this dissertation. Furthermore, I would like to thank Dr. Laura Shankman, Dr. Rebecca Deaton, Dr. Dan M. Cooper, and Dr. Sana Syed for their support, feedback, and advice throughout my Ph.D. journey.

A special acknowledgment goes to my beloved family. To my mother and father, words cannot adequately convey my gratitude for the sacrifices you've made on my behalf. Your prayers have been a sustaining force, shaping me into the person I am today. To my sister, your unconditional love and support are priceless. Your constant presence and support have left an indelible mark for which I am forever indebted.

Finally, and of utmost significance, I want to express my deepest thanks to my beloved husband, Hassan. His love and unwavering support have been a constant source of encouragement throughout this entire journey. Hassan, you have been my best friend and a steadfast companion since you entered my life. I am profoundly grateful for the countless sacrifices, support, and invaluable discussions that steered me away from wrong turns and helped me reach this significant milestone.

Contents

Chapter 1

Introduction

Recently, deep learning-based methods which are a subfield of machine learning have achieved promising performance in real-world problems in different domain applications. These methods contain the hierarchical architectures that learn the multiple levels of distributed representations, i.e. intermediate representations. The advantage of these methods over the conventional machine learning approaches is their capability to learn both low-dimensional feature representations and a prediction model in an end-to-end fashion, simultaneously [1]. In addition, they require less domain knowledge compared to traditional methods. Despite the prevalence of supervised deep learning approaches in real-world tasks, the success of these approaches hinges on (1) access to the massive amount of labeled data for training and (2) independent and identically distributed (i.i.d) assumption of training and test datasets [2]. However, in many application domains, such as semantic segmentation or multivariate time series classification, achieving a large amount of high-quality reliable labeled data is labor-expensive, error-prone, and time-consuming to train accurate deep models. Recently, new groups of techniques, such as Weakly-supervised Learning (WSL), Unsupervised Domain adaptation (UDA), and semi-supervised learning (SSL) approaches, have emerged as vital solutions to situations where only limited or insufficiently labeled data is available. These approaches focus on building more robust models that learn from fewer labeled samples and/or with better out-of-distribution generalization [3, 4].

In the following sections, we provide an introduction to the current literature and recent advances in three crucial areas: multivariate time series data classification, nuclei detection and classification,

and domain adaptation. We delve into the existing knowledge, methodologies, and breakthroughs in these domains, shedding light on their significance. Moreover, we explore potential challenges that researchers and practitioners may face, particularly when dealing with insufficiently labeled data. Finally, we outline the structure of the dissertation, offering a roadmap for the subsequent chapters and discussions.

1.1 Multivariate Time Series Classification

The goal of Time Series Classification (TSC) is to predict the class label for a given time series data, which is a sequence of real-value observations ordered by time. While most state-of-the-art methods for TSC have focused on univariate TSC, where each case consists of a single series (i.e., one dimension), real-world time series datasets in many applications are multivariate—containing multiple dimensions but a single label. With the advancement of sensor technologies, the Multivariate Time Series Classification (MTSC) problem has received great attention in a wide range of research domains and applications such as Human Activity Recognition [5], EEG/ECG data analysis [6], and Motion Recognition [7].

An ideal TSC method should be accurate, efficient, and interpretable. However, even accurate state-of-the-art TSC models suffer from a lack of interoperability or efficiency. Most general TSC approaches involve a preliminary learning phase to extract feature candidates from the time series data, such as a bag of patterns [8] or time series shapelet [9]. These methods become less computationally efficient when dealing with long-time series data as selecting features from a larger feature space increases the computational complexity of the model. The challenge is amplified in the multivariate case, where feature selection from a vast feature space becomes more difficult [10]. Recently, ensemble methods have achieved high accuracy for TSC tasks, while their computational complexity increases with the number of time steps and dimensions. For instance, the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTE) [11], has high training complexity $O(N^2 \cdot T^4)$, as highlighted by [12], where T represents the length of the series and N is the number of dimensions. The latest version, HIVE-COTE v2.0 [13], for multivariate data requires a substantial run time [14]. However, studies indicate that deep learning models significantly surpass HIVE-COTE in terms of run

time. Importantly, these methods do not provide interpretable results.

Recently, deep learning-based methods with cross-entropy loss function have demonstrated promising performance in TSC tasks (e.g. ResNet [15], Inception [16]). One of the main advantages of the deep learning approaches is their capability to manage large feature spaces by learning low-dimensional feature representations [10]. Moreover, these approaches require less domain-specific knowledge compared to the traditional methods for handling time series data. However, these advantages come at the cost of a substantial requirement for a large amount of labeled data during training, posing challenges when dealing with time series data that has limited labeling. Zhang *et al.* (2020) [10] suggested that the traditional TSC models can effectively mitigate the issue of limited data by using distance-based methods. They proposed the TapNet deep learning model [10] with a distance-based loss function instead of a cross-entropy loss function to address the issue of limited data.

In this dissertation, we designed the *Supervised Contrastive learning* for *Time Series Classification* (SupCon-TSC) model to enable deep learning models to handle limited labelings in TSC tasks while learning the low-dimensional feature representations. It is based on Supervised Contrastive learning (SupCon) and provides interpretable outcomes.

1.2 Nuclei Detection and Classification in 3D Cardiovascular Immunofluorescent Images

Two major causes of death in the United States and worldwide are stroke and myocardial infarction (MI) [17]. The underlying cause of both is thrombi released from ruptured or eroded unstable atherosclerotic plaques that occlude vessels in the heart (MI) or the brain (stroke) [18, 19]. Unstable plaques are more prone to rupture or erosion, leading to possible MI or stroke. Human morphological studies have shown that the critical factor of plaque stability is plaque composition rather than lesion size [20, 21]. Virmani *et al.* (2000) [17] have extensively studied the composition of human lesions and established that lesions with a thin extracellular matrix (ECM)- rich protective fibrous cap and a predominance of CD68+ relative to ACTA2+ cells, presumed to be macrophages (MΦ) and smooth muscle cells (SMC), respectively, are prone to plaque rupture [22]. Another study showed that loss of

endothelial cells (EC) overlying lesions, and increased CD31+ ACTA2+ cells assumed to be EC that have undergone EC to mesenchymal transition (EndoMT), are prone to erosion. However, lineage tracing studies in [23] highlighted that more than 80% of SMCs in advanced mouse atherosclerotic lesions no longer had detectable ACTA2. Further, a subset of these cells expressed LGALS3, a marker that would have traditionally classified them as a (MΦ).

Plaque composition can be determined by immunofluorescent staining of histological cross-sections from diseased vessels. These morphological studies are extremely valuable for understanding the underlying mechanisms for plaque rupture and determining the mechanisms that promote atherosclerotic plaque stability. However, to determine the plaque composition from the immunofluorescent images, we need to first accurately detect cells and classify them based on co-expressed markers. This process requires hours of manual detection of the various cell types, which is slow, expensive, and prone to human error. The challenges of manual cell counting indicate a need for automated image processing to localize and count various cell types in order to find their distribution in fluorescent microscopy images, and thereby classify lesions as stable or unstable. Several challenges arise when designing automated image analysis, such as the heterogeneity of cell types (shape and size), autofluorescent signal from the tissue, and low image contrast [24]. Moreover, automatic localization and counting of various cell types in 3D immunofluorescent images have the added challenge of overlapping cells and cellularly dense regions that are difficult to count. In addition, there can be variability in the depth of imaging and thickness of the sample tissue itself.

Recently, modern deep learning-based nuclei segmentation approaches [25, 26, 27, 28, 29] have become popular over traditional methods [30] to quantify the histopathology and fluorescent microscopy images. However, these neural networks are usually categorized as fully supervised approaches that require a large amount of pixel-wise annotated data for training. Collecting pixel-wise annotated data is expensive, time-consuming, and difficult because it requires classification of every pixel in an image and is near impossible to perform on 3D images. Alternatively, adapting weak-annotation methods such as labeling each nucleus with a point reduces the burden of pixel-wise annotations. Several studies [31, 32, 33] have tried to address training neural networks based on point annotations but focus on the nuclei segmentation problem using point annotations. In these approaches, as point annotations alone are not sufficient to train a neural network model, the authors take advantage of the original

images and the shape of nuclei, among others, to get extra information to train the model. None of these studies pay attention to the nucleus' boundary [34], even though it plays a key role to separate clustered nuclei. While all these studies focus on developing weakly supervised models for nuclei detection in 2D images, our study is the first to leverage weakly annotated data for 3D multi-channel Immunofluorescent images.

This dissertation developed the *Label-efficient Contrastive learning-based* (LECL) model to detect and classify various types of nuclei in 3D immunofluorescent images using weak annotations (i.e., point annotations). Developing and training weakly-supervised learning models for 3D images is a challenging task because these images contain multiple channels (z-axis) for nuclei and different markers separately, which makes training using point annotations difficult. Previous methods use Maximum Intensity Projection (MIP) to convert immunofluorescent images with multiple slices to 2D images, which can cause signals from different z-stacks to falsely appear associated with each other. To overcome this we devised a novel approach called Extended Maximum Intensity Projection (EMIP) that addresses issues using MIP. Moreover, in order to enhance the model's performance in a weak setting, the suggested method incorporates a semi-supervised learning approach. This involves applying entropy minimization loss specifically over the boundaries of nuclei, treating them as unlabeled areas. Moreover, the suggested approach incorporates a supervised contrastive learning method to improve the model's performance in nuclei classification metrics.

1.3 Unsupervised Domain Adaptation Model for Semantic Segmentation

In recent years, there has been remarkable progress in semantic segmentation, a technique that assigns semantic class labels to each pixel in an image. However, achieving the generalization of deep neural networks to unseen domains is vital for critical applications, including autonomous driving [35, 36], and medical analysis [37]. Unfortunately, this progress heavily depends on acquiring large-scale pixel-level annotations, a costly and time-consuming process when done manually.

To address this challenge, researchers have been exploring alternative approaches like generating

simulated data or leveraging out-of-domain (source) annotations to reduce manual effort and improve neural network applicability across domains. However, domain shift remains a major obstacle, leading to a performance decline when applying a well-trained model from the source domain to the target domain. To tackle this issue, a solution known as unsupervised domain adaptation (UDA) has been developed, which transfers knowledge from a label-rich source domain (synthetic) to a label-scarce target domain (real) [38]. Recent trends in UDA for semantic segmentation have led to two main approaches: domain alignment and self-training.

Domain alignment methods employ adversarial learning training algorithms to reduce the domain shift in various spaces, including image level [39], feature level [40], or output level [41]. While these methods can bring the two domains closer together on a global scale, they do not ensure that the feature representations for different classes in the target domain are sufficiently discriminative. Consequently, this limitation hampers the model's overall capacity for image segmentation.

Alternatively, self-training methods aim to utilize target-specific knowledge by selecting high-confidence pseudo-labels in the target domain for the next round of training. Despite the promising performance, these methods suffer from significant limitations. The segmentation model tends to be biased toward the source domain, leading to error-prone pseudo-labels for the target domain. Moreover, relying solely on highly confident predictions offers limited supervision information during model training. Effectively addressing noisy labels and managing bias toward the source domain is vital for ensuring the effectiveness of self-training and achieving superior performance in the target domain. Some studies tackle the issue of noisy pseudo labels by implementing techniques such as confidence estimation [42, 43], consistency regularization [44], or label denoising [45, 46]. These approaches aim to alleviate the impact of noise in the pseudo labels to improve the overall performance.

In many papers, a common approach to train the classifier is by using cross-entropy loss. However, cross-entropy loss primarily focuses on bringing similar features together and does not effectively differentiate features across distinct classes. As a result, it is crucial to ensure that features corresponding to different classes are properly separated while aggregating features belonging to the same class in the latent space. In this context, contrastive learning emerges as a relevant topic, allowing models to learn meaningful visual representations by comparing diverse unlabeled data [47]. One straight-

forward approach for contrastive learning involves employing a memory bank mechanism. During training, this memory bank is updated by adding the averaged features of each category from the current source image while removing the oldest ones. However, this method may be susceptible to class biases, as it updates underrepresented classes (e.g., truck, bus) less frequently, and it can be computationally expensive. Alternatively, another approach is to use global prototypes, which are the averaged features of each category across the entire source domain. However, this approach has its drawbacks. It tends to overlook variations in certain attributes (e.g., shape, color, illumination), potentially reducing the discriminability of the learned features. Furthermore, it relies on the unimodality assumption of each category. While contrastive learning losses have shown improvements in domain adaptation, their effectiveness is constrained by their dependence on pseudo-labels generated using a discriminative classifier trained with the cross-entropy loss function. During the early stages of training, the backpropagation signals originating from the contrastive loss can be excessively high, especially when the model's predictions are highly unreliable. This situation negatively impacts the overall performance. To mitigate this issue, Vayyat *et al.* (2022) [48] introduce a weight on the loss based on the confidence of the teacher-network predictions. This approach helps in stabilizing the training process and improving overall performance. However, this teacher network is a discriminative classifier that is biased toward the source domain data. Moreover, these discriminative classifiers suffer from various limitations: 1) neglecting to model the underlying data distribution, 2) unimodality for each class, and 3) these models suffer from accuracy degradation away from decision boundaries, hampering adaptation for critical tasks [49].

In this dissertation, to address these challenges we present the *Multi-prototype Gaussian-Mixture-based model* (ProtoGMM) that overcomes the limitations of existing methods. Unlike prevailing self-training approaches that focus solely on the discriminative classifier of $p(class|pixel feature)$, ProtoGMM adopts a hybrid training approach, integrating both discriminative and generative models.

1.4 Generalized Domain Adaptation Model for Semantic Segmentation

To reduce annotation tasks, the Unsupervised Domain Adaptation (UDA) techniques [50, 51, 52] aim to leverage the valuable knowledge from a labeled source domain to enhance learning in another unlabeled target domain [53, 54]. However, UDA methods assume fully labeled source data and completely unlabeled target domains, which is often not the case in practice due to partial or noisy labels in both domains. In such cases, we suggest leveraging additional weak or unlabeled data from both the source and target domains to enhance UDA performance by narrowing the gap between these two domains. Therefore, we introduce a novel domain adaptation setting called Generalized Domain Adaptation (GDA) which possesses the following characteristics: 1) Partially or noisy labeled source data, 2) Weakly or unlabeled target data.

The GDA setting relaxes the problem of UDA by allowing the use of unlabeled or weakly labeled data from the source domain and weak labels from the target domain. Nevertheless, effectively leveraging these unlabeled or weakly labeled source data and weakly labeled target data is a non-trivial task. There is limited research that focuses on the incorporation of weak labels from the target domain. Even though Akata *et al.* (2022) [55] and paul *et al.* (2020) [56] introduce weak labels from the target domain as supplementary sources of supervision, they do not employ them to align features between the source and target domains. In contrast, Das *et al.* (2023) [57] proposed a novel approach, utilizing class prototypes generated by exploiting these weak labels. These prototypes were employed for both intra-domain feature alignment within individual domains and inter-domain feature alignment between the source and target domains, effectively reducing the domain gap. However, it's crucial to highlight that prototype-based approaches, relying on the unimodality assumption within each category, may not fully account for variations in specific attributes like shape, color, or illumination [49]. This oversight could potentially diminish the distinguishability of the acquired features. Furthermore, despite limited studies on weekly supervised domain adaptation, the incorporation of unlabeled/weakly labeled source data remains an underexplored area.

This dissertation addresses domain adaptation challenges in GDA settings with partially labeled

source and target data. A common strategy is self-training, iteratively refining predictions on unlabeled source and target data, but it's important to note that the pseudo-labels generated in this process can be noisy. To tackle these issues, we introduce a Generalized Gaussian mixture-based (GenGMM) Domain Adaptation Model, leveraging the source and target domain distributions to enhance the quality of weak and pseudo labels and achieve alignment between the source and target domains.

1.5 Dissertation Outline

This dissertation starts with a literature review that provides background on time series classification, nuclei detection and classification, and domain adaptation. Following this, it includes four chapters that highlight our contributions to the literature, and their summaries along with corresponding contributions are as follows:

1.5.1 Universal Representation Learning for Multivariate Time Series using the instance-level and cluster-level Supervised Contrastive Learning

Abstract: *The Multivariate Time Series Classification (MTSC) task aims to predict a class label for a given time series. Recently, modern deep learning-based approaches have achieved promising performance over traditional methods for MTSC tasks. The success of these approaches relies on access to the massive amount of labeled data (i.e., annotating or assigning tags to each sample that shows its corresponding category). However, obtaining a massive amount of labeled data is usually very time-consuming and expensive in many real-world applications such as medicine, because it requires domain experts' knowledge to annotate data. Insufficient labeled data prevents these models from learning discriminative features, resulting in poor margins that reduce generalization performance. To address this challenge, we developed a novel approach: Supervised Contrastive learning for Time Series Classification (SupCon-TSC). This approach improves the classification performance by learning the discriminative low-dimensional representations of multivariate time series, and its end-to-end structure allows for interpretable outcomes. It is based on Supervised Contrastive (SupCon) loss to learn the inherent structure of multivariate time series. First, two separate augmentation families, includ-*

ing strong and weak augmentation methods, are utilized to generate augmented data for the source and target networks, respectively. Second, we developed the instance-level, and cluster-level SupCon learning approaches to capture contextual information to learn the discriminative and universal representation for multivariate time series datasets. In the instance-level SupCon learning approach, for each given anchor instance that comes from the source network, the low-variance output encodings from the target network are sampled as positive and negative instances based on their labels. However, the cluster-level approach is performed between each instance and cluster centers among batches, as opposed to the instance-level approach. The cluster-level SupCon loss attempts to maximize the similarities between each instance and cluster centers among batches. We tested this novel approach on two small cardiopulmonary exercise testing (CPET) datasets and the real-world UEA Multivariate time series archive. The results of the SupCon-TSC model on CPET datasets indicate its capability to learn more discriminative features than existing approaches in situations where the size of the dataset is small. Moreover, the results on the UEA archive show that training a classifier on top of the universal representation features learned by our developed method outperforms the state-of-the-art approaches.

1.5.2 Label-efficient Contrastive Learning-based model for nuclei detection and classification in 3D Cardiovascular Immunofluorescent Images

Abstract: *Recently, deep learning-based methods achieved promising performance in nuclei detection and classification applications. However, training deep learning-based methods requires a large amount of pixel-wise annotated data, which is time-consuming and labor-intensive, especially in 3D images. An alternative approach is to adapt weak-annotation methods, such as labeling each nucleus with a point, but this method does not extend from 2D histopathology images (for which it was originally developed) to 3D immunofluorescent images. The reason is that 3D images contain multiple channels (z-axis) for nuclei and different markers separately, which makes training using point annotations difficult. To address this challenge, we designed the Label-efficient Contrastive learning-based (LECL) model to detect and classify various types of nuclei in 3D immunofluorescent images. Previous methods use Maximum Intensity Projection (MIP) to convert immunofluorescent images with multiple slices to 2D images, which can cause signals from different z-stacks to falsely appear associated with each*

other. To overcome this, we devised an Extended Maximum Intensity Projection (EMIP) approach that addresses issues using MIP. Furthermore, we performed a Supervised Contrastive Learning (SCL) approach for weakly supervised settings. We conducted experiments on cardiovascular datasets and found that our framework is effective and efficient in detecting and classifying various types of nuclei in 3D immunofluorescent images.

1.5.3 ProtoGMM: Multi-prototype Gaussian-Mixture-based Domain Adaptation Model for Semantic Segmentation

Abstract: *Domain adaptive semantic segmentation aims to generate accurate and dense predictions for an unlabeled target domain by leveraging a supervised model trained on a labeled source domain. The prevalent self-training approach involves retraining the dense discriminative classifier of $p(class|pixelfeature)$ using the pseudo-labels from the target domain. While many methods focus on mitigating the issue of noisy pseudo-labels, they often overlook the underlying data distribution $p(pixelfeature|class)$ in both the source and target domains. To address this limitation, we developed the multi-prototype Gaussian-Mixture-based (ProtoGMM) model, which incorporates the Gaussian mixture model into contrastive losses to perform guided contrastive learning. Contrastive losses are commonly executed in the literature using memory banks, which can lead to class biases due to underrepresented classes. Furthermore, memory banks often have fixed capacities, potentially restricting the model's ability to capture diverse representations of the target/source domains. An alternative approach is to use global class prototypes (i.e. averaged features per category). However, the global prototypes are based on the unimodal distribution assumption for each class, disregarding within-class variation. To address these challenges, we designed the ProtoGMM model. This novel approach involves estimating the underlying multi-prototype source distribution by utilizing the Gaussian Mixture model on the feature space of the source samples. The components of the GMM model act as representative prototypes, effectively adapting to the multimodal data density and capturing within-class variations. To achieve increased intra-class semantic similarity, decreased inter-class similarity, and domain alignment between the source and target domains, we employ multi-prototype contrastive learning between source distribution and target samples. The experiments show the effectiveness of our method on UDA benchmarks.*

1.5.4 GenGMM: Generalized Gaussian-Mixture-based Domain Adaptation Model for Semantic Segmentation

Abstract: *Domain adaptive semantic segmentation is the task of generating precise and dense predictions for an unlabeled target domain using a model trained on a labeled source domain. While significant efforts have been devoted to improving unsupervised domain adaptation for this task, it's crucial to note that many promising domain adaptation models rely on a strong assumption: that the source data is entirely and accurately labeled, while the target data is unlabeled. In real-world scenarios, however, we often encounter partially or noisy labeled data in source and target domains, referred to as Generalized Domain Adaptation (GDA). In such cases, we suggest leveraging weak or unlabeled data from both domains to narrow the gap between them, leading to more effective adaptation. To facilitate this, we introduce the Generalized Gaussian mixture-based (GenGMM) Domain Adaptation Model, which harnesses the underlying data distribution in both domains to refine noisy weak and pseudo labels. Our experiments across different benchmarks of cell-type adaptation and urban scenes demonstrate that our framework yields substantial improvements when compared to existing approaches.*

Chapter 2

Literature Review

2.1 Introduction

In this section, we first provide an overview of relevant studies related to time series classification and image segmentation, especially in cases where the dataset is insufficiently labeled or small. We then proceed to recent developments and state-of-the-art domain adaptation techniques to outline relevant methods.

2.2 Multivariate Time Series Classification

In this section, we discuss relevant related work in the area of time-series classification. The state-of-the-art MTS classifiers are generally categorized into three groups: similarity-based, feature-based, and deep learning methods.

The similarity-based approaches typically utilize a similarity function such as Euclidean distance [58], edit distance [59], wavelets [60], and Dynamic Time Warping (DTW) [61] to measure the similarity between two instances. In these approaches, the new time series instance is classified best on its similarity to the top-k neighbors in the historical data. DTW is the most popular distance function, and two versions of it for MTSC are the independent (DTW_I) and dependent approaches (DTW_D)

[62]. The independent strategy defines a different point-wise distance matrix for each dimension and then sums them up. In contrast, the dependent strategy performs warping over all the given dimensions simultaneously by calculating the Euclidean distance between vectors containing all dimensions.

On the other hand, conventional feature-based classification methods involve the manual design of feature extraction algorithms combined with machine learning models for classification. Based on the literature, Shapelets-based (gRSF [63] and UFS [64]) and Bag of Word-based classifiers (LPS [65], mv-ARF [66], SMTS [67] and WEASEL+MUSE [68]) are two popular feature-based algorithms. To classify time series data, Shapelets-based models transform the original time series into a lower-dimensional space by using subsequences. However, Bag of Word-based classifiers perform the classification by converting time series into a Bag of Words (BoW) and building a classifier upon the BoW representation. Recently, the WEASEL+MUSE [68] model, which uses the bag of Symbolic Fourier Approximation (SFA) symbol model, outperforms gRSF, LPS, mv-ARF, SMTS, and UFS. However, both shapelets-based and BoW-based methods are computationally expensive and have a long learning process [69].

Recently, deep learning techniques (XCM [70], FCN [15], MLSTM-FCN [71], MTEX-CNN [72], ResNet [15], and TapNet [10]) have been used extensively for time series classification. These techniques offer the advantage of automatically extracting the important features from time-series data for classification, as opposed to the feature-based methods listed above that require significant manual effort. However, a large amount of data is needed to train these models. These techniques commonly contain the stack of CNN layers and LSTM layers to extract features along with the softmax layer to predict the label. We describe these techniques briefly below. However, Ismail *et al.* (2019) [73] provides a more elaborate survey. Karim *et al.* (2019) [71] proposed a model named MLSTM-FCN which consists of an LSTM layer and a stacked CNN layer to extract features.

Assaf *et al.* (2019) [72] proposed MTEX-CNN, which utilizes a sequence of 2D and 1D convolution filters to extract MTS features corresponding to the observed variables and time, respectively. However, this model has some limitations which have been addressed by [70]. Fauvel *et al.* (2021) [70] propose the XCM model, which uses the 2D and 1D convolution filters parallelly over the input data to extract features corresponding to observed variables and time, separately.

Even though deep learning-based methods can learn the latent features by training convolutional or recurrent networks, they require large-scale labeled data. Recently, Zhang *et al.* (2020) [10] proposed the TapNet model with a distance-based loss function instead of a cross-entropy loss function to address the issue of limited data. None of the existing work addresses the problem of the limited labeled data, except TapNet.

2.3 Nuclei Detection and Classification

Deep learning algorithms have achieved great success in medical image segmentation/ classification [74], quantifying both histopathology and fluorescent microscopy images. However, these methods typically require a large amount of pixel-level annotations, which can be time-consuming and labor-intensive to obtain, especially when dealing with 3D images. Hence, we used point-level annotation to accurately detect and segment the nuclei while reducing the annotation burden. Below we provide an overview of some of the recently proposed 1) Nuclei instance segmentation techniques and 2) Weakly Supervised Image Segmentation using Point annotation.

2.3.1 Nuclei Instance Segmentation

Due to the small size of nuclei and their overlapping structures, nuclei instance segmentation is a challenging task. As a result, different strategies for separating nuclear boundaries have been proposed in the literature. Kumar *et al.* (2017) [26] incorporated boundary pixels with nuclei and background for the segmentation model training and performed anisotropic region growing as a post-processing step. Kang *et al.* (2019) [75] further extended the three-class segmentation approach by using it as an intermediate task for estimating coarse boundaries followed by fine-grained segmentation. Naylor *et al.* (2017) [27] formulated the segmentation problem as a regression task of the distance map for separating the touching or overlapping nuclei. Schmidt *et al.* (2018) [76] used star-convex polygons for localizing cell nuclei. Another branch of approach that has shown promising results uses auxiliary task learning to separate overlapping nuclei. Chen *et al.* (2017) [77] proposed a deep contour-aware network integrating instance appearance and contour information into a multi-task learning framework and a

weighted auxiliary classifier to address the vanishing gradient problem. Oda *et al.* (2018) [78], in their Boundary-Enhanced Segmentation Network, added another decoding path in the U-Net architecture for enhancing the boundaries of cells. Liu *et al.* (2019) [79] designed a dual-branch segmentation model integrating the auxiliary semantic segmentation branch with the instance segmentation branch via a feature fusion mechanism. Zhou *et al.* (2019) [80] proposed a Contour-aware informative aggregation network aggregating the spatial and texture dependencies of nuclei and contour in the decoder's bi-directional feature aggregation module. Hover-Net, one of the popular methods for instance segmentation and classification, used horizontal and vertical distance maps to the nuclear center with segmentation and classification maps for learning. Most recently, He *et al.* (2021) [81] learned the spatial relationship between nucleus pixels via the centripetal direction feature. These direction features were then used to separate instances.

2.3.2 Weakly Supervised Image Segmentation using Point annotation

Since Bearman *et al.* (2015) [82] proposed point annotations for semantic segmentation and established it as an effective strategy for object detection and counting tasks, it has been extended to other domains, including medical imaging and nuclei segmentation. Zhou *et al.* (2018) [83] designed architecture with sibling branches for cell nuclei detection and classification tasks and trained them using centroid point annotations. Yoo *et al.* (2019) [84] introduced an auxiliary task, Pseudoegnet, for accurately detecting nuclei boundaries without edge annotations. Nishimura *et al.* (20119) [33] used contribution pixel analysis in the centroid detection network for instance segmentation. They used guided backpropagation focusing on particular regions for determining the contributing pixels for a predicted centroid. Qu *et al.* (2015) [31] generated the Voronoi label and cluster label from the point label and used them to train the U-Net model with CRF loss for segmentation. In the follow-up to this work, they tackled a more challenging scenario of partial point annotation and used a two-stage learning framework for nuclei segmentation [29]. In the first stage, they used self-training for generating nuclei annotation for the unlabeled region, followed by their weakly supervised segmentation module. Chamanzar *et al.* (2020) [32] used Voronoi transformation, local pixel clustering, and repel encoding for generating pixel-level labels for U-Net training via a multi-task scheduler. Tian *et al.* (2020) [34] proposed a coarse-to-fine two-staged training framework. In the first stage for generating

coarse maps, they employed an iterative self-supervision strategy for generating high confidence point-distance maps along with Voronoi edge distance maps for training. Further, in the second stage, they refined predictions by incorporating contour-sensitive constraints.

Most of the existing weak approaches in the literature have predominantly concentrated on 2D analysis, with relatively less emphasis on 3D immunofluorescent images. While numerous studies have been dedicated to the development of weakly supervised models for nuclei detection in 2D images, our research marks a significant departure. It stands as the pioneering effort in leveraging weakly annotated data while meticulously preserving the intrinsic 3D nature of the images.

2.4 Unsupervised Domain Adaptation

Because reliable labels are not available for various applications, there is a strong demand to apply the trained model over the label-rich source domain to the label-scarce domain. However, the performance of the trained model can be severely dropped on the target domain because of the domain shift. Unsupervised domain adaptation technique (UDA) which is a special case of transfer learning as shown in Figure 2.1 aims to mitigate the domain shift between source and target domain [85].

2.4.1 UDA Definition

In the UDA, $p_s(x, y) \in p_{\mathcal{S}}$ and $p_t(x, y) \in p_{\mathcal{T}}$ are the underlying source and target domain distributions, respectively. Then, the labeled data $\mathcal{D}_S$ is sampled i.i.d from the source domain distribution (i.e. $p_s(x, y)$) and unlabeled data $\mathcal{D}_T$ is selected i.i.d from marginal target domain distribution (i.e. $p_t(x)$). The UDA aims to train the model on both $\mathcal{D}_T$ and $\mathcal{D}_S$ to improve the performance of the trained model on the target domain. $\mathcal{Y} = 1, 2, ..., c$ is the set of the class label. The UDA is motivated by the following Theorem [85]:

Theorem 1. *For a hypothesis h*
$\mathcal{L}_t(h) \leqslant \mathcal{L}_s(h) + d[p_{\mathcal{S}}, p_{\mathcal{T}}] + min[\mathbb{E}_{x\sim p_s}|p_s(y|x) - p_t(y|x)|, \mathbb{E}_{x\sim p_t}|p_s(y|x) - p_t(y|x)|]$.

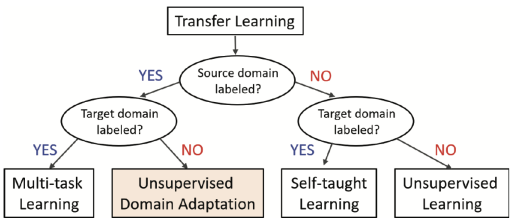


Figure 2.1: A taxonomy of transfer learning approaches based on the availability of labeled data in a source or target domain [2]

Here, $\mathcal{L}_t(h)$ and $\mathcal{L}_s(h)$ indicate the expected loss in the target and source domain, respectively. The second term on the right hand of the theorem, $d[.]$, shows the divergence measure between source and target distributions, e.g. the Jensen–Shannon (JS) divergence in the case of conventional adversarial UDA [86]. Finally, the third term on the right hand, $min[\mathbb{E}_{x\sim p_s}|p_s(y|x) - p_t(y|x)|, \mathbb{E}_{x\sim p_t}|p_s(y|x) - p_t(y|x)|]$, is negligible. Therefore, the first and second terms in this theorem can be considered as the upper bound of the expected loss in the target domain. To lower the generalization error on the target domain, UDA methods minimize the upper bound by minimizing the divergence or distribution shift between two domains.

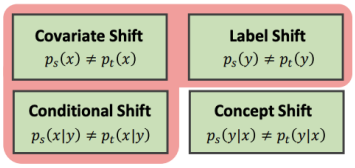


Figure 2.2: A summary of the possible shifts. [87]

The domain shift can be divided into four categories [85], as shown in Figure 2.2. In the presence of the covariate shift i.e. $p(x)$, the objective is to align the marginal distributions of the source and target domains. The more realistic shift is the conditional shift, which aims to align the shift of $p(x|y)$, because each class may have its own shift protocol. Moreover, when the proportion of classes differs between source and target domains, the label shift (i.e. the target shift) occurs. Finally, the concept shift [85], happens when classifying caret as vegetable or fruit in different countries. The concept shift is not a common problem in the segmentation tasks [2]. Most studies focus on a single shift only while they assume that other shifts remain invariant between source and target domains. The solutions to the UDA problem can be categorized into self-training [88, 89], and feature-level adversarial learning [90].

2.4.2 Adversarial Training

Various techniques address the distribution disparity between the source and target domains, targeting The pixel level, feature level, and output level through adversarial learning. Pixel-level alignment is often referred to as image-to-image translation or style transfer. Works along this line attempt to translate source images to the target domain, or vice versa, such as classic CycleGAN [91]. Recent studies have expanded the pixel-level alignment approach by incorporating feature-level alignment to enhance segmentation precision. Chen *et al.* (2019) [92] introduce feature alignment, which optimizes the image-to-image translation by aligning intermediate features. Additionally, Li *et al.* (2019) [39] incorporate the segmentation model to supervise the style transformation within the framework of cycle-GAN, ensuring the preservation of semantics that align with the segmentation task's objectives. Conversely, Hoffman *et al.* (2016) [93] propose to conduct alignment in the feature level. Tsai *et al.* (2018) [94] discover that aligning the distribution of the output features yields greater effectiveness compared to aligning the distribution of the intermediate feature space. Nonetheless, the direct alignment of feature distributions in a high-dimensional space presents challenges. Sankaranarayanan *et al.* (2018) [95] tackle this by reducing feature dimensions that contain the essential feature components. Subsequently, they are mapped back to the original feature space. Long *et al.* (2018) [96] propose that the global distribution alignment can compromise the distinctiveness of features within the target domain. To address this issue, Wang *et al.* (2020) [97] incorporate the class information into the adversarial loss. However, adversarial training often encounters issues with stability because of the lack of a comprehensive understanding of each category. As a result, some studies opt for the utilization of category anchors [98] derived from source data to enhance the alignment process. However, choosing these category anchors is challenging. Furthermore, constructing the global category anchors based on the unimodal distribution assumption for each class will disregard the within-class variation.

2.4.3 Self-training

In self-training approaches, pseudo labels are assigned to samples from the target domain to facilitate iterative training. The central concern in these techniques is how to achieve stable model training in the presence of noisy pseudo labels. Some studies proposed a variety of strategies such as dynamic

threshold strategy [43], or uncertainty estimation strategy [99], to select high-quality pseudo labels. In some other studies design curriculum learning [100], or anti-noise learning strategies [101] have been proposed to achieve model stability during training. Recently, Zhang *et al.* (2021) [45] use the relative feature distance to the prototypes to refine further target pseudo labels. Several studies have integrated self-training and adversarial training [102, 103], aimed at entropy minimization [104], boundary refinement [105], or curriculum-based approaches [106]. Most existing self-training approaches rely on training the classifier, i.e. discriminative model, using the cross-entropy loss function over the source ground truth and target pseudo labels. These discriminative classifiers suffer from various limitations: 1) neglecting to model the underlying data distribution, 2) unimodality for each class, and 3) these models suffer from accuracy degradation away from decision boundaries, hampering adaptation for critical tasks [49]. Furthermore, these UDA methods assume fully labeled source data and completely unlabeled target domains, which is often not the case in practice due to partial or noisy labels in both source and target domains.

Chapter 3

Universal Representation Learning for Multivariate Time Series using the instance-level and cluster-level Supervised Contrastive Learning

3.1 Introduction

In this dissertation, we introduce the *Supervised Contrastive Learning* for *Time Series Classification* (SupCon-TSC) model. This model is designed to empower deep learning models in dealing with limited labels in TSC tasks while acquiring low-dimensional feature representations. It builds upon Supervised Contrastive Learning (SupCon) principles and yields interpretable outcomes.

The recent success of the SupCon learning approach in various computer vision tasks inspired us to adapt this competitive approach for the TSC tasks. The SupCon loss function overcomes the shortcomings of the cross-entropy loss function, such as a lack of robustness to noisy labels [107, 108] and the potential for decision boundaries with poor margins resulting in poor classification performance. Leveraging the SupCon learning approach alleviates the challenge of defining classification boundaries

between classes. It achieves this by bringing the representations of instances with the same label closer together while moving them farther from those with different labels. In addition, because the SupCon loss function is a distance-based loss, it effectively addresses the issue of limited data in time series tasks. However, despite the advantages of the SupCon loss function, the intra-class variances and inter-class similarities found in many real-world time series make it challenging to learn universal low-dimensional feature representations using SupCon loss. To address this issue, we extend the SupCon learning approach to learn the low-dimensional universal representation, not only by applying the SupCon loss between time series instances but also between the clusters of instances across batches, as depicted in Figure 3.2. In this approach, we cluster the time series instances based on their labels within each batch. Subsequently, we apply the SupCon learning approach between each instance and centers of generated clusters across batches. This introduces cluster-level SupCon as a complement to an instance-level contrastive strategy. We introduce a cluster memory bank that allows us to access representations of clusters generated in previous batches during training. This approach helps in bringing clusters with the same label closer and distancing those with different labels. This process results in clearer boundary decisions by reducing intra-class variances and inter-class similarities. Unlike existing contrastive loss function studies, our approach does not depend on designing complex augmentation methods, which are challenging for time series data. The temporal dependencies in time series data present challenges in designing augmentation methods. This complexity is amplified when dealing with the MTSC task, as it requires considering the cross-correlations between variables across time. The major contributions of this study are summarized as follows:

1. We developed SupCon-TSC for time series data to capture contextual information, which provides interpretable outputs.

2. Even though the contrastive objective is usually based on augmented context views to get good results, our approach does not depend on adopting well-known augmentation methods. In other words, the developed approach is capable of learning the universal low-dimensional feature representations without introducing undetected inductive bias created by adopting well-known augmentation methods such as transformation- and cropping-invariance.

3. We evaluate the performance of the SupCon-TSC model on two small CPET datasets to demon-

strate the model's capability for learning better discriminative features than existing models.

4. We conduct extensive experiments on multivariate time series data to show the effectiveness of our method compared to standard approaches in the literature. Our new approach outperforms existing SOTAs on 29 UEA Archive datasets.

5. We design a SupCon loss at the cluster level in addition to the instance level to alleviate the negative impact induced by intra-class variances and inter-class similarities during training.

3.2 Methodology

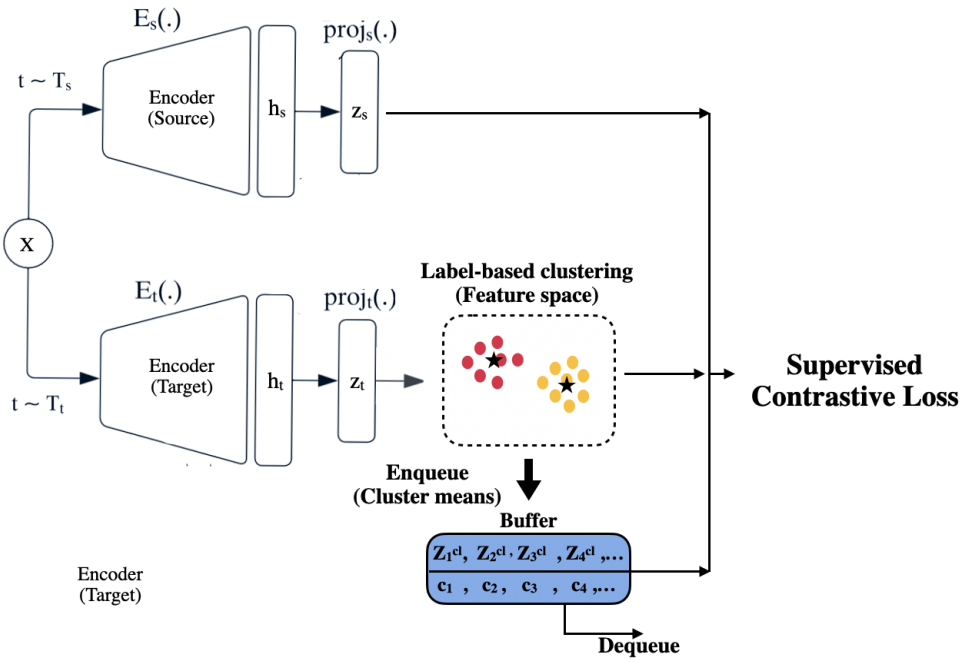
In this section, we first provide a brief introduction to the problem formulation in Section 3.1. Following that, we elaborate on the details of the developed method and our framework in Section 3.2.

3.2.1 Problem Formulation

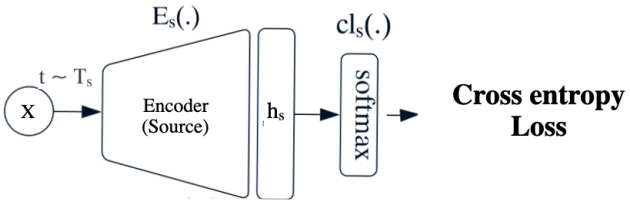
In multivariate time series classification, a data set consists of pairs $(\mathcal{X}, \mathbf{y})$, where $\mathcal{X} = \{\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}, ..., \mathbf{X_n}\} \in R^{n \times m \times l}$ contains n multi-dimensional time series observations and $\mathbf{y} \in R^n$ contains corresponding discrete class variables with c possible values for each observation. Here, each time series observation can be represented as a matrix with the dimension m and time series length l. The goal of the MTSC tasks is to train a classifier on the observed pairs of $(\mathcal{X}, \mathbf{y})$, enabling it to predict the class label of a new, unlabeled time series observation.

3.2.2 Model

In this section, we introduce our novel approach, i.e., SupCon-TSC, which aims to enhance model performance for downstream tasks like classification by learning a universal representation for multivariate time series data. Our approach consists of two stages: a) Learning the universal representation, and b) Training the classifier, as depicted in Figure 3.1. The first stage of SupCon-TSC is built upon the SupCon framework [109], initially designed for image representation learning. However, we have made modifications to adapt it to learning a universal representation of multivariate time series data



a) Learning the Universal representation



b) Training the classifier

Figure 3.1: Diagram of training process

for supervised MTSC. Algorithm 1 outlines the pseudo-code for this first stage. Specifically, the provided pseudo-code outlines an algorithm for learning a universal representation for multivariate time series data using instance-level and cluster-level supervised contrastive learning. The algorithm begins by initializing hyperparameters, encoder, and projection head weights, and creating an empty buffer. During the training process, as the algorithm progresses through a fixed number of epochs (N_e), a check is performed to determine whether the current epoch falls within the warm-up period (N_w). If the current epoch is within the warm-up period, the variable α is set to 0, implying that the cluster-level contrastive learning step is skipped. However, if the current epoch is equal to or greater than the number of warm-up epochs, α is set to 1, indicating that the cluster-level contrastive learning step will be executed as part of the algorithm for that epoch. The algorithm then iterates over sampled mini-batches. For each instance in the mini-batch, the algorithm applies augmentation techniques to generate weak (x_k^w) and strong (x_k^s)views of the given input sequence. Then, the encoder (E) processes these augmented sequences, and the projection head ($proj$) projects their hidden representations into lower-dimensional feature vectors. The algorithm performs clustering on the instances in the mini-batch based on their labels according to lines 16 to 18. Each instance is assigned to the cluster with the same label. Then, for each unique label, the algorithm calculates the average feature vector of instances (z_i^{cl}) with the associated label (c_k) and adds it to the buffer along with the corresponding label. The algorithm then proceeds to compute the instance-level and cluster-level contrastive losses. More details on Learning the Universal Representation, instance-level, and cluster-level contrastive learning approaches have been provided in the following sections.

The second stage of SupCon-TSC contains training the multilayer perceptron (MLP) classifier on top of the frozen representations using a cross-entropy loss.

Algorithm 1: Instance-level and cluster-level SupCon algorithm

Input: Input multi-dimension time series instances (X), Labels (Y)

Parameter: Buffer size (β), Batch size (N), Number of epochs (N_e), Number of warm-up epochs (N_w) Number of unique labels (N_l), Temperature (τ),

Initialize the weights of encoder (f) and projection head (g), Initialize buffer (B).

for *epoch:=1:*N_e **do**

  if $epoch < N_w$ **then**

    $\alpha = 0$

  else

    $\alpha = 1$

  for *sampled minibatch* **do**

    for $k \in 1, ..., N$ **do**

      $x_k^s = T_s(x_k)$

      $x_k^t = T_t(x_k)$

      $h_k^s = E(x_k^s)$

      $h_k^t = E(x_k^t)$

      $z_k^s = proj(h_k^s)$

      $z_k^t = proj(h_k^t)$

      Cluster instances in the batch

      Assign each time series instance (x_k) to the cluster with the same label (c_k)

    for $i \in 1, ..., N_l$ **do**

      $z_i^{cl} = \frac{\sum_{k=1}^{N} I\{c_k=i\} z_k^t}{\sum_{k=1}^{N} I\{c_k=i\}}$

      Update the Buffer B by adding μ_i and corresponding label c_k to it

    for $k \in 1, ..., N$ **do**

      Instance-level SupCon

      $A(k) = \{1, ..., N\}$

      $P(k) = \{p \in A(k) : y_k = y_p\}$

      $L_k^{Ins-level} = \frac{-1}{|P(k)|} \sum_{p \in P(k)} log \frac{exp(z_k^s \cdot z_p^t / \tau)}{\sum_{a \in A(i)} exp(z_k^s \cdot z_a^t / \tau)}$

      Cluster-level SupCon

      $A_{buf}(k) = \{1, ..., \beta\}$

      $P_{buf}(k) = \{p \in A_{buf}(k) : y_k = y_p\}$

      $L_k^{clus-level} = \frac{-1}{|P_{buf}(k)|} \sum_{p \in P_{buf}(k)} log \frac{exp(z_k^s \cdot z_p^{clus} / \tau)}{\sum_{a \in A_{buf}(i)} exp(z_k^s \cdot z_a^{clus} / \tau)}$

    $L = \sum_{k=1}^{N} L_k^{Ins-level} + \alpha L_k^{cl-level}$

3.2.2.1 Learning the Universal Representation

This stage serves as the pre-training phase for training the encoder to generate the universal representation. As depicted in Figure 3.1-a, the Siamese network consists of source (E_s) and target encoders (E_t), which take two augmented versions of a multivariate time series instance sampled from two distinct augmentation families.

$$x^s \sim T_s(x)$$

$$x^t \sim T_t(x)$$

where, x^s, and x^t represent the strongly and weakly augmented view of x, respectively. The high-variance strong augmentation (T_s) and low-variance weak augmentation (T_t) families are used to generate these strongly and weakly augmented views of x for the source and target networks, respectively. Wang *et al.* (2022) [110] demonstrated that these settings enhance the model performance on downstream tasks such as classification. Noted, even though an essential part of the success of the contrastive learning methods is designing and utilizing good data augmentation methods [111], our approach does not depend on the well-known augmentation methods. We utilize only jittering augmentation with low variance (weak augmentation) for the target network and high variance (strong augmentation) for the source network. After generating the augmented views of a given instance (x), they are passed to the encoder to learn the universal low dimensional representations ($h = E(x)$). To train the encoder, first, the encoder output will be sent to the MLP projection head to obtain the normalized embedding ($z = proj(E(x))$. In each iteration, the buffer is updated with the output from the target network. For every iteration, the target outputs of the given batch are clustered according to their labels, and the buffer is updated with the mean value of the clusters. Subsequently, the SupCon loss is calculated between the output of the source network, the output of the target network, and the buffer. This process aims to learn a discriminative representation that effectively characterizes instance x. The SupCon loss function enforces the normalized embeddings from the same class to pull closer together than embeddings from different classes. For this purpose, it tries to maximize the dot product between the given anchor and positive samples (i.e., samples with the same labels) while minimizing the dot product with negative samples (i.e., samples with different labels) within the batch. The SupCon learning is conducted at the instance and cluster level, which are explained in the following sections in detail.

3.2.2.2 Supervised Contrastive learning at the instance-level:

As depicted in Algorithm 1, within a batch of N samples, two encoding representations are generated for each instance: the source encoding representation (z^s) and the target encoding representation (z^t). We expect the source encoding to have higher variance in comparison with the target encoding representation as we use higher variance in the corresponding augmentation method.

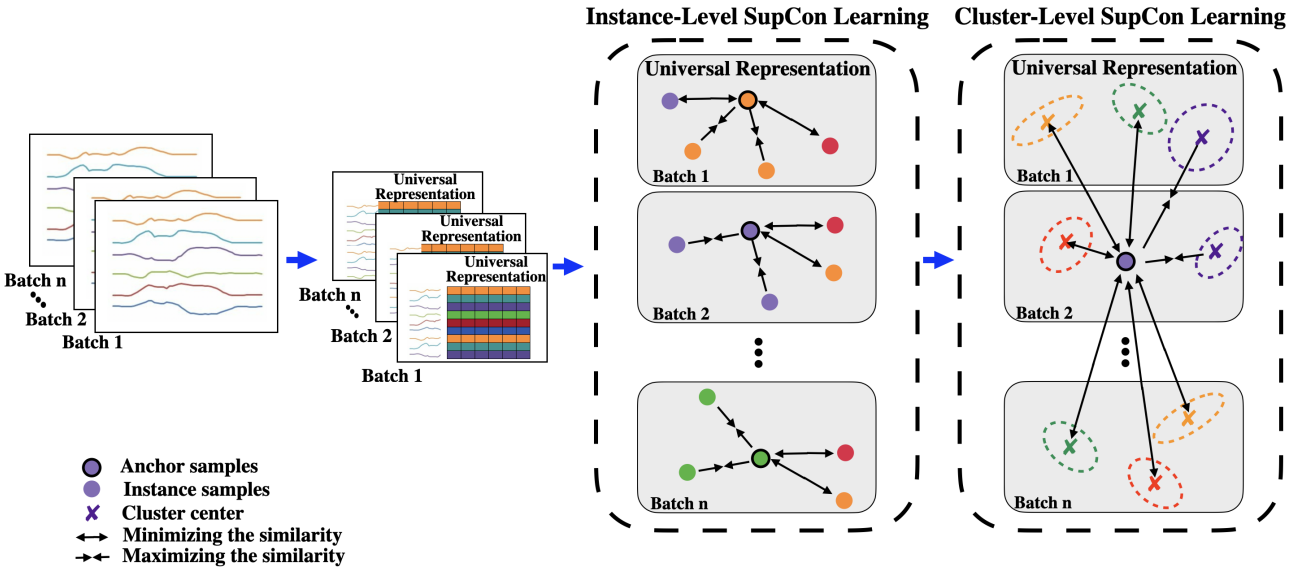


Figure 3.2: Diagram of SUpCon-TSC

The instance-level Supervised contrastive loss is as follows:

$$L^{SupCon} = \frac{-1}{\mid P(k) \mid} \sum_{p \in P(k)} log \frac{exp(z_k^s \cdot z_p^t / \tau)}{\sum_{a \in A(i)} exp(z_k^s \cdot z_a^t / \tau)} \tag{3.1}$$

where, τ is the temperature. For an anchor embedding z_k^s that comes from the source network, we denote z_p^t as a positive sample which is the output of the target network corresponding to the sample in the batch with the same label as the anchor image. Hence, (z_k^s, z_p^t) is a positive pair and the number of positive pairs for the anchor k is equal to the number of instances with the same label as the anchor instance in the batch. $A(i)$ is a set of all indexes in the given batch, while $P(k)$ indicates a set of positive samples for the anchor k. $P(k)$ contains indexes of those samples in the batch which have the same label as the anchor k.

Noted, the size of negative samples for the anchor k is $N(k) =\mid A(i) \mid - \mid P(k) \mid$. Figure 3.2 presents the Instance-level supervised contrastive learning between a given anchor and positive and negative samples in each batch.

3.2.2.3 Supervised Contrastive learning at the cluster-level among batches

In this approach, we design a cluster memory bank that contains the representation of the cluster's center generated in the previous batches during training. In each batch with N samples, we perform clustering over the target embeddings based on their labels. We assign the target embedding of each time series sample x_k to the cluster with the same label (c_k). Then, we determine the cluster centers using Equation 3.2. The representations of the cluster centers generated in each batch will be stored in the cluster memory bank. The cluster memory bank is built with size $N_{buffer} \times N_l \times D$, where N_{buffer}, N_l, and D are the memory size, number of unique classes for time series data set and the dimension of representation embedding, respectively.

$$z_i^{cl} = \frac{\sum_{k=1}^{N} I\{c_k = i\} z_k^t}{\sum_{k=1}^{N} I\{c_k = i\}} \tag{3.2}$$

As shown in Algorithm 1, the cluster-level SupCon learning is conducted using Equation 3.3 among the batches during training in addition to the instance-level SupCon learning in each batch.

$$L_k^{clus-level} = \frac{-1}{\mid P_{buf}(k) \mid} \sum_{p \in P_{buf}(k)} log \frac{exp(z_k^s \cdot z_p^{clus}/\tau)}{\sum_{a \in A_{buf}(i)} exp(z_k^s \cdot z_a^{clus}/\tau)} \tag{3.3}$$

We aim to optimize the following objectives: 1) Maximize the similarity between each instance embedding in a batch z_k^s and positive samples z_p^{clus} retrieved from the cluster memory bank, 2) Minimizing the similarity between each instance embedding in a batch z_k^s and negative samples also sourced from the cluster memory bank. In Equation 3.3, $A_{buf}(i)$ denotes the set of all indexes within the cluster memory bank, while $p_{buf}(k)$ represents the set of positive samples which have the same label as the anchor k in the cluster memory bank. Figure 3.2 outlines the cluster-level SupCon learning approach, depicting the interaction between a given anchor instance and positive and negative samples (i.e. centers of the clusters with the same and different labels) extracted from the cluster memory bank. The overall piece-wise training loss can be defined as follows:

$$L = \sum_{k=1}^{N} L_k^{Ins-level} + \alpha L_k^{cl-level} \tag{3.4}$$

$$\alpha = \begin{cases} 0 & epoch \leq N_w \\ 1 & epoch > N_w \end{cases} \tag{3.5}$$

We only utilize the instance-level contrastive loss to train the model during the first epochs. After training the model for N_w epochs, we take into account the cluster-level loss in addition to the instance-level loss to train the model.

3.2.2.4 Training the classifier

Illustrated in Figure 3.1-b, the objective of the second stage is to train a classifier on top of the source encoder, utilizing cross-entropy loss for predicting class labels in MTSC tasks. During this step, we discard the projection head ($proj(.)$), and the classifier is incorporated into the preserved frozen universal representation. Subsequently, the classifier is trained using the cross-entropy loss function.

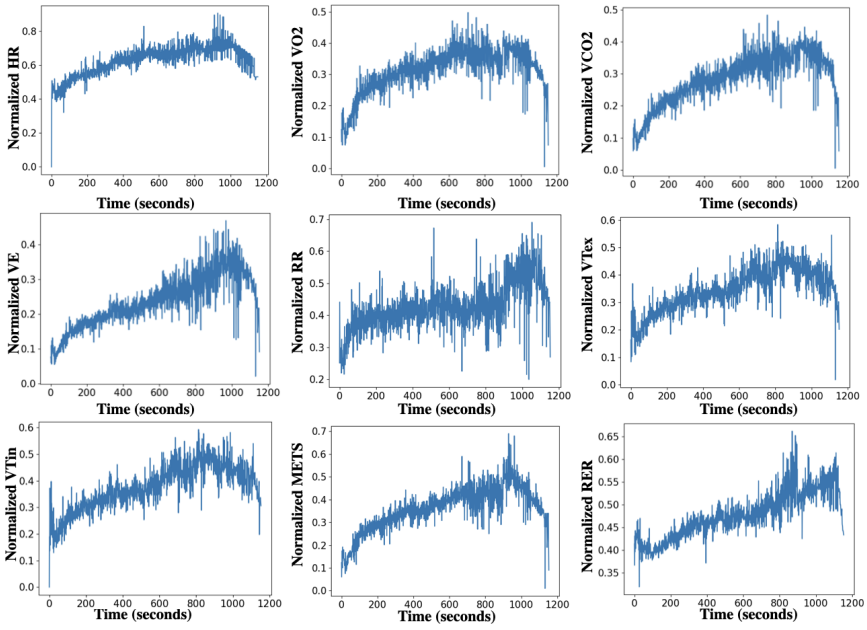


Figure 3.3: The aggregated second-by-second VE, RER, VTex, VTin, METS, RR, VCO2, VO2, for patients with label HF

3.3 Experiments

In this section, we assess the performance of SupCon-TSC on three different datasets: the UEA Multivariate Time Series Archive dataset and two Cardiopulmonary Exercise Testing datasets. Firstly, we provide detailed descriptions of the datasets, metrics used for evaluation, and the implementation specifics. Subsequently, we present a comprehensive analysis of experimental results, comparing the performance across diverse datasets. Finally, we delve into the ablation studies section, conducting in-depth analyses to further understand the model's effectiveness.

3.3.1 Datasets

1. **UEA Multivariate time series archive** [1] [112]: The archive includes data sets collected from different applications such as Human Activity Recognition, Motion classification, and ECG/EEG signal classification. For variable-length datasets, we pad all series to the same length, setting NaNs for missing observations. When an observation is missing (NaN), the corresponding mask position is set to zero. Also, we noticed inconsistencies between the current ERing dataset available at the UEA Multivariate time series archive and the dataset used in the referenced papers [70, 10]. To ensure the integrity of our experiments, we removed the ERing dataset from our analysis.

2. **Cardiopulmonary exercise testing (CPET) dataset 1** [113]: The CPET dataset consists of the breath-by-breath readings of 30 patients with two clinically diagnosed conditions: Heart Failure (HF) and Metabolic Syndrome (MS) (15 patients each). The testing protocol for gathering data involved using a treadmill with three stages: rest, testing, and recovery. This dataset contains the following variables: Metabolic equivalent of task (METS)(1 MET = 3.5ml/kg/min); Heart Rate (HR); inspired Volumes of Oxygen (VO2); expired Volumes of Carbon Dioxide (VCO2); Ventilation (VE); Respiratory Rate (RR); expiratory tidal volume (VTex); and inspiratory tidal volume (VTin); Respiratory Exchange Ratio (RER); Speed of the treadmill; Elevation of the treadmill; binary outcome variable indicating the clinically diagnosed condition

[1] Datasets are available at http://timeseriesclassification.com

of the patient. The aggregated second-by-second values of normalized CPET variables (i.e. HR, RR, VO2, VE, VCO2, RER, VTin, VTex) for participants with label HF as an example is shown in Figure 3.3. In other words, we compute the mean of each CPET variable per second over all participants with the label HF.

3. **Cardiopulmonary exercise testing (CPET) dataset 2** [114]: This dataset comprises breath-by-breath readings from 78 healthy children and adolescents who underwent the (Multiple Brief Exercise Bouts) (MBEB) task at low, moderate, and high-intensity work rates. Even though all participants completed the ten bouts at low and moderate-tensity, half of them failed and stopped before all ten bouts had been completed (task failure) high-tensity work rate. This dataset the following variables: Heart Rate (HR); inspired Volumes of Oxygen (VO2); expired Volumes of Carbon Dioxide (VCO2); Respiratory Rate (RR); gender; maturational status; body mass; total fat; binary outcome variable indicating whether the participant completed the test. The aggregated second-by-second values of CPET variables (i.e. HR, RR, VO2, VCO2) over all participants are shown in Figure 3.4.

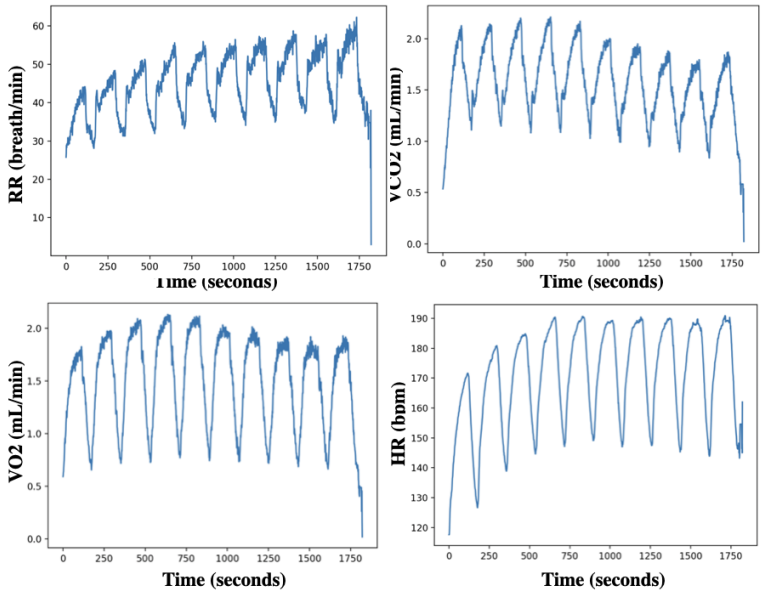


Figure 3.4: The aggregated second-by-second RR, VCO2, VO2, and HR over all participants from CPET dataset 2

3.3.2 Metric

Each model is evaluated using the accuracy score (i.e. $\frac{TP+TN}{TP+FP+TN+FN}$). where TP, FP, TN, and FN are True Positive, False Positive, True Negative, and False Negative, respectively.

3.3.3 Friedman test and Wilcoxon test

To find the differences between the methods, we leverage the Freidman test which is a non-parametric statistical test. Moreover, the Wilcoxon-signed rank test is used to compare pairs of classifiers over the datasets. The Friedman test and Wilcoxon-signed rank test with Holm's $\alpha(5\%)$ are conducted by following the process described in [115].

3.3.4 Interpretability

Gradient-weighted Class Activation Mapping (Grad-CAM) [116] is one of the well-known methods for generating saliency maps to support convolutional neural network predictions. The Grad-CAM aims to identify the regions of the input data that the most influence the predictions using the class-specific gradient information. In this study, we use the Grad-CAM approach to identify those time steps of the time series that influence the most on the model's decision for a specifically assigned label. The following paragraph explains how we adapt Grad-CAM for the SupCon-TSC model.

In order to build the attribution map, we apply grad-CAM to the output features of the last 1D convolution layer. First, we compute the importance of each feature map (w_k^c) by obtaining the gradient of the output score for specific class c (y_c) with respect to each feature map activation A^k as:

$$w_k^c = \frac{1}{Z} \sum_i \frac{\sigma y_c}{\sigma A_i^k} \tag{3.6}$$

where Z is the total number of units in A. Then, w_k^c is used to compute a weight combination of feature maps for class c by Equation 3.7. The ReLU non-linearity is used to keep only positive values.

$$L_{1D}^c = ReLU(\sum_k w_k^c A^k) \tag{3.7}$$

3.3.5 Architecture Details

The model architecture is as follows:

1. Encoder: ResNet [15]
2. Head: two linear layers with ReLu activation function.
3. Classifier: two linear layers with ReLu activation function and Softmax on top.

3.3.6 Hyperparameters

The grid search along with the 5-fold cross-validation on the training set is used to set hyperparameters for each dataset. Please refer to Table 3.1 for the hyperparameters used in our experiments.

3.3.7 Models

We have compared the performance of our method with the following state-of-the-art MTSC models on the UEA Multivariate time series archive datasets.

- **TapNet:** Multivariate Time Series Classification with Attentional Prototypical Network was applied to time series data [10].
- **WEASEL+MUSE (WM):** Word ExtrAction for time Series cLassification plus Multivariate Unsupervised Symbols and dErivatives was applied to time series data [68].
- **MLSTM-FCN (MF):** Multivariate LSTM Fully Convolutional Networks for Time Series Classification was applied to time series data [71].
- **MTEX-CNN (MC):** Multivariate Time Series EXplanations for Predictions with Convolutional Neural Networks was applied to time series data [72].
- **CMFM+RF (CMRF):** Random Forest (RF) was applied to the set of time series features obtained by Complexity Measures and Features for Multivariate Time Series (CMFMTS) approach [117].

Table 3.1: Selected hyperparameters. Abbreviations: LR_1- Learning rate 1, LR_2- Learning rate 1, BS_1- Batch size 1, BS_2- Batch size 2

Datasets	LR_1	BS_1	$Epoch1$	LR_2	BS_2	$Epoch2$
ArticularyWordRecognition	0.001	40	100	0.005	20	100
AtrialFibrillation	0.001	15	100	1e-05	15	100
BasicMotions	0.001	10	100	0.001	5	100
CharacterTrajectories	0.001	50	100	0.001	50	100
Cricket	0.001	10	100	0.001	50	100
DuckDuckGeese	0.001	30	100	0.001	5	150
EigenWorms	0.001	10	100	0.001	10	150
Epilepsy	0.001	10	100	0.001	50	150
EthanolConcentration	0.001	10	100	0.001	20	150
FaceDetection	0.001	50	100	0.001	70	100
HandMovementDirection	0.001	50	100	0.0001	5	100
FingerMovements	0.005	100	100	0.0005	100	150
Handwriting	0.001	30	100	0.001	5	150
Heartbeat	0.001	50	100	0.001	10	100
InsectWingbeat	0.001	1000	100	0.0001	1000	100
JapaneseVowels	0.001	20	100	0.001	5	100
Libras	0.0001	30	100	0.001	5	150
LSST	0.001	20	100	0.001	5	100
MotorImagery	0.001	70	100	0.001	10	100
NATOPS	0.005	25	100	0.005	10	100
PenDigits	0.001	100	100	0.001	50	100
PEMS-SF	0.001	70	100	0.001	5	100
Phoneme	0.001	50	100	0.001	200	100
RacketSports	0.001	30	100	1e-05	5	150
SelfRegulationSCP1	0.001	20	100	1e-05	100	100
SelfRegulationSCP2	0.001	20	100	0.0001	5	100
SpokenArabicDigits	0.001	20	100	0.001	10	100
StandWalkJump	0.001	3	100	0.001	9	100
UWaveGestureLibrary	0.001	15	100	0.001	10	150

- **CMFM+SVM (CMSVM):** Support Vector Machine (SVM) was applied to the set of time series features obtained by CMFMTS approach [117].
- **CMFM+ C5.0B (CMC5.0B):** C5.0 with boosting (C5.0B) was applied to the set of time series features obtained by CMFMTS approach [117].
- **CMFM+1NN (CM1NN):** 1-Nearest Neighbor classifier with Euclidean distance (1NN-ED) was applied to the set of time series features obtained by CMFMTS approach [117].
- **XCM:** The eXplainable Convolutional neural network model was applied to time series data [70].
- **LCEM:** Local Cascade Ensemble for Multivariate data classification (LCEM) was applied to time series data [118].
- **XGBM:** The Extreme Gradient Boosting algorithm was applied to the LCEM transformation [118].
- **RFM:** Random Forest for Multivariate (RFM) algorithm was applied to the LCEM transformation [118].
- DW_I / $DW_I(n)$**:** a 1-Nearest Neighbor classifier was applied to the sum of DTW distances for each dimension with and without normalization (n) [62].
- DW_D / $DW_D(n)$**:** Dimension-dependent dynamic time warping [62] was employed with and without normalization (n). Distances are computed using multidimensional points, and subsequently, a 1-Nearest Neighbor classifier was applied to them.

3.3.8 Classification Performance Evaluation

We evaluate the performance of the SupCon-TSC model on two small CPET datasets and the UEA Multivariate time series archive.

3.3.8.1 CPET datasets

Table 3.2 shows the performance of the SupCon-TSC alongside the state-of-the-art deep learning models on small CPET datasets 1 and 2. To maintain consistency with prior research [113, 114], we conducted experimentation through the same k-fold cross-validation method. Additionally, for our experiment, we focused exclusively on the initial four bouts from the second dataset. We then proceeded to smooth and align these bouts as recommended in [114]. Four bouts of CPET variables after converting the discrete time series to 78 smoothed and aligned curves are shown in Figure 3.5. As shown, the SupCon-TSC model has achieved better accuracy on both datasets.

Table 3.2: The model's performance on the second CPET datasets 1 and 2

Dataset	**Model**	**k-fold CV**	**Accuracy (%)**
CPET 1	CNN [113]	5-fold	90
	SupCon-TSC	5-fold	**97**
CPET 2	GADF + Attention [114]	10-fold	80.8
	SupCon-TSC	10-fold	**86.07**

To investigate the interpretability of the model, we present a comprehensive analysis of the attention mechanism of our SupCon-TSC model when applied to CPET dataset 2. The dataset consists of samples with binary labels indicating whether the participant completed the test. We sought to understand how the model's attention is distributed across the input data during the prediction process. Figure 3.6 shows the network's attention for two samples with different labels from CPET dataset 2. The attention maps provide valuable insights into the regions of interest that the model deems crucial for making predictions. As shown, the network's attention is spread approximately across time steps 150-190, 310-380, 510-540, and 690-710, which are associated with the valleys in the graphs (i.e., displayed by red circles on the first HR graph). Remarkably, these identified intervals align remarkably well with the recovery points observed in the heart rate (HR) and gas exchange change graphs. From a physiological standpoint, these recovery points have significant implications as they are widely recognized indicators of an individual's fitness level [119, 120]. Notably, we found that the identified recovery points align with the findings from studies [114, 121]. These studies suggest that incomplete

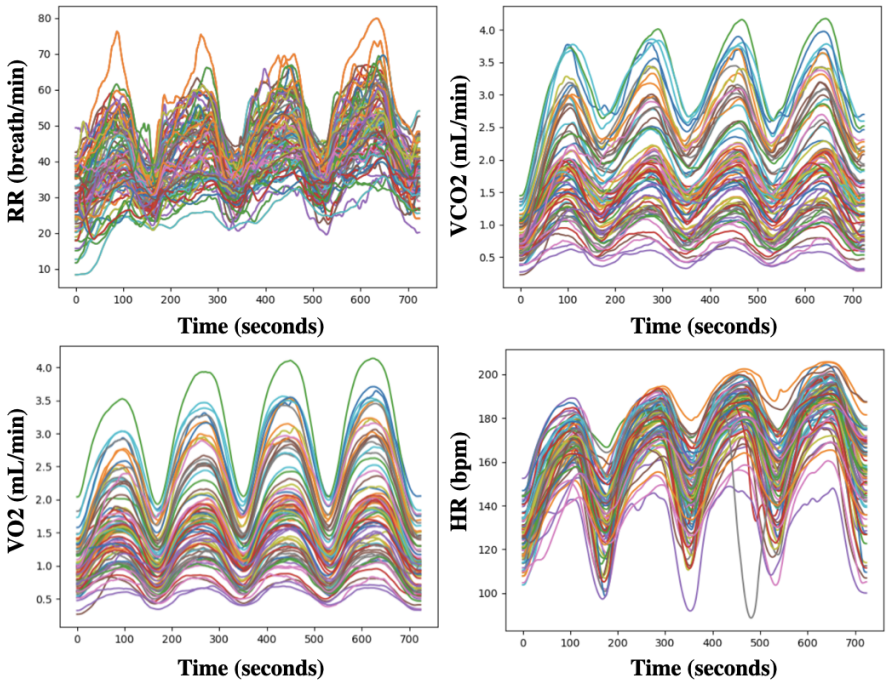


Figure 3.5: Four bouts of CPET variables after smoothing and aligning the curves

recovery from individual exercise bouts may result in a cumulative response deficiency. This deficiency, over time, could potentially manifest in physiological signals that can impact cognitive exercise behavior, which aligns with the patterns identified by the SupCon-TSC model.

3.3.8.2 UEA Multivariate time series archive

The accuracy results of SupCon-TSC and the other state-of-the-art algorithms on the public UEA test sets are presented in Table 3.3. In the SupCon-TSC approach, ensemble learning is used to make the final prediction by taking the average over the five different models' outputs trained using 5-fold cross-validation. We perform the hyper-parameter tunning for XCM, TapNet, MTEX-CNN, and MLSTM-FCN models. The results of other baseline models are taken from the [70, 117]. The dash shows that the approach ran out of memory. Also, the best accuracy for each dataset is boldfaced. The SupCon-TSC was implemented in Python3 using Pytorch 1.10 and all the experiments are conducted on a single Tesla k80 GPU with 11GB memory. As Table 3.3 indicates, SupCon-TSC achieves better

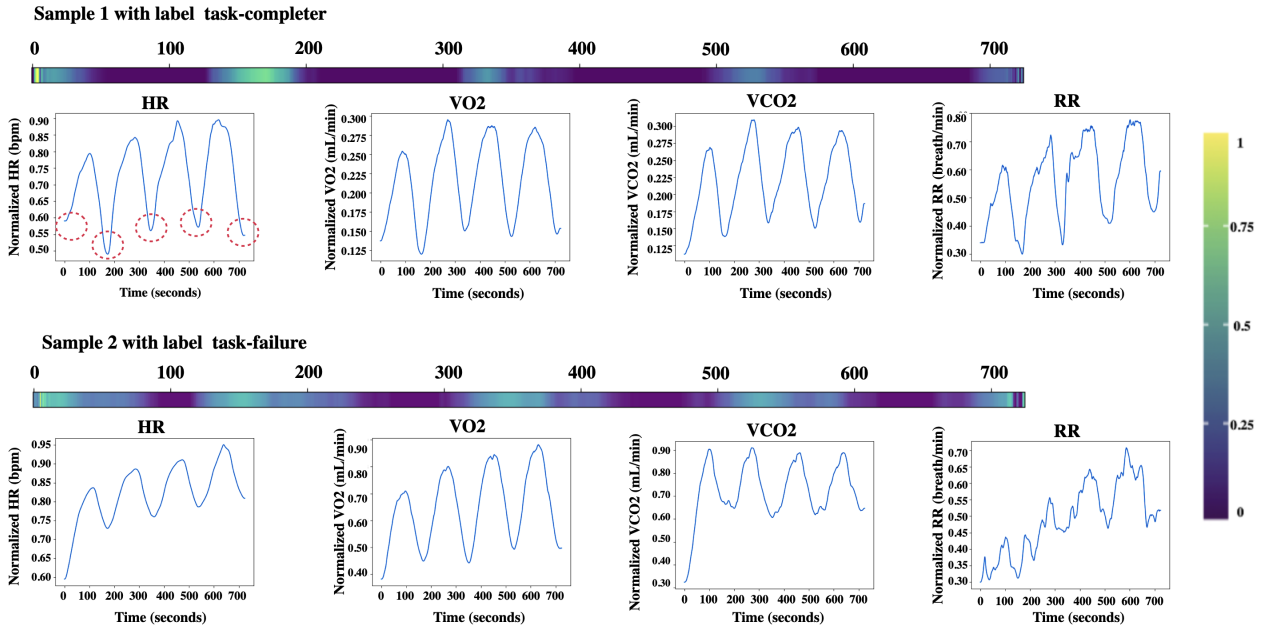


Figure 3.6: Time attention corresponding to a prediction for two participants with label task-failure and task completer

performance on 11 out of 29 UEA datasets in comparison with the baseline methods followed by LCEM with 7 datasets. The average rank is computed using a pairwise Wilcoxon signed rank test and we observe that the best average rank belongs to SupCon-TSC (5.07) which is followed by LCEM (5.26). Furthermore, Table 3.3 indicates that the SupCon-TSC approach outperforms LCEM methods in 18 out of 29 datasets.

We applied the Friedman test to investigate if there is a significant difference between the methods. The output of the Friedman test is $p = 4.205e - 19$, which is smaller than $\alpha = 0.05$, indicating that there is a significant difference among all ten methods. Figure 3.7 shows the accuracy scatter plots of SupCon-TSC against each of the LCEM and MLSTM-FCN. Figure 3.8 shows a critical difference diagram obtained by using the pairwise Wilcoxon signed-rank test. The numbers on each line are the average rank of the corresponding method and the solid bars indicate the groups of methods between which there are no significant differences in terms of accuracy. As shown in Figure 3.8, the SupCon-TSC model has the first rank followed by LCEM and MLSTM-FCN approaches.

Table 3.3: Accuracy results on the UEA Multivariate time series datasets. Abbreviations:ST-SupCon-TSC,WM-WEASEL+MUSE, MF-MLSTM-FCN, MC-MTEX-CNN, CMRF-CMFM+RF, CMSVM-CMFM+SVM, CM1NN-CMFM+1NN

Datasets	**ST**	**TapNet**	**MC**	**XCM**	**MF**	**WM**	**LCEM**	**XGBM**	**RFM**	**CMRF**	**CMSVM**	**CM1NN**	**CMC5.0B**	DW_I	DW_D	DW_I**(n)**	DW_D**(n)**
ArticularyWordRecognition (AW)	0.98	0.964	0.913	0.977	0.986	**0.993**	**0.993**	0.99	0.99	0.99	0.977	0.983	0.91	0.98	0.987	0.98	0.987
AtrialFibrillation (AF)	**0.467**	0.333	0.333	**0.467**	0.133	0.267	**0.467**	0.40	0.333	0.20	0.267	0.133	0.20	0.267	0.20	0.267	0.220
BasicMotions (BM)	**1**	**1**	0.68	**1**	**1**	**1**	**1**	**1**	**1**	0.975	0.925	0.95	0.85	**1**	0.975	**1**	0.975
CharacterTrajectories (CT)	**0.997**	**0.997**	0.974	0.995	0.993	0.990	0.979	0.983	0.985	0.970	0.970	0.933	0.942	0.969	0.990	0.969	0.989
Cricket (C)	**1**	0.958	0.78	0.986	0.986	0.986	0.986	0.972	0.986	0.972	0.958	0.972	0.861	0.986	**1**	0.986	**1**
DuckDuckGeese (DDG)	0.54	0.44	0.4	0.3	0.579	0.575	0.375	0.40	0.40	0.52	0.44	0.40	0.42	0.55	**0.60**	0.55	**0.60**
EigenWorms (EW)	0.885	0.86	0.419	0.526	**0.908**	0.89	0.527	0.55	**1**	0.817	0.84	0.794	0.817	0.603	0.618	-	0.618
Epilepsy (EP)	0.993	0.978	0.94	0.94	0.985	0.993	0.986	0.978	0.986	**1**	0.978	0.957	0.884	0.978	0.964	0.978	0.964
EthanolConcentration (EC)	0.231	0.231	0.251	0.32	0.254	0.316	0.372	0.422	**0.433**	0.335	0.327	0.304	0.35	0.304	0.323	0.304	0.323
FaceDetection (FD)	0.565	0.55	0.50	0.58	0.556	0.545	0.614	**0.629**	0.614	0.548	0.548	0.579	0.54	0.513	0.529	-	0.529
HandMovementDirection (HMD)	0.338	0.37	0.432	0.405	0.472	0.378	**0.649**	0.541	0.50	0.284	0.324	0.189	0.338	0.306	0.231	0.306	0.231
FingerMovements (FM)	**0.61**	0.52	**0.61**	0.59	0.579	0.54	0.59	0.53	0.56	0.52	0.46	0.53	0.44	0.52	0.53	0.52	0.53
Handwriting (HW)	0.566	0.37	0.17	0.4	0.544	0.531	0.287	0.267	0.267	0.282	0.184	0.249	0.165	0.509	**0.607**	0.316	0.286
Heartbeat (HB)	0.746	0.752	0.721	0.72	0.731	0.727	0.761	0.693	**0.80**	0.766	0.732	0.62	0.741	0.659	0.717	0.658	0.717
InsectWingbeat (IW)	0.667	0.208	0.105	0.105	0.105	-	0.228	0.237	0.224	**0.677**	0.10	0.266	-	-	0.115	-	-
JapaneseVowels (JV)	0.987	0.965	0.951	0.986	**0.992**	0.978	0.978	0.968	0.970	0.837	0.778	0.695	0.795	0.959	0.949	0.959	0.949
Libras (LIB)	0.85	0.877	0.6	0.77	0.883	**0.894**	0.772	0.767	0.783	0.867	0.833	0.828	0.839	0.894	0.872	0.894	0.870
LSST (LSST)	**0.657**	0.55	0.57	0.51	0.601	0.628	0.652	0.633	0.612	0.652	0.648	0.50	0.631	0.575	0.551	0.575	0.551
MotorImagery (MI)	0.59	0.53	0.5	0.5	0.529	0.50	**0.60**	0.46	0.55	0.51	0.50	0.44	0.49	0.39	0.50	-	0.50
NATOPS (NATO)	0.894	**0.93**	0.75	0.71	0.905	0.883	0.916	0.90	0.911	0.817	0.75	0.739	0.817	0.85	0.88	0.85	0.883
PenDigits (PD)	**0.993**	0.98	0.896	0.98	0.99	0.969	0.977	0.951	0.951	0.951	0.959	0.944	0.933	0.939	0.977	0.939	0.977
PEMS-SF (PEMS)	0.861	0.77	0.838	0.83	0.809	-	0.942	0.983	0.983	**1**	0.694	0.775	0.965	0.734	0.711	0.734	0.711
PhonemeSpectra (PS)	**0.322**	0.19	0.08	0.13	0.266	0.19	0.288	0.187	0.222	0.287	0.25	0.158	0.224	0.151	0.151	0.151	0.151
RacketSportsc(RS)	0.875	0.83	0.723	0.78	0.875	0.914	**0.941**	0.928	0.921	0.809	0.809	0.711	0.728	0.842	0.803	0.842	0.803
SelfRegulationSCP1 (SRS1)	0.73	0.75	0.767	**0.860**	0.829	0.744	0.839	0.829	0.826	0.812	0.792	0.703	0.812	0.765	0.775	0.765	0.775
SelfRegulationSCP2 (SRS2)	**0.55**	**0.55**	0.50	**0.55**	0.494	0.522	**0.55**	0.483	0.478	0.417	0.461	0.50	0.539	0.533	0.539	0.533	0.539
SpokenArabicDigits (SA)	**0.995**	0.983	0.986	**0.995**	0.994	0.982	0.973	0.970	0.968	0.976	0.979	0.915	0.933	0.960	0.963	0.959	0.963
StandWalkJump (SWJ)	**0.6**	0.47	0.4	0.533	**0.6**	0.333	0.40	0.333	0.467	0.333	0.20	0.133	0.257	0.333	0.20	0.333	0.20
UWaveGestureLibrary (UW)	0.812	0.89	0.69	0.88	0.881	0.903	0.897	0.894	**0.907**	0.772	0.738	0.753	0.641	0.869	0.903	0.868	0.903
Total best acc	11	4	1	5	4	3	7	2	5	3	0	0	0	1	3	1	2
Ours 1-to-1-Wins/ties	-	23	26	24	19	19	18	20	18	21	27	26	26	24	22	21	23
Avg. Rank	5.07	7.4	12.36	8.47	6.09	7.37	5.26	8.03	6.11	8.52	11.31	13.41	12.10	10.81	9.48	11.23	9.98

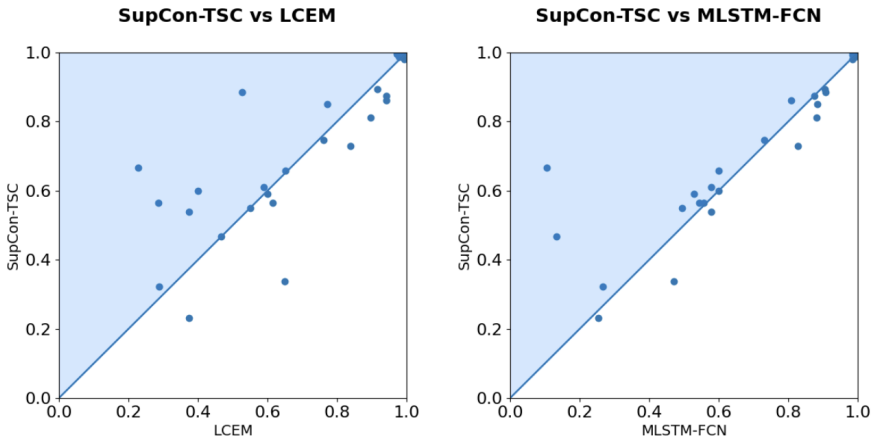


Figure 3.7: Scatter plots of accuracy on 29 UEA MTSC problems. *Left*: SupCon-TSC vs LCEM showing that SupCon-TSC beats LCEM on 18 problems. *Right*: SupCon-TSC vs MLSTM-FCN showing that SupCon-TSC beats MLSTM-FCN on 19 problems

3.3.9 Ablation studies

To study the effect of our developed Supervised Contrastive Learning method, we separately train ResNet models with and without designed Supervised Contrastive Learning. As shown in Table 3.4, the Supervised Contrastive Learning component improves the performance of the model in 22 out of 29 datasets which verifies the effectiveness of our approach.

3.4 Conclusion

This dissertation developed Supervised Contrastive learning for time series classification (SupCon-TSC). This model is based on the instance-level and cluster-level supervised contrastive learning approaches to learn the discriminative and universal representation for the multivariate time series dataset. As this approach is an end-to-end model, it allows us to detect those time steps of the time series that have the maximum influence on the model's prediction via utilizing the Grad-CAM method. The experimental results on small CPET datasets indicate the capability of our SupCon-TSC model to learn discriminative features where the labeled dataset is insufficient. Furthermore, the new

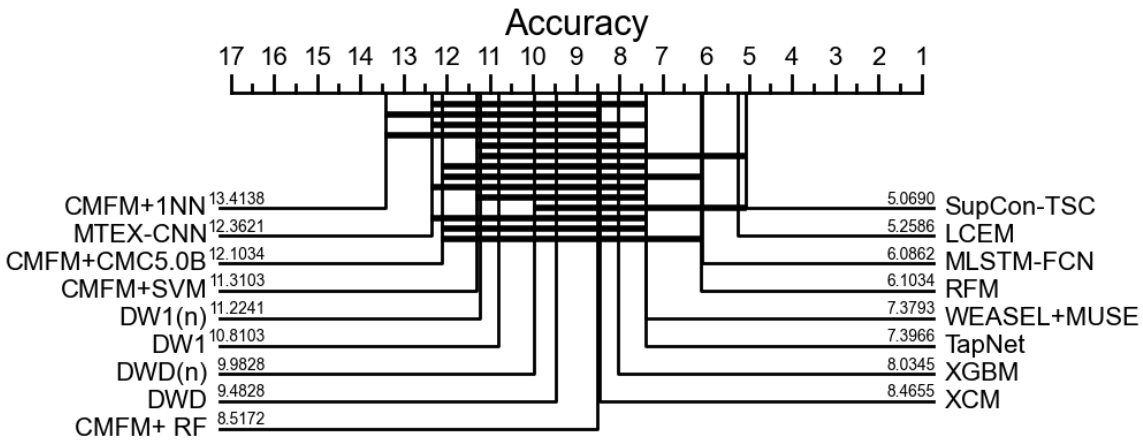


Figure 3.8: Critical difference diagram ($\alpha = 0.05$)

Table 3.4: Effect of our Supervised Contrastive Learning method

Datasets	**AW**	**AF**	**BM**	**CT**	**C**	**DDG**
w/o SupCon	0.97	0.266	**1.0**	0.995	0.986	0.44
w/SupCon	**0.98**	**0.467**	**1.0**	**0.997**	**1.0**	**0.54**
Datasets	**EW**	**EP**	**EC**	**FD**	**HMD**	**FM**
w/o SupCon	0.862	0.985	**0.277**	0.559	**0.378**	0.52
w/ SupCon	**0.885**	**0.993**	0.231	**0.565**	0.338	**0.61**
Datasets	**LIB**	**LSST**	**MI**	**NATO**	**PD**	**PEMS**
w/o SupCon	**0.872**	**0.662**	0.59	**0.911**	0.986	0.843
w/ SupCon	0.85	0.657	**0.59**	0.894	**0.993**	**0.861**
Datasets	**HW**	**HB**	**IW**	**JV**	**PS**	**SA**
w/o SupCon	**624**	0.741	0.665	0.983	0.313	0.993
w/ SupCon	0.566	**0.746**	**0.667**	**0.987**	**0.322**	**0.995**
Datasets	**RS**	**SRS1**	**SRS2**	**SWJ**	**UW**	
w/o SupCon	0.848	0.703	0.488	0.333	**0.837**	
w/ SupCon	**0.875**	**0.730**	**0.55**	**0.6**	0.812	

model outperforms the state-of-the-art models in 11 out of 29 UEA archive datasets. In our future work, we would like to focus on the augmentation methods and evaluate their impact on SupCon-TSC performance.

Chapter 4

Label-efficient Contrastive Learning-based model for nuclei detection and classification in 3D Cardiovascular Immunofluorescent Images

4.1 Introduction

This dissertation introduces the Label-efficient Contrastive learning-based (LECL) model for detecting and classifying various types of nuclei in 3D immunofluorescent images using weak annotations (point annotations). Addressing the challenge of training weakly-supervised learning models for 3D images with multiple channels (z-axis), our model avoids the limitations of Maximum Intensity Projection (MIP) by introducing a novel approach called Extended Maximum Intensity Projection (EMIP). The MIP approach is a common technique used to reduce the computational burden in image analysis. Several studies, such as Noguchi et al. (2023) [122] and Nagao et al. (2020) [123], have successfully

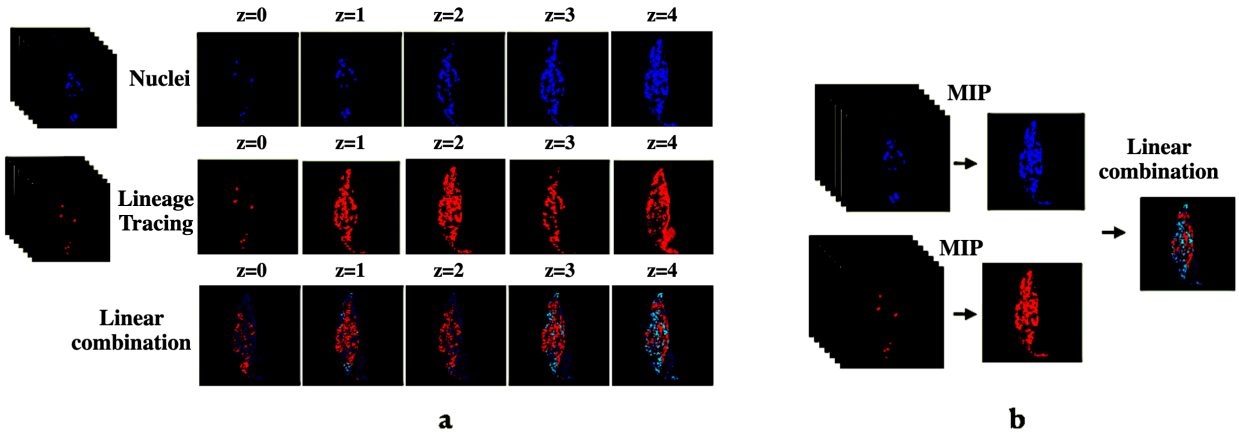


Figure 4.1: a) It presents the sequence of channels (i.e. z=0,..,n) for the nuclei (first row) and the Lineage Tracing marker (second row). The nuclei and Lineage Tracing marker channels are associated with each other in order. The third row indicates the linear combination of the nuclei and Lineage Tracing marker per slice.

employed the MIP technique for image preprocessing to convert 3D images into 2D format in tasks like cell segmentation and cell cycle phase classification. Nevertheless, it's important to consider that the MIP approach might not be the optimal option for nuclei detection and classification models, as elaborated in the subsequent section. The LECL model aims to detect nuclei and assign them a classification label based on specific markers (e.g., Lineage Tracing) with minimum labeling cost. To reduce the labeling cost, we request expert point annotations rather than fine pixel-wise annotations, which require much less effort. Das et al. [57] confirm this by stating that annotating 2975 images of the Cityscapes dataset with point labels costs 37.2 hours, while fine labeling costs 4463 hours. We assign the label "positive" if and only if the given detected nucleus overlaps with the given marker, otherwise, it is labeled as "negative". It is notable that training the model to classify the type of each nucleus using weak annotations in these images is a difficult task because these images contain multiple channels (Z-axis) for nuclei and different markers, as shown in Figure 4.1-a. The main challenges of nuclei detection/classification in 3D images are described in detail in section 4.2. To address these challenges, the EMIP approach partially performs the maximum intensity projection per nucleus where z levels contain the given nucleus to convert multi-channel images to 2D z-stack images. Additionally, to

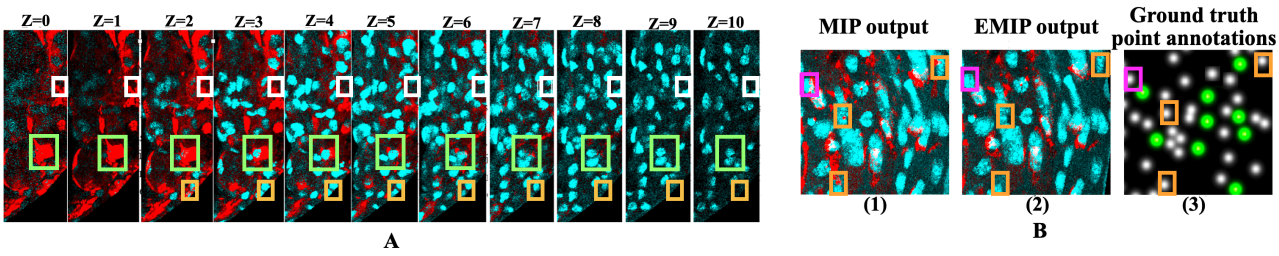


Figure 4.2: A) Challenges: **(1) nucleus in the yellow square:** Even though the ground truth labels for the nucleus in the yellow and green squares are positive, they are only coincident in the second slice, **(2) nucleus in the white square:** it shows an example of nonoverlapping marker and nucleus, (B-1) The output of the MIP approach, (B-2) The output of the EMIP approach, (B-3) Ground truth point annotations: green color is associated with the nuclei with label positive and white is associated with the nuclei with label negative

enhance performance in a weak setting, the model incorporates semi-supervised learning with entropy minimization loss over nuclei boundaries as unlabeled areas. Furthermore, a supervised contrastive learning method is implemented to improve the model's nuclei classification metrics.

The major contributions of this study are summarized as follows:

1. We develope an automated approach called LECL for 3D nuclei detection and classification in fluorescent images with minimum labeling cost.
2. We design the EMIP approach that addresses the limitations of MIP approach to convert multi-channel images to 2D z-stack images.
3. We show that the SCL loss enhances the model's performance by capturing global semantic relationships.

4.2 Challenges

The main challenges associated with detecting and classifying the types of nuclei in fluorescent images can be categorized into three groups as follows:

1. **One specific nucleus might spread over multiple z-slices, as shown in Figure 4.2-A, but only have a point annotation in one z-slice.** For example, the blue color nucleus in the orange square spreads over z-slices from two to eight, but the experts are asked to only annotate that nucleus in one of the slices to minimize the labeling cost. Therefore, only a subset of nuclei are annotated in each z-slice. Having incomplete nuclei annotations in each z-slice makes it challenging to train the model over each z-slice separately.

2. **The marker and nucleus might not be coincident in all z-slices.** In fluorescent images, the given nucleus is labeled as positive if that nucleus and the marker overlap at least in one of the z-slices. In other words, even though the ground truth label for the nucleus is positive, the nucleus might not contain the marker in some slices as shown in Figure 4.2-A.

3. **Maximum Intensity Projection (MIP) can cause objects to appear coincident that are actually separated in space.** Based on the literature, some approaches convert multi-channel 3D images to 2D z-stack images using MIP, as shown in Figure 4.1-b. This approach utilizes maximum intensity projection over nuclei/marker channels to convert these 3D multi-channel nuclei/marker images to 2D images (i.e., collapse images along with the z-axis). Then, the 2D nuclei image is combined with the 2D marker image using the linear combination method. However, this approach can be problematic when there are non-overlapping nucleus and marker in the same x, y, but at different z-axis. Figure 4.2-A illustrates this, where the blue nucleus and red marker in the white square indicate non-overlapping objects that could be falsely shown as overlapping using the MIP approach.

4.3 Method

In this section, we describe the our Label-efficient Contrastive learning-based (LECL) model (Figure 4.3), which consists of two components: a) Extended Maximum Intensity Projection (EMIP) and b) Supervised Contrastive Learning-based (SCL) training strategy.

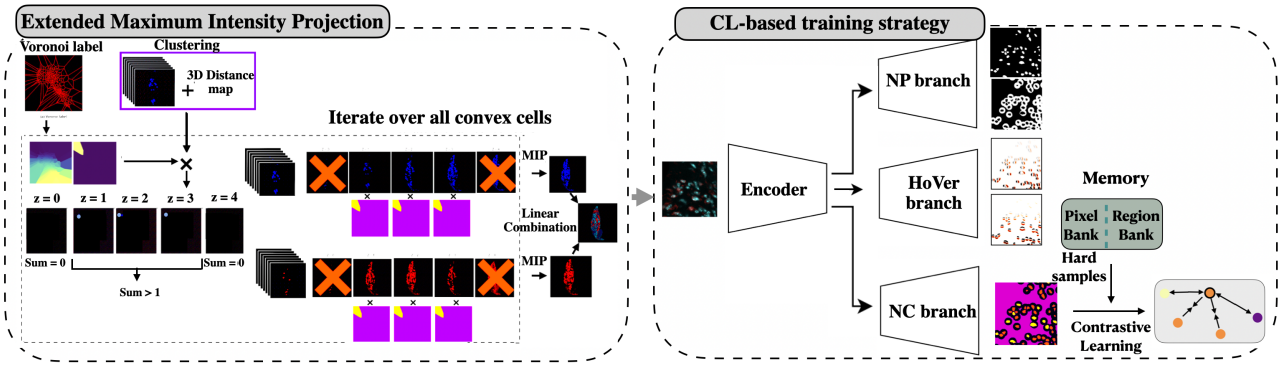


Figure 4.3: Schematic representation of the LECL model

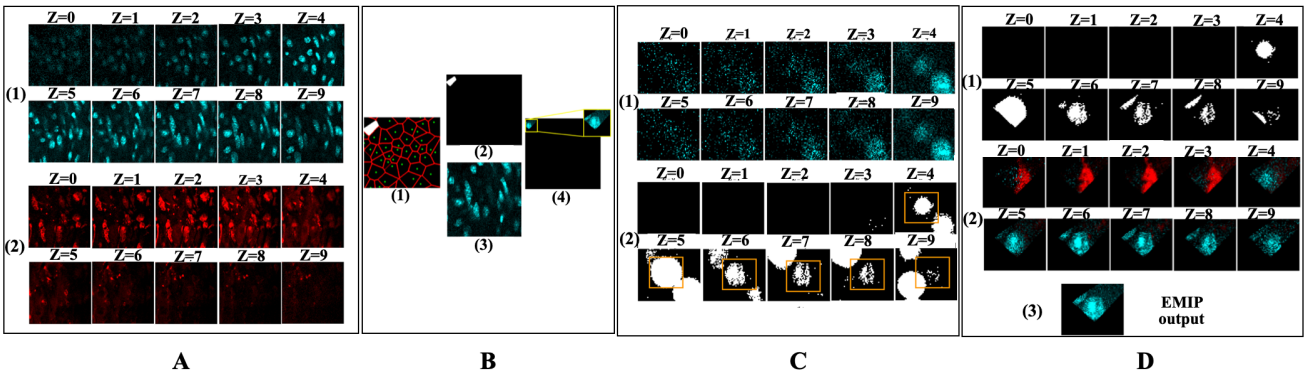


Figure 4.4: (A-1) The nuclei z-slices, (A-2) The marker z-slices, (B-1) The Voronoi label, (B-2) The Voronoi Cell (VC) binary mask associated to convex cell j that assigns label 1 to convex cell j and zero to others, (B-3) The z-slice 6 of nuclei channel, (B-4) The multiplication's output of VC mask and z-slice 6 which depicts the nucleus located in convex cell j, (C-1) nuclei z-slices, (C-2) The 3D binary mask, (D-1) The intersection between VC mask (B-2) and 3D binary mask (C-2), (D-2) the intersection between the VC binary mask and the nuclei/marker z-slices, (D-3) EMIP output

4.3.1 Extended Maximum Intensity Projection (EMIP)

To address the issue of non-overlapping nuclei and markers when applying the MIP approach, the EMIP method is developed. The EMIP approach utilizes the maximum intensity projection for each nucleus separately and only over the z-slices that include that specific nucleus. For example, for a nucleus that spans z-slices from seven to ten (as shown in the white square in Figure 4.2-A), the MIP

should only be applied to those slices. This ensures that the generated 2D image accurately represents the nucleus without mistakenly overlapping with the marker. Figure 4.2-B shows a comparison of the output from the MIP and EMIP approaches for the image shown in Figure 4.4 -A. As depicted, the EMIP approach prevents the lineage tracing marker (depicted in red) from falsely appearing over nuclei with the ground truth label negative. As shown, nuclei in the pink and orange squares with ground truth label negative falsely contain signals of the marker in the output of the MIP, as opposed to the EMIP. The steps of the EMIP are shown in Algorithm 2. To perform the EMIP approach, two types

Algorithm 2: Extended Maximum Intensity Projection

linenosize= **input:** multi-channel images, Point-level annotations

for $i = 1 : N$ *(Number of images)* **do**

1. Generate the 3D distance map (D_i) using point annotations
2. Create the feature map by combining the distance map and nuclei z-slices
3. Apply k-mean clustering on the feature map
4. Identify background cluster (i.e., Min overlap with the dilated point labels)
5. Generate 3D binary masks
6. Generate the 2D Voronoi label using point annotations

   for $j = 1 : N_{cell}$ *(Number of convex cells in the 2D Voronoi label)* **do**

   (a) Generate the Voronoi Cell (VC) binary mask for cell j

   (b) Find the intersection between VC mask and 3D binary mask (I_j^{3D})

   (c) Determine the set of slices (S_j) containing the $Nuclous_j$ by taking summation over the z-slices in I_j^{3D}

   (d) Find the intersection between VC and the nuclei/marker z-slice

   (e) Compute the maximum intensity projection for nuclei and marker channels only over the corresponding slices (S_j) for convex cell j

   end

end

of information are required: (a) which z-slices are associated with each individual nucleus and (b) the

boundaries of each nucleus over the x- and y-axes. Since the boundaries of nuclei are not clear when using point annotations, we utilize a k-mean clustering map and Voronoi label approaches to determine the approximate boundaries of each nucleus over the x-, y-, and z-axes (Steps 1-6 in Algorithm 2). It is essential to note that all nuclei have been annotated with points located at their centers, ensuring we have ground truth point annotations for all nuclei in the dataset. However, due to the nature of 3D images, nuclei are often spread across multiple slices, and the center of each nucleus is only located in one of the slices. Therefore, while every nucleus has a point annotation, these annotations are limited to the z-slice where the nucleus center is present. Consequently, point annotations for all nuclei per slice are unavailable. Using k-means clustering, we generate 3D binary masks (steps 3-5). First, we create a 3D distance map from the distance transform of point annotations (step 1). This map represents distances to the nearest nuclear point. Combining the distance map with nuclei channels of the multi-channel nuclei/marker image creates a features map (step 2). Next, k-means clustering (k=3) is applied to the feature maps, resulting in 3D binary masks. Label 0 represents the background cluster with minimal overlap with dilated point labels, and label 1 corresponds to nuclei pixels. An example of the sequence of the nuclei channels and corresponding 3D binary mask is depicted in Figure 4.4 -C. As shown, the binary mask indicates that the given nucleus in the orange square is spreading only over z-slices from four to nine (i.e., it approximates the nucleus' boundaries over the z-axis). To find the nuclei boundaries on the x- and y-axes, Voronoi labels are created (step 6) using point annotations (Figure 4.4-B-(1)). Assuming that each Voronoi convex cell contains only one nucleus, the Voronoi edges separate all nuclei from each other well. Next, we iterate through Voronoi convex cells (steps a-e) and create a Voronoi Cell (VC) binary mask for each cell, approximating nuclei boundaries on the x- and y-axes. Figure 4.4-B-(2) shows the VC binary mask for convex cell j. Since each cell is assumed to contain only one nucleus, the intersection of the VC binary mask with nuclei/marker channels reveals the nucleus in that cell (Figure 4.4-B-(4)). Likewise, the intersection of the VC and 3D binary masks will reveal only the nucleus mask (represented by the color white) within the corresponding cell (Figure 4.4-D-(1)). As nuclei and background take values of one (i.e., white color) and zero (i.e., black color) respectively, simply, summation over the z-slices can be used as a detector of the presence of the nucleus. If the sum is greater than one, it implies the nucleus is present in that z-slice. Figure 4.4-D-(1) shows that the nucleus corresponding to the convex cell j

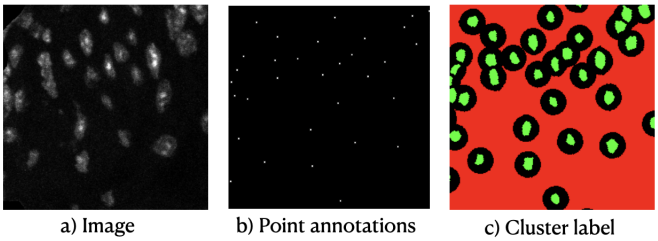


Figure 4.5: (a) Original image, (b) Point annotation, (c) Cluster label which is refined using Voronoi diagram

is spreading over z-slices four to nine, and its boundaries over the x- and y- axes can be determined using the given convex cell edges. Figure 4.4-D-(3) illustrates that a MIP is performed over the z-slices spanning from four to nine for the nucleus situated within the convex cell. This technique avoids the lineage tracing marker, which spreads over slices zero to three, from overlapping with the nucleus.

4.3.2 Supervised Contrastive Learning-based (SCL) training strategy

The Hover-Net model [25] is used for nuclei detection and classification due to its strong generalizability and instance detection performance. As we can not directly use the point-level labels for training the Hover-Net model, we used the cluster label approach proposed in [31] to extract pixel-level labels from point annotation via information obtained from the shape of nuclei in the original image. The generated pixel-wise masks contain three regions: nuclei, background, and an ignored area (Figure 4.5). For training the NP branch, Equation 4.1 is used, employing the cross-entropy (L_{CE}), Dice (L_{Dice}), and entropy-minimization ($L_{entropy}$) loss functions. We adopted a semi-supervised learning approach and used entropy minimization loss function in these unlabeled areas to train the model over them without requiring labels. The entropy minimization loss encourages the model to output confident predictions over these unlabeled areas. The NC branch is trained by using Equation 4.2, employing the cross-entropy (L_{CE}), Dice (L_{Dice}), and SCL (L_{SCL}) loss functions. The Hover branche was trained using the approach from [124]. The L_{CE}, L_{Dice}, L_{SCL} and $L_{entropy}$ losses are computed via Equation 4.3.

$$L_{NP} = L_{CE} + L_{Dice} + L_{entropy} \tag{4.1}$$

$$L_{NC} = L_{CE} + L_{Dice} + L_{SupCon} \tag{4.2}$$

$$\begin{aligned} L_{CE} &= -\frac{1}{n}\Sigma_{i=1}^{N}\Sigma_{k=1}^{K} X_{i,k}(I) log Y_{i,k}(I) \\ L_{Dice} &= 1 - \frac{2\Sigma_{i=1}^{N}(X_i(I) \times Y_i(I)) + \epsilon}{\Sigma_{i=1}^{N} X_i(I) + \Sigma_{i=1}^{N} Y_i(I) + \epsilon} \\ L_{SCL} &= \frac{-1}{|P(q)|} \sum_{q^+ \in P(q)} log \frac{exp(q \cdot q^+/\tau)}{exp(q \cdot q^+/\tau) + \sum_{q^- \in N(i)} exp(q \cdot q^-/\tau)} \\ L_{entropy} &= -\Sigma_{i=1}^{N}\Sigma_{k=1}^{K} Y_{i,k}(I) log Y_{i,k}(I) \end{aligned} \tag{4.3}$$

where Y, X, K, N, and $\epsilon(1.0e-3)$ are the prediction, ground truth, number of classes, number of images, and smoothness constant, respectively. The Cross-Entropy Loss function has two limitations: 1) It penalizes pixel-wise predictions independently without considering their relationships, and 2) It does not directly supervise the learned representations. HoVer-Net improves upon the Cross-Entropy Loss function by incorporating the Dice loss function, which considers pixel dependencies within an image. However, the Dice loss function does not account for global semantic relationships across images. To address the issue, we enhance our model's performance by incorporating Pixel-to-Pixel and Pixel-to-Region Supervised Contrastive Learning (SCL) [125] techniques alongside cross-entropy and Dice losses in the third branch. We introduce a projection head in the NC branch, outputting the embedding q per pixel, which is optimized using the last row of Equation 1. where, $p(q)$ and $N(q)$ indicate the set of positive and negative embedding samples, respectively.

4.4 Experimental Results

Metrics: To evaluate the model's performance, we utilize the popular detection/classification metrics: precision ($P = \frac{TP}{TP+FP}$), recall ($R = \frac{TP}{TP+FN}$), and F1-score ($F1 = \frac{2TP}{2TP+FP+FN}$).

Datasets: We experimented with three datasets: **Cardiovascular dataset 1 (D1)** and **Cardiovascular dataset 2 (D2)**, containing advanced atherosclerotic lesion images from two mouse models.

D1 has 13 images, with 11 used for training and 2 for testing. These images vary in size along the z-axis (8 to 13). We extract 256 × 256-pixel patches with 10% overlap. The train and test sets have 370 and 74 patches respectively. Additionally, we have a separate evaluation set called D2 (29 images) that are used for further evaluation. Our aim was to developed a label-efficient model that achieves comparable results with minimum labeling effort, so we trained our model on the smaller dataset. Please refer to Table 4.1 for more details. Additionally, we used the **CoNSeP dataset** [1] [25], which contains 24,332 nuclei from 41 whole slide images (26 for training and 14 for testing), with 7 different classes: fibroblast, dysplastic/malignant epithelial, inflammatory, healthy epithelial, muscle, other, and endothelial.

Table 4.1: Cardiovascular datasets

Charecteristices	**Dataset**	**Value**
Model	D1	$Myh11 - CreER_{T2} - RADROSA26 - STOP_{flox} - tdTomApoe - /-$
	D2	$Myh11 - CreER_{T2} - RADROSA26 - STOP_{flox} - tdTomIrs1_{\Delta/\Delta}Irs2_{\Delta/\Delta}$
Diet	D1	Western diet for 18 weeks
	D2	Western diet for 18 weeks

Test time: We combine nuclei and marker channels per slice using the linear combination method (Figure 4.1-a). The model detects and classifies nuclei in each slice individually. The final output is integrated over the slices with this rule: If a nucleus is predicted positive in at least one slice, it is labeled positive, otherwise negative.

Results: Table 4.2 shows the performance of different approaches on D1 and D2. The first row shows the results of using regular MIP during both the training and test stages, while the second row shows the model's performance trained using EMIP. The NP branch indicates the model's performance in detecting nuclei, and the NC branch denotes the model's performance in classifying the type of detected nuclei. As observed, the EMIP approach improves precision and F1 score metrics by 16.43%

[1] https://warwick.ac.uk/fac/cross_fac/tia/data/hovernet/

Table 4.2: The performance of our methods on D1 and D2

Model	Branch	D1			D2		
		Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
HoVer-Net [25] (MIP)	NP	**0.8898**	0.8894	0.8883	**0.9233**	0.8455	0.8816
	NC	0.6608	0.8511	0.7424	0.7150	0.6663	0.6703
HoVer-Net [25] (EMIP)	NP	0.8551	**0.9353**	0.8880	0.9064	0.8743	0.8894
	NC	0.7694	0.7800	0.7718	0.8114	0.7718	0.7760
Qu et al. [31] (EMIP)	NP	0.7774	0.8489	0.8084	0.6525	0.877	0.7431
	NC	**0.8548**	0.6048	0.6881	0.8140	0.5749	0.6577
LECL	NP	0.8764	0.9154	**0.8942**	0.9215	**0.877**	**0.8978**
	NC	0.8277	**0.7668**	**0.7890**	**0.8392**	**0.7840**	**0.7953**

Table 4.3: The effect of SCL based training approach on the CoNSep dataset

Model	F_d	F_c^e	F_c^i	F_c^s	F_c^m
HoVer-Net [25] w/o SCL (Weakly)	0.735	0.578	0.542	0.461	0.147
HoVer-Net [25] w SCL (Weakly)	0.738	0.576	0.551	0.480	0.212

and 3.96% on D1, respectively, indicating a decrease in false positives. To ensure a comprehensive evaluation of our method, we have included Dataset D2 in our study. The selection of D2 was based on its larger size and representativeness, making it suitable for robust performance assessment. As observed, the EMIP approach achieves higher precision, recall, and F1 scores than the MIP method. The study found that the EMIP approach reduces false positives in lineage tracing markers overlapping with nuclei. Furthermore, we compare the performance of the HoVer-Net [25] model with Qu *et al.*[31] on both datasets D1 and D2. Hyper-parameters for Qu *et al.*[31] was borrowed from [31]. As observed, the HoVer-Net model [25] outperforms Qu*et al.*[31] in both nuclei detection and classification. We investigate the benefits of combining SCL-based training and EMIP in the LECL model. The SCL loss enhances the model's performance by capturing global semantic relationships between pixel samples, resulting in better intra-class compactness and inter-class separability. On both D1 and D2, the LECL model outperforms other models. For visualization examples, refer to Figure 4.6. Furthermore, the

hyperparameters for all experiments have been provided in Table 4.4.

Table 4.4: Training setup for all experiments

Characteristic	**Value**	**Characteristic**		**Value**
Pytorch	1.10	**Loss function weights**	Entropy	0.5
			Cross-entropy	1
			Dice loss	1
GPU	Tesla p100	**Number of epochs**		100
Projection head	**Two convolutional layers, outputting a 256 $l2$-normalized feature vector**			

Ablation study: To investigate further the performance of the SCL-based HoVer-Net, we evaluate the model on the ConSep dataset (Table 4.3). Here, F_d represents the F1-score for nuclei detection, while F_c^e, F_c^i, F_c^s, and F_c^m indicate the F1-scores for epithelial, inflammatory, spindle-shaped, and miscellaneous, respectively. The SCL-based model achieves better performance.

4.5 Conclusion

Developing an automated approach for 3D nuclei detection and classification in fluorescent images requires expensive pixel-wise annotations. To overcome this, we designed the LECL model, which includes the EMIP and SCL components. The EMIP approach improves upon the limitations of the MIP approach, while the SCL learning approach enhances the model's performance by learning more discriminative features.

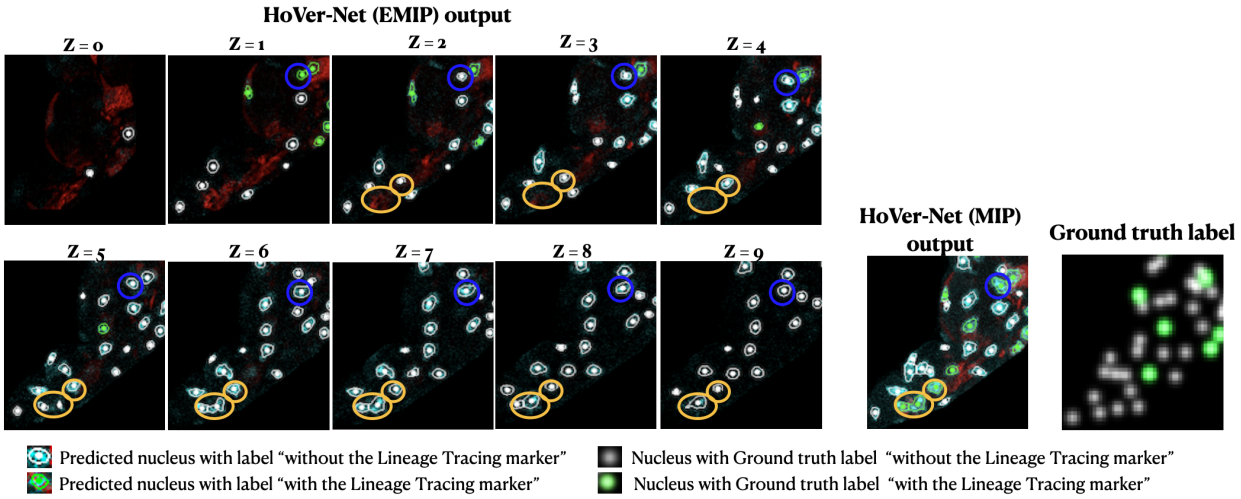


Figure 4.6: The model's output trained using the HoVer-Net(MIP) and HoVer-Net(EMIP). The HoVer-Net (EMIP) model correctly predicts label positive for nuclei in the yellow circle, which is consistent with the ground truth labels. In contrast, HoVer-Net (MIP) incorrectly predicts these nuclei as negative. Both models incorrectly predict the nuclei's labels in the blue circle.

Chapter 5

ProtoGMM: Multi-prototype Gaussian-Mixture-based Domain Adaptation Model for Semantic Segmentation

5.1 Introduction

In this dissertation, we present a Multi-prototype Gaussian-Mixture-based model (ProtoGMM) that overcomes the limitations of existing UDA methods. The schematic representation of the ProtoGMM model is shown in Figure 5.1. Unlike prevailing self-training approaches that focus solely on the discriminative classifier of $p(class|pixel feature)$, ProtoGMM adopts a hybrid training approach, integrating both discriminative and generative models. The core of the ProtoGMM framework lies in modeling the underlying distribution of source pixel features using generative Gaussian Mixture Models (GMMs) (Figure 5.1-a), optimized through Expectation-Maximization (EM). This novel approach allows ProtoGMM to adapt effectively to multimodal data densities. Rather than relying on noisy pseudo-labels from the discriminative classifier, ProtoGMM leverages the generative GMM model for

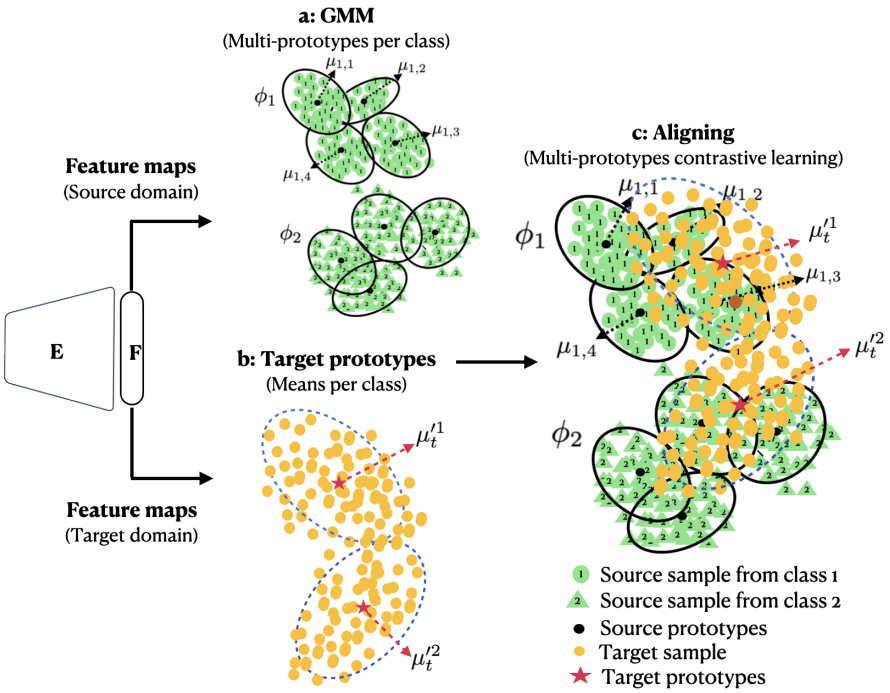


Figure 5.1: Diagram of the ProtoGMM model

more efficient contrastive learning. The components of the Gaussian mixture model, which describe the underlying distribution of pixel representations, serve as the most suitable representative prototypes for contrastive losses. The ProtoGMM approach is rooted in the domain closeness assumption [126], which suggests that features from two domains cluster together in a shared space, and clusters with identical semantic labels are located in close proximity. Based on this foundational insight, the ProtoGMM method departs from using noisy pseudo-labels from the discriminative classifier to perform contrastive learning. Instead, it determines the positive and negative clusters for the given target sample by considering the underlying distribution of the source domain and the category prototypes of the target domain. This, in turn, allows the method to identify positive and negative samples with greater precision, enabling more guided and effective contrastive learning. By incorporating the GMM-based model with contrastive learning loss, ProtoGMM functions as a generative model, significantly improving the performance of the domain adaptation model when used alongside the discriminative classifier. In addition to its advantages in contrastive learning, ProtoGMM excels at addressing the label distribution shift, a common challenge in UDA tasks. The major contributions of this study are

summarized as follows:

1. We present the ProtoGMM model for UDA in semantic segmentation.
2. By utilizing guided contrastive learning, our approach enhances self-supervised learning, elevating intra-class semantic similarity and reducing inter-class similarity between source and target domains.
3. We address biases inherent in discriminative classifiers by combining a generative model with a discriminative model.
4. We showcase the superior performance of our approach compared to the current state-of-the-art in two scenarios: 1) GTA → Cityscapes, 2) Synthia → Cityscapes

5.2 Methodology

5.2.1 Problem formulation

In the UDA, $p_s(x, y) \in p_{\mathcal{S}}$ and $p_t(x, y) \in p_{\mathcal{T}}$ are the underlying source and target domain distributions, respectively. Then, the labeled data $\mathcal{D}_S$ (i.e. $x_s \in R^{H \times W \times 3}$ and $y_s \in R^{H \times W \times C}$) is sampled i.i.d from the source domain distribution (i.e. $p_s(x, y)$) and unlabeled data $\mathcal{D}_T$ (i.e. $x_t \in R^{H \times W \times 3}$) is selected i.i.d from marginal target domain distribution (i.e. $p_t(x)$). Here, H and W represent the height and width of the images, respectively, and C denotes the number of classes. The primary goal of UDA is to train the model using both $\mathcal{D}T$ and $\mathcal{D}S$ to enhance the model's performance on the target domain. The model itself is composed of three components: an encoder (E), a multi-class segmentation head (CL), and an auxiliary projection head (F). When given an input image x, the auxiliary projection head processes the encoder's output to obtain a feature map ($f = F(E(x))$). All features are mapped to the $l2$-normalized feature vector. Subsequently, the multi-class segmentation head operates on the encoder's output to produce a class probability map ($pred = CL(E(x))$). We utilize the cross-entropy loss (L_{ce}) and ProtoGMM loss functions to train the model. The cross-entropy loss is computed for the source and target domain images using their ground truth labels (y_i^s) and pseudo labels ($\hat{y}_i^t$) as

follows:

$$
\begin{aligned}
L_{ce}^{s} &= -\sum_{i=1}^{H\times W}\sum_{c=1}^{C} I_{[y_i^s=c]} log(pred_{s,i,c}) \\
L_{ce}^{t} &= -\sum_{i=1}^{H\times W}\sum_{c=1}^{C} w_{t,i,c} I_{[\hat{y}_i^t=c]} log(pred_{t,i,c}) \\
\hat{y}_i^t &= \underset{c}{argmax}\ pred_{t,i,c}, \qquad I \in \{1,2,..,H\times W\}
\end{aligned}
\tag{5.1}
$$

To reduce pseudo-label noise, we applied confidence weights $w_{t,i,c} = \frac{\sum_{i=1}^{H\times W} 1_{[\underset{c}{max}\ pred_{i,c}^t]>\beta}}{H\times W}$ [51]. We adopt the teacher-student architecture [127] and the same framework used in [51] as the strong backbone. The weights of the teacher network are assigned as the exponential moving average (EMA) of the student network's weights in every iteration.

Algorithm 3: ProtoGMM model

Initialize the weights of the model.

for *Iter = 1:* N_{Iter} **do**

  for $n \in 1,...,N_{batch}^{s}$ *(source minibatch)* **do**

    Update source pixel data distribution GMM model $\{\phi_c^*\}$ using Sinkhorn EM

    if $Iter > Iter_{dist}$ **then**

      Apply source domain multi-prototype Contrastive Learning for the feature map f_s

    end

    Update the source prior distribution δ_{source}^{c}

  end

  for $n \in 1,...,N_{batch}^{t}$ *(target minibatch)* **do**

    Update the target prior distribution δ_{target}^{c}

    Update target bank by choosing reliable f_t

    if $Iter > Iter_{dist}$ **then**

      Aligning source and target domain by applying target domain multi-prototype

      Contrastive Learning for the feature map f_t

    end

  end

end

5.2.2 ProtoGMM model

Our primary goal is to enhance the performance of domain adaptation techniques by improving the alignment of features between source and target domains. While most self-training techniques rely heavily on domain alignment methods, they often neglect the importance of precise domain alignment [51, 38]. To tackle this issue, we developed a novel approach called multi-prototype-guided alignments in the embedding space. Our method involves identifying the most representative prototypes per category and utilizing them to align the source and target domains. However, the challenge lies in finding prototypes that can effectively capture the diversity in semantic concepts for each category. If sub-class labels are available, we can use them to define these prototypes for each class. Another possible approach is to utilize global category prototypes from the source domain to guide the alignment between the source and target domains. However, this method has limitations, as global prototypes only capture the common characteristics of each category and do not fully leverage the potential strength of semantic information [51]. Moreover, this approach is based on the unimodality assumption of each category, ignoring within-class variations. To overcome these challenges, we introduce the ProtoGMM approach, which aims to address the issues associated with existing methods and improve domain alignment for enhanced domain adaptation. In this approach, we estimate the underlying multi-prototype source distribution by employing the GMM model on the feature space of the source samples. The components of the GMM model serve as the most suitable representative prototypes. The GMM model adapts effectively to multimodal data density, capturing within-class variations. In this approach, to increase intra-class semantic similarity and decrease inter-class similarity across the source domain, we compute the multi-prototype contrastive learning loss between source pixel embeddings and the source prototypes. As illustrated in Algorithm 3, during each iteration, we first update the Gaussian Mixture Model (GMM) of the source pixel data distribution using the high-dimensional l_2-normalized features f_s of the source pixels. Subsequently, we compute the multi-prototype Contrastive Learning for the source sample features (f_s). For the semantic alignment of the source and target domain on the feature space, we perform contrastive learning between the embedding of the target samples and source domain multi-prototypes, as shown in Algorithm 3. Since the model is biased toward the source domain and exhibits a discrepancy between the source and target domains,

we perform an alignment mechanism to identify reliable prototypes for the given target sample. Our approach's details are elaborated in the following sections.

5.2.3 Multiprototype source domain distribution

The goal is to represent the Multiprototype source domain joint distribution $p(f_s, c)$ in the latent feature space as shown in Figure 5.1-a. To achieve this, we can approximate the joint distribution by estimating the class conditional distribution $p(f_s|c)$ together with the class prior $p(c)$:

$$p(f_s, c) = p(f_s|c)p(c) \tag{5.2}$$

In our approach, we establish uniform class source priors, achieved through the adoption of a class-balanced sampling technique called rare class sampling (RCS) proposed in [50]. By employing RCS, each class is equally represented during training, leading to a balanced distribution. To further enhance the capabilities of our model, we employ generative GMMs to estimate the class-conditional distribution $p(f_s|c)$ for each category. This innovative technique enables ProtoGMM to adapt adeptly to datasets with multiple modes of data densities. The GMM consists of a weighted mixture of M multivariate Gaussians, which effectively models the pixel data distribution of each class c in the D-dimensional feature space, as shown in Equation 5.3.

$$\begin{aligned} p(\boldsymbol{f_s}|c; \boldsymbol{\phi_c}) &= \Sigma_{m=1}^{M} p(m|c; \boldsymbol{\pi}_c) p(\boldsymbol{f_s}|c, m; \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}) \\ &= \Sigma_{m=1}^{M} \pi_{cm} \mathcal{N}(\boldsymbol{f_s}; \boldsymbol{\mu_{cm}}, \boldsymbol{\Sigma_{cm}}) \end{aligned} \tag{5.3}$$

where, π_{cm} is a prior probability for each class and $\Sigma_{m=1}^{M} \pi_{cm} = 1$; Σ_c and μ_c are the covariance matrix and mean vector.

The GMM classifier is parameterized by $\{\phi_c^* = \{\boldsymbol{\pi_c}, \boldsymbol{\mu_c}, \boldsymbol{\Sigma c}\}\}_{c=1}^{C}$ and is optimized online using a momentum-based version of the (Sinkhorn) EM (Expectation-Maximization) algorithm, as proposed in [49]. The objective of the EM method is to maximize the log-likelihood over the feature-label pairs, which is expressed as follows:

$$\phi_c^* = \arg\max_{\phi_c} \sum_{f_s : y_s = c} log \sum_{m=1}^{M} p(f_s, m|c; \phi_c) \tag{5.4}$$

5.2.4 Source domain multi-prototype Contrastive Learning

We apply contrastive learning between source pixel embeddings and class prototypes, calculated as the means of GMM components for each class, using Equation 5.5. The question that arises is how to select negative and positive prototypes for the given source sample. As previous works have confirmed the significance of hard negatives in metric learning [128], we perform a hard sampling mechanism to enable multi-prototype contrastive learning on the source domain. Employing Bayes' rule and assuming uniform class priors, we compute the posterior using Equation 5.6. The probability $p_s(m|f_s, c; \phi_c^*)$ indicates the likelihood of data f_s being assigned to component m in class c.

Based on the values of $p_s(m|f_s, c; \phi_c^*)$ for the given source sample, considering its ground truth label c, we choose the prototype corresponding to the mean of the closest component with the same label as a positive prototype (Equation 5.7). Moreover, from the GMM distribution of the categories with different labels, we select the closest component as the hardest component for each category as hard negative prototypes:

$$l_{protoGMM} = -log\frac{e^{f_s q^+/\tau}}{e^{f_s q^+/\tau} + \Sigma_{n=1}^{N} e^{f_s q_c^-/\tau}} \tag{5.5}$$

$$p_s(m|f_s, c; \phi_c^*) = \frac{\pi_{c,m}\mathcal{N}(f_s|\mu_{c,m}, \Sigma_{c,m})}{\Sigma_{m'=1}^{M}\pi_{c,m'}\mathcal{N}(f_s|\mu_{c,m'}, \Sigma_{c,m'})} \tag{5.6}$$

$$q^+ = \{\mu_{c,m^+}|\ m^+ = \arg\max_m\ p_s(m|f_s, c; \phi_c^*), c = y^s\} \tag{5.7}$$

$$q_c^- = \{\mu_{c,m^-}|\ m^- = \arg\max_m\ p_s(m|f_s, c; \phi_c^*), c\} \forall c \neq y^s \tag{5.8}$$

5.2.5 Prior distribution update

We update the prior distribution of the target and source domain using Equation 5.9. The equation functions for both domains and the index d indicates whether it pertains to the source or target domains. Where, δ_{Iter}^c denotes the proportion of pixels belonging to the c-th category in the given iteration,N_{batch}^d is a number of the images in the given minibatch, $H \times W$ is the multiplication of the height and width of the image indicating the total number of the pixels, $y_{n,i}^d$ shows the pixel's ground

truth label for the source domain and pseudo label for the target domain.

$$\begin{aligned} \delta^c_{Iter} &= \frac{1}{N^d_{batch} \times H \times W} \Sigma_{n=1}^{N^d_{batch}} \Sigma_{i=1}^{H \times W} y^d_{n,i} \\ \delta^c_d &= \alpha \delta^c_d + (1-\alpha)\delta^c_{Iter} \qquad\qquad \forall d = \{s,t\} \end{aligned} \tag{5.9}$$

5.2.6 Update target bank

The target bank is updated in each iteration by incorporating reliable pixel embeddings from each target mini-batch. To identify these reliable embeddings, we begin by computing the average pixel embedding per class within the given mini-batch, utilizing their pseudo labels (Equation 5.10). Next, we assess their cosine similarity with the average mean per class and select the M pixel embeddings with the highest cosine similarity scores as the most reliable representations using Equation 5.11. In each iteration, M represents the selected pixel embeddings for updating the target memory bank.

$$\mu'^c_t = \frac{\Sigma_i^{N^t_{batch} \times H \times W} f_{t,i} \times I(\hat{y}_{t,i} = c)}{\Sigma_i^{N^t_{batch} \times H \times W} I(\hat{y}_{t,i} = c)} \tag{5.10}$$

$$\begin{aligned} S &= \{(s_i, f_i) | s_i = cosine(\mu'^c_t, f_{t,i})\} \\ S^* &= \underset{s_i}{sort}(S)[:M] \end{aligned} \tag{5.11}$$

5.2.7 Target domain prototypes

We estimate the underlying distribution of the target domain by computing the class prototypes, as shown in Figure 5.1-b. The target domain prototypes per category will be updated using the target memory bank and the exponential moving average as follows:

$$\mu^c_t = \alpha \mu^c_t + (1-\alpha)\mu'^c_t \qquad\qquad \forall d = \{s,t\} \tag{5.12}$$

Noted, We employ Class-balanced Cropping (CBC) [51] on the unlabeled target image, a technique that encourages the model to prioritize cropping from regions with multiple classes.

5.2.8 Aligning source and target domain distribution

To align the distributions of the source and target domains, we employ multi-prototype Contrastive Learning between the target pixel embeddings and source multi-prototypes (Figure 5.1-c). Since the true labels of target samples are unavailable, we assign a pseudo label to the given target sample using the posterior probability $p_t(c|f_t;\phi_c^*)$ and its similarity to the target prototypes, as shown in Equation 5.13. For example the sample shown with the orange color is close to the target prototype of class 1 and source component 3 of class 1, as depicted in Figure 5.1-c. The posterior probability represents the likelihood of the given target sample belonging to class c. The rationale for utilizing the posterior probability to assign pseudo labels is grounded in the domain closeness assumption. This assumption suggests that features from two domains are clustered in a shared space, wherein clusters with identical semantic labels are situated close to each other [38]. With this premise, we posit that the target sample should be in proximity to the source domain distribution with the same category within the feature space.

$$\hat{y}_t = \arg\max_c \; p_t(c|f_t;\phi_c^*) \times \frac{e^{cosine(\mu_c^t, f_t)}}{\Sigma_{c'} e^{cosine(\mu_{c'}^t, f_t)}} \tag{5.13}$$

where the first term is the posterior probability and is computed using Proposition 1; the second term computes the cosine similarity of the given target sample with the target prototype. The second term corrects the posterior probabilities of class c for the given target sample based on its similarity to the target prototype of class c.

Proposition 1: Given $p_s(c|f_t;\phi_c^*) = \sum_{m'} p_s(c,m'|f_t;\phi_c^*)$, the posterior probability for the given target sample f_t is computed as follows:

$$p_t(c|f_t;\phi_c^*) = p_s(c|f_t;\phi_c^*) \times \frac{\delta_{target}^c}{\delta_{source}^c} \tag{5.14}$$

Noted, adjusting the posterior using the ratio $\frac{\delta_{target}^c}{\delta_{source}^c}$ addresses the issue of label shift as noted in [52].

Proof. Based on the Bayes rule, We have the below relationship for the posterior probability trained

Figure 5.2: Qualitative analysis on GTA → Cityscapes (first row) and Synthia → Cityscapes (second row). The ProtoGMM shows a clear visual improvement.

on the source (p_s) and the target (p_t) domains.

$$\begin{aligned} p_t(c|f_t;\phi_c^*) \,\alpha\, p_t(f(x)|c;\phi_c^*)p_t(c) \\ p_s(c|f_t;\phi_c^*) \,\alpha\, p_s(f(x)|c;\phi_c^*)p_s(c) \end{aligned} \tag{5.15}$$

If we assume the conditional data distribution is well aligned, i.e. $p_t(f(x)|c) = p_s(f(x)|c)$. We can extract the below relationship between the posterior probabilities using Equation 5.15.

$$p_t(c|f_t;\phi_c^*) \;= p_s(c|f_t;\phi_c^*) \times \frac{p_t(c)}{p_s(c)} \quad = p_s(c|f_t;\phi_c^*) \times \frac{\delta^c_{target}}{\delta^c_{source}} \tag{5.16}$$

□

To perform the multi-prototype Contrastive Learning between the target pixel embeddings and source multi-prototypes, the positive and negative prototypes are chosen using Equation 5.17.

$$\begin{aligned} q^+ &= \{\mu_{c,m^+} |\; m^+ = \arg\max_m \; p_t(m|f_t, c;\phi_c^*), c = \hat{y}_t\} \\ q_c^- &= \{\mu_{c,m^-} |\; m^- = \arg\max_m \; p_t(m|f_t, c;\phi_c^*), c\} \forall c \neq \hat{y}_t \end{aligned} \tag{5.17}$$

5.3 Experiments

5.3.1 Datasets

- **Cityscapes [129]:** This dataset comprises real urban scenes captured across 50 cities in Germany and nearby nations. This dataset has segmentation masks with 19 distinct categories

Table 5.1: Comparison with state-of-the-art methods for UDA

GTA5→Cityscapes																				
Model	Road	S.Walk	Build.	Wall	Fence	Pole	T.Light	T.Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
AdaptSeg	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4
CBST	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
DACS	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
CorDA	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	47.0	89.7	66.7	35.9	90.2	48.9	57.5	0.0	39.8	56.0	56.6
BAPA	94.4	61.0	88.0	26.8	39.9	38.3	46.1	55.3	87.8	46.1	89.4	68.8	40.0	90.2	60.4	59.0	0.0	45.1	54.2	57.4
ProDA	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
DAFormer	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
DAFormer+ **ProtoGMM**	**97.32**	**79.53**	90.13	55.57	52.22	53.69	58.18	63.38	90.53	49.53	91.83	74.56	46.42	93.25	73.21	79.96	68.74	53.77	65.41	70.4
HRDA	96.4	74.4	91	**61.6**	51.5	57.1	**63.9**	69.3	**91.3**	48.4	94.2	79.0	52.9	93.9	84.1	85.7	75.9	63.9	**67.5**	73.8
HRDA+ **ProtoGMM**	96.83	77.31	**90.9**	60.5	**55.11**	**59.87**	62.88	**73.59**	90.75	**49.96**	**94.78**	**79.22**	**53.48**	**94.7**	**86.77**	**89.45**	**78.23**	**65.3**	67.34	**75.1**
Synthia→Cityscapes																				
Model	Road	S.Walk	Build.	Wall	Fence	Pole	T.Light	T.Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
AdaptSeg	79.2	37.2	78.8	-	-	-	9.9	10.5	78.2	-	80.5	53.5	19.6	67.0	-	29.5	-	21.6	31.3	37.2
CBST	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	-	78.3	60.6	28.3	81.6	-	23.5	-	18.8	39.8	42.6
DACS	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	-	90.8	67.5	38.3	82.9	-	38.9	-	28.5	47.6	48.3
CorDA	93.3	61.6	85.3	19.6	5.1	37.8	36.6	42.8	84.9	-	90.4	69.7	41.8	85.6	-	38.4	-	32.6	53.9	55.0
BAPA	91.7	53.8	83.9	22.4	0.8	34.9	30.5	42.8	86.6	-	88.2	66.0	34.1	86.8	-	51.3	-	29.4	50.5	53.3
ProDA	93.3	61.6	85.3	19.6	5.1	37.8	36.6	42.8	84.9	-	90.4	69.7	41.8	85.6	-	38.4	-	32.6	53.9	55.0
DAFormer	84.5	40.7	88.4	41.5	6.5	50.0	55.0	54.6	86.0	-	89.8	73.2	48.2	87.2	-	53.2	-	53.9	61.7	60.9
DAFormer+ **ProtoGMM**	**93.4**	**64.3**	87.8	23.5	**14.1**	53.6	60.1	59.4	86.3	-	88.6	65.2	49.5	89.3	-	62.3	-	52.3	63.6	63.3
HRDA	85.2	47.7	**88.8**	**49.5**	4.8	57.2	65.7	60.9	85.3	-	92.9	79.4	52.8	89.0	-	**64.7**	-	63.9	64.9	65.8
HRDA+ **ProtoGMM**	91.92	59.15	88.67	46.16	4.47	**59.59**	**66.63**	**62.26**	**87.48**	-	**94.07**	**81.08**	**57.77**	**91.5**	-	50.42	-	**65.25**	**66.92**	**67.09**

Table 5.2: Comparison with state-of-the-art Contrastive learning approaches

GTA→Cityscapes			
Model	BankCL	UniProto	ProtoGMM
mIoU	69.12	69.48	**70.4**
Synthia→Cityscapes			
Model	BankCL	UniProto	ProtoGMM
mIoU	61.46	61.71	**62.88**

for 2,975 training images and 500 validation images, all at a resolution of 2048×1024 pixels. We use the unlabeled training set as a target domain in our experiments, while evaluations are conducted using the corresponding validation set.

- **GTA [53]:** The dataset comprises 24,966 synthetic images extracted from the immersive open-world game "Grand Theft Auto V", each accompanied by corresponding semantic segmentation maps. These images have a resolution of 1914x1052. This dataset is used as a source domain and has 19 common semantic annotations with the Cityscapes dataset.

- **Synthia [54]:** It is a synthetic dataset encompassing 9400 photo-realistic frames with a resolution of 1280 × 960. These frames, rendered from a virtual, are paired with pixel-level annotations. This dataset serves as the source domain and shares 13 common semantic annotations with the Cityscapes dataset.

5.3.2 Implementation Details

Network architecture: For DAFormer+ProtoGMM and HRDA+ProtoGMM, we adopt the same framework and mainstream pipelines as suggested in [50] and [130], respectively. Furthermore, we incorporate two 1 × 1 convolutional layers with ReLU [51] as a projection head into the network, which maps the high-dimensional pixel embeddings into a 64-dimensional $l-2$ normalized feature vector (D=64). Also, the covariance matrices $\Sigma \in R^{D\times D}$ used in GMM are restricted to a diagonal structure.

Table 5.3: Number of components

Model	**GTA5→City.**				**Synthia→City.**			
M	1	3	5	7	1	3	5	7
mIoU	69.2	69.7	**70.4**	69.6	61.7	**63.3**	62.9	61.2

Table 5.4: Comparison with CL baseline methods

Model	UniProto	BankCL	ProtoGMM
mIoU	60.0	59.42	**61.4**

Training: We follow the same training regime as described in [50] and [130] for DAFormer+ProtoGMM, and HRDA+ProtoGMM, respectively. All models are developed using PyTorch 1.8.1 and trained on a single NVIDIA Tesla V100-32G GPU. In DAFormer+ProtoGMM and HRDA+ProtoGMM, We used the AdamW optimizer [131] and set the betas set and weight decay at (0.9, 0.999) and 0.01. We incorporate the learning rate warmup policy as same as [50]. We set α to 0.9 and the EMA weight update parameter (β) for the teacher network to 0.999. We train the the DAFormer+ProtoGMM model for 60000 epochs and the HRDA+ProtoGMM model for 80000 epochs. For the generative optimization of GMM, we follow the same framework from GMMSeg [49]. In each iteration, we perform one iteration of the momentum (Sinkhorn) Expectation-Maximization (EM) process, on both the current training batch and the external memory. The size of external memory is 32K pixel features per category and the number of components per category is set to 5. This memory is updated by the first-in, first-out matter and by selecting 100 pixels per class from each image. The size of the target memory bank is 16K pixel features per class. Our evaluation metric employs per-class intersection-over-union (IoU), mean IoU across all classes, precision, recall, and F1-score for different scenaios.

5.3.3 Comparison with existing UDA methods

We compare the ProtoGMM with existing methods. We show that the ProtoGMM improves the performance of existing UDA methods in two representative synthetic-to-real adaptation scenarios: GTA5 → Cityscapes and Synthia → Cityscapes in Table 5.1. The UDA methods include AdaptSeg

[94], CBST [132], DACS [133], CorDA [134], BAPA [105], ProDA [45], and DAFormer [50]. Our results reveal that DAFormer+ProtoGMM surpasses the current method by a notable margin of +2.1 mIoU in the GTA5 → Cityscapes scenario and +2.4 mIoU in the Synthia → Cityscapes case, as outlined in Table 5.1. Additionally, we note that HRDA+ProtoGMM exhibits superior performance over the existing method, achieving a margin of +1.3 mIoU in the GTA5 → Cityscapes scenario and +1.29 mIoU in the Synthia → Cityscapes case, as detailed in Table 5.1. Figure 5.2 shows that the ProtoGMM approach improves the performance of the HRDA model in classes like wall, walk-side, Bus, Sign, etc indicated by white dotted boxes.

Ablation study:

The comparison between the ProtoGMM and state-of-the-art pixel contrast methods, including UniProto and BankCL, is illustrated in Table 5.4. In the UniProto approach, global class prototypes are utilized as both positive and negative samples. This results in one positive class and $C-1$ negative classes per sample. Conversely, the BankCL method employs an approach that incorporates multiple positive and negative samples from a memory bank [51]. This involves storing the local category centroids of individual source images in the memory bank. As Table 5.4 indicates the ProtoGMM outperforms both BankCL and UniProto methods in both GTA5 → Cityscapes and Synthia → Cityscapes cases. Table 5.3 shows optimal values for M: 5 for GTA5 and 3 for Synthia. Furthermore, to illustrate our method's effectiveness, we compared the DAFormer+ProtoGMM model with the DeepLab-V2 backbone to baseline CL methods (BankCL, UniProto) for the GTA5 → Cityscapes scenario. As observed the DAFormer+ProtoGMM model outperforms the baseline ones.

5.4 Conclusion

In this dissertation, we introduce the protoGMM model which involves estimating the multi-prototype source distribution by using GMM models in the feature space. The GMM components serve as representative prototypes, effectively adapting to the diversity of the data and capturing variations within classes. To enhance intra-class semantic similarity, reduce inter-class similarity, and align the source and target domains, we apply multi-prototype CL between the source distribution and target samples. Experimental results demonstrate the effectiveness of our approach on the UDA benchmarks.

Chapter 6

GenGMM: Generalized Gaussian-Mixture-based Domain Adaptation Model for Semantic Segmentation

6.1 Introduction

In this dissertation, we introduce a novel domain adaptation setting called Generalized Domain Adaptation (GDA) which possesses the following characteristics: 1) Partially or noisy labeled source data, 2) Weakly or unlabeled target data. The GDA setting relaxes the problem of UDA by allowing the use of unlabeled or weakly labeled data from the source domain and weak labels from the target domain. This work addresses domain adaptation challenges in GDA settings with partially labeled source and target data. We introduce a Generalized Gaussian mixture-based (GenGMM) Domain Adaptation Model, leveraging the source and target domain distributions to enhance the quality of weak and pseudo labels and achieve alignment between the source and target domains. Instead of solely relying on noisy pseudo-labels generated by a discriminative classifier, GenGMM employs generative Gaussian

Mixture Models (GMMs) for both the source and target domains to facilitate more efficient contrastive learning and alignment as well as addressing the issue of partially labeled source and target domain. The GenGMM approach is based on two key principles: 1) the domain closeness assumption [126] and 2) feature similarity between labeled and unlabeled pixels [135]. The first principle implies that both domains inherently cluster in a shared space, with items of identical semantic labels situated closely within these clusters. The second principle highlights that unlabeled pixel embeddings tend to be closer to labeled pixel embeddings with matching semantic labels within each domain. Expanding on these principles, GenGMM diverges from the common approach of relying on unreliable pseudo-labels generated by the discriminative classifier for contrastive learning. Instead, it identifies positive and negative clusters for a given target sample by considering the underlying distribution of both source and target domains. We estimate the source domain's underlying distribution using the generative GMM, optimized through Expectation-Maximization (EM) on labeled source pixel features. There are three distinct advantages in estimating the source pixel feature distribution using the GMM. First, it adapts effectively to multimodal data densities. Second, by capturing category-wise Gaussian mixtures for feature representations, the components of the GMM can serve as the most suitable representative prototypes for contrastive loss to align the source and target domain. Third, the GMM model provides reliable information from labeled pixels to refine the unlabeled or weakly labeled pixels, achieving more reliable supervision on both source and target domains. Inspired by [135], we model the similarity between the labeled and unlabeled pixels of the target domain by modeling the underlying target distribution using the adaptive GMM. Specifically, we use target weak labeled pixels as the centers of Gaussian mixtures, allowing us to model the data distribution for each class in the high-dimensional feature space. This enables us to utilize the soft GMM predictions to provide more probabilistic guidance for unlabeled regions, enabling more guided and effective contrastive learning. Our framework utilizes reliable information from unlabeled/weakly labeled data to enhance model performance in GDA settings on the target domain. We summarise our main contributions as:

1. We formally introduced the GDA setting, where both the source and target domains are weakly labeled.

2. We developed the GenGMM model, which incorporates the underlying distribution of the source

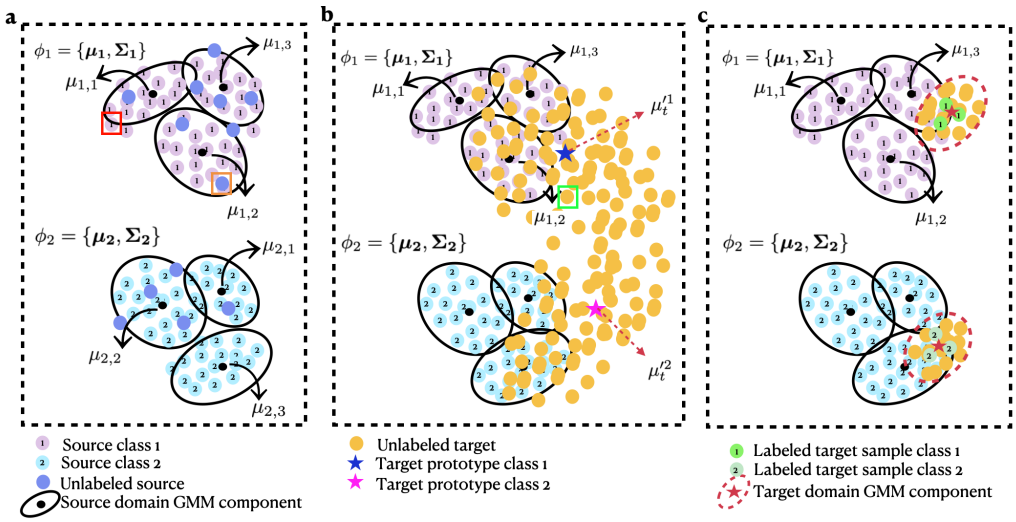


Figure 6.1: GMM-based contrastive learning. a) Labeled and unlabeled source data along with the GMM model with 3 components fitted on labeled source data. b) Unlabeled target data, together with the source GMM model. c) Unlabeled target data, coupled with the adaptive GMM model fitted to labeled target data and the source GMM model.

and target domain data using a GMM as a generative model, in conjunction with a discriminative classifier, to enhance the performance of the UDA model.

3. Extensive experiments conducted on numerous benchmark datasets have confirmed the effectiveness of the GenGMM approach within the GDA settings.

4. We showcase the superior performance of our approach compared to the current state-of-the-art on cell-type adaptation in immunofluorescent images, where each cell type serves as an individual domain. We highlight the effectiveness of the GenGMM model in improving segmentation/detection performance across different cell types.

5. The experimental results indicate that the GenGMM approach achieved high performance where the source data contains real-world label noise.

6.2 Methodology

In this section, we begin by providing an overview of the foundations (Sec. 6.2.1), in which we introduce the GDA settings and the preliminary information essential for understanding our approach. Following this, we will delve into the key components of the GenGMM approach, beginning with the estimation of the source domain distribution (as detailed in Sec. 6.2.2), where we employed GMM to estimate the underlying source domain's pixel distribution. We then proceed to explain our GMM-based contrastive learning approach (Sec. 6.2.3), in which we outline our strategy for reducing the domain gap between the source and target domains using contrastive learning, guided by the underlying pixel distribution of the source and target domains.

6.2.1 Preliminaries

In the GDA setting, the underlying source and target domain distributions are $p_s(x,y) \in p_{\mathcal{S}}$ and $p_t(x,y) \in p_{\mathcal{T}}$, from which the source $\mathcal{D}_S$ and target $\mathcal{D}_T$ datasets are sampled i.i.d. We sample labeled source data $S_l = \{x_i^{s,l}, y_i^{s,l}\}$ from $\mathcal{D}_S$, unlabeled source data $S_u = \{x_i^{s,u}\}$ from the marginal distribution of $\mathcal{D}_S$ over X, labeled target data $T_l = \{x_i^{t,l}, y_i^{t,l}\}$ from $\mathcal{D}_T$, unlabeled target data $T_u = \{x_i^{t,u}\}$ from the marginal distribution of $\mathcal{D}_T$, where $x \in R^{H\times W\times 3}$, $y \in R^{H\times W\times C}$ and C is the number of classes.

$$L_{ce}^{l} = -\sum_{d\in\{s,t\}}\sum_{i=1}^{H\times W}\sum_{c=1}^{C} I_{[y_i^{d,l}=c]} log(\hat{y}_{i,c}^{d,l}) \tag{6.1}$$

$$L_{ce}^{u} = -\sum_{d\in\{s,t\}}\sum_{i=1}^{H\times W}\sum_{c=1}^{C} w_{t,i,c} I_{[\hat{y}_i^{d,u}=c]} log(\hat{y}_{i,c}^{d,u})$$

$$\hat{y}_i^{d,u} = \underset{c}{argmax}\ \hat{y}_{i,c}^{d,u}, \qquad I \in \{1,2,..,H\times W\} \tag{6.2}$$

$$w_{t,i,c} = \frac{\sum_{i=1}^{H\times W} 1_{[\max_c \hat{y}_{i,c}^{d,u}]>\beta}}{H\times W}$$

To enhance the model's performance in the target domain, GDA primarily focuses on training with all S_l, S_u, T_l, and T_u. The model itself is comprised of three key elements: an encoder (E), a multi-class segmentation head (CL), and an auxiliary projection head (F). When given an input image x, the auxiliary projection head processes the encoder's output to generate a feature map (f =

$F(E(x))$). These features are then transformed into an l_2-normalized feature vector. Subsequently, the multi-class segmentation head operates on the encoder's output to produce a class probability map ($\hat{y} = CL(E(x))$). For model training, we employ the cross-entropy loss (L_{ce}) over both labeled and unlabeled data (i.e., Eqs. 6.1 - 6.2), in addition to the GenGMM loss functions. Eq. 6.1 represents the cross-entropy loss function applied to the labeled data from both the source and target domains, utilizing the ground truth labels $y_i^{d,l}$, with d indicating the domain, which can take values source (s) or target (t). However, the cross-entropy loss function for unlabeled data from both the source and target domains is computed using Eq. 6.2. Since pseudo labels are typically noisy, as suggested in [51], we apply weights to the loss values. These weights are determined by the confidence weights ($w_{t,i,c}$) computed in Eq. 6.2. In addition, we implement the teacher-student architecture [127] and employ the same framework as used in [51, 50] to establish a robust foundation. Notably, the weights of the teacher network are assigned as the exponential moving average of the student network's weights in each iteration [127].

6.2.2 Source domain distribution

Our approach focuses on estimating the source domain distribution, denoted as $p(f_s, c) = p(f_s|c)p(c)$, using labeled source data. To achieve this, we estimate two vital components: the class conditional distribution $p(f_s|c)$ and the class prior $p(c)$. We establish uniform class source priors using rare class sampling (RCS), a technique introduced in DAFormer [50]. Additionally, as shown in Fig. 6.1-a, we employ a GMM model that consists of a weighted mixture of M multivariate Gaussians to estimate the class-conditional distribution $p(f_s|c)$ for each category c. The GMM model effectively models the pixel data distribution within the D-dimensional feature space, as described by Eq. 6.3. The GMM classifier, parameterized as $\phi_c = \{\boldsymbol{\pi_c}, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}\}_{c=1}^{C}$, is optimized online using a momentum-based variant of the (Sinkhorn) EM (Expectation-Maximization) algorithm, as proposed by Liang et al. (2022) [49]. The GMM model is optimized exclusively using source pixel embeddings with pseudo labels that match their ground truth labels. Consequently, the model remains unaffected by real-world label noise in noisy source domains. The EM method's objective is to maximize the log-likelihood over feature-label

pairs (i.e. $\phi_c^* = \arg\max_{\phi_c} \sum_{f_s:y_s=c} log\sum_{m=1}^{M} p(f_s, m|c;\phi_c)$).

$$\begin{aligned} p(\boldsymbol{f_s}|c;\boldsymbol{\phi_c}) &= \Sigma_{m=1}^{M} p(m|c;\boldsymbol{\pi}_c) p(\boldsymbol{f_s}|c,m;\boldsymbol{\mu_c},\boldsymbol{\Sigma_c}) \\ &= \Sigma_{m=1}^{M} \pi_{c,m} \mathcal{N}(\boldsymbol{f_s};\boldsymbol{\mu_{c,m}},\boldsymbol{\Sigma_{c,m}}) \end{aligned} \tag{6.3}$$

where, $\pi_{c,m}$ represents the prior probability for each class, with the constraint that $\Sigma_{m=1}^{M}\pi_{cm} = 1$. $\boldsymbol{\Sigma_c}$ and $\boldsymbol{\mu_c}$ signifies the covariance matrix and mean vector, respectively.

6.2.3 GMM-based contrastive learning

We perform contrastive learning between the pixel embeddings ($f_d, \forall d \in \{s,t\}$) and the components of the source GMM model, which act as the most representative prototypes, using Eqs. 6.4. Nevertheless, the selection of negative (q^-) and positive (q^+) prototypes is challenging for unlabeled or weakly labeled pixels, as pseudo-labels from the discriminative classifier are noisy. Consequently, using them may result in a significant drop in performance [48].

$$l_{GMMCl} = -\alpha log \frac{e^{f_d q_c^+/\tau}}{e^{f_d q_c^+/\tau} + \Sigma_{\substack{c=1 \\ c'\neq c}}^{C} e^{f_d q_{c'}^-/\tau}} \tag{6.4}$$

To address this challenge, we leverage two foundational assumptions: 1) the domain closeness assumption [126] and 2) the feature similarity between labeled and unlabeled pixels [135]. According to these two assumptions, we select the positive/negative prototypes based on the similarity between unlabeled or weakly labeled pixel embeddings and prototypes, which is measured using the fitted GMM model on the labeled source pixel embeddings. This, in turn, allows the method to identify positive/negative samples with greater precision. To further mitigate noise, we introduce weighting (α) that considers each feature embedding's proximity to the positive prototypes in contrastive training. The details of choosing the positive/negative prototypes for source and target pixel embeddings, along with the value of α, are described in the following sections. The summary of the GenGMM model is shown in Algorithm 4.

Algorithm 4: GenGMM model

for *Iter = 1:* N_{Iter} **do**

  for $n \in 1, ..., N^{s,l}_{batch}$ *(Labeled source minibatch)* **do**

    Get pixel source features f^l_s

    Update source pixel data distribution GMM model $\{\phi^*_c\}$ using Sinkhorn EM(Sec. 6.2.2)

    Identify positive&negative prototypes(Eq. 6.5-6.6)

    Set $\alpha = 1$ for pixel feature f^l_s Compute loss l_{GMMCl} for feature f^l_s (Eq.6.4)

  end

  for $n \in 1, ..., N^{s,u}_{batch}$ *(Un/weakly labeled source minibatch)* **do**

    Get pixel source features f^u_s

    Determine positive&negative prototypes (Eq.6.7)

    Compute α for pixel feature f^u_s (Eq.6.8)

    Compute loss l_{GMMCl} for feature f^u_s (Eq.6.4)

  end

  for $n \in 1, ..., N^t_{batch}$ *(Target minibatch)* **do**

    Get pixel target features f_t

    Assign pseudo-label to unlabeled data (Eq. 6.9-6.10)

    Determine positive&negative prototypes(Eq.6.11)

    if *Contains weak labeled target data* **then**

      Estimate target GMM via weak labels (Eq. 6.12) Compute α for pixel feature f_t (Eq.6.13)

    end

    if *Contains only unlabeled target data* **then**

      Set $\alpha = 1$ for pixel feature f_t

    end

    Compute loss l_{GMMCl} for feature f_t (Eq.6.4)

  end

end

6.2.3.1 Labeled source pixel embeddings

Based on the values of $p_s(m|f_s, c; \phi_c)$ for the given source sample and its corresponding ground truth label c, we determine both positive and negative prototypes for labeled pixels (S_l). Here, $p_s(m|f_s, c; \phi_c)$ represents the posterior probability, signifying the likelihood of data f_s being assigned to component m within class c. We compute this probability using Bayes' rule while assuming uniform class priors, as detailed in Eq. 6.5. We choose positive prototypes by taking the prototype associated with the mean of the nearest component that shares the same label, as indicated in Eq. 6.6. In parallel, we recognize the importance of the hardest negatives in metric learning, drawing from previous research [128]. To find the hardest negative prototypes across various categories with distinct labels, we identify the closest component per category, following the procedure described in Eq. 6.6. Noted, the number of hardest negative prototypes for each pixel embedding is C-1. For example, as depicted in Fig. 6.1-a, consider the labeled source sample with a ground truth label of 1 in the red square. In this case, we select $\mu_{1,1}$ and $\mu_{2,2}$ as the positive and negative prototypes. Contrastive training is conducted using Eq. 6.4, with α set to 1.

$$p_s(m|f_s, c; \phi_c^*) = \frac{\pi_{c,m}\mathcal{N}(f_s|\mu_{c,m}, \Sigma_{c,m})}{\Sigma_{m'=1}^{M}\pi_{c,m'}\mathcal{N}(f_s|\mu_{c,m'}, \Sigma_{c,m'})} \tag{6.5}$$

$$\begin{aligned} q^+ &= \{\mu_{c,m^+} |\ m^+ = \arg\max_m\ p_s(m|f_s, c; \phi_c^*), c = y^{s,l}\} \\ q_c^- &= \{\mu_{c,m^-} |\ m^- = \arg\max_m\ p_s(m|f_s, c; \phi_c^*), c\}, \forall c \neq y^{s,l} \end{aligned} \tag{6.6}$$

6.2.3.2 Unlabeled source pixel embeddings

Regarding the unlabeled source pixels ($S_u = \{x_i^{s,u}\}$), they pose a challenge as they lack any available ground truth labels for determining positive and negative prototypes. A common method is to assign pseudo labels to $X^{s,u}$ using the output of a discriminative classifier. However, these pseudo-labels are often imprecise and unreliable for supervising contrastive training, leading to a noticeable drop in performance [48]. In response to this challenge, we develop an effective approach for guiding unlabeled pixels. Given that pixels with identical semantic labels tend to cluster together in the feature space [135], we leverage the GMM model fitted to labeled source data in Sec. 6.2.2 to model the similarity

between labeled and unlabeled pixels. By utilizing the GMM model and measuring the similarity of a given pixel's embedding with the components of the GMM model, we select positive and hardest negative components following the procedure outlined in Eq. 6.7. For instance, in Fig. 6.1-a, consider the unlabeled source sample in the orange square, where we choose $\mu_{1,2}$ as the positive prototype and $\mu_{2,1}$ as the hardest negative prototype. To enhance the training process, we introduce a weighting mechanism for the contrastive training loss associated with each pixel embedding. This weight, denoted as α and calculated using the closest GMM component with the same labels as described in Eq. 6.8.

$$\begin{aligned} q^+ &= \{\mu_{c^+,m^+} |\ c^+, m^+ = \arg\max_{c,m}\ p_s(c, m|f_s, \phi_c)\} \\ q_c^- &= \{\mu_{c,m^-} |\ m^- = \arg\max_{m}\ p_s(m|f_s, c; \phi_c^*), c\}, \forall c \neq c^+ \end{aligned} \tag{6.7}$$

$$\alpha = e^{-\frac{d^2}{2\sigma_{c^+,m^+}}} \tag{6.8}$$

where, d represents the difference between f_s and μ_{c^+,m^+}. Weighting the contrastive training loss is essential to mitigate the impact of noise, as it can significantly affect contrastive training [48]. Notably, by omitting the term $\frac{1}{\sqrt{2\pi\sigma^2}}$, we restrict α to the range of 0 to 1. The value of α indicates the proximity of a given pixel to its associated GMM component, with higher values indicating greater similarity. As a result, the soft scoring mechanism assigns more weight to pixel embeddings that closely resemble their associated GMM components.

6.2.3.3 Noisy source-labeled pixel embeddings

In the presence of noisy source-labeled data, we apply weighted contrastive training to alleviate the impact of noise, following the same framework as described in Sec. 6.2.3.2. As detailed in Sec. 6.2.3.2 the weights (α) are computed based on the closest GMM component with the same label as the given pixel embedding. Despite the noise in the source domain data, the source GMM model remains reliable. This reliability is due to its construction process, detailed in Sec. 6.2.2, which involves using pixel embeddings with pseudo-labels that match their ground truth labels, effectively mitigating the influence of noise.

6.2.3.4 Unlabeled target pixel embeddings

To determine the positive and hardest negative prototypes for the unlabeled target pixels ($T_u = \{x_i^{t,u}\}$), we assign pseudo-labels to them using the posterior probability $p_t(c|f_t;\phi_c^*)$ and their similarities to the target prototypes, as demonstrated in Eq. 6.9. Then, the selection of positive and hardest negative prototypes is carried out using Eq. 6.11. The rationale behind utilizing Eq. 6.9 for reliable pseudo-label generation is rooted in the assumption that features from both domains cluster together in a shared space. In this shared space, clusters with matching semantic labels are expected to be closely situated, as observed in [38]. Then, we expect that unlabeled target embeddings should exhibit proximity to the associated source GMM component and target prototype sharing the same semantic label, as depicted in Fig. 6.1-b. For example, in Fig. 6.1-b, the unlabeled target sample within the green square is close to target prototype class 1 and the source GMM components of class 1, rather than those of class 2. These proximities can be computed using Eq. 6.9.

$$\hat{y}_t^c = \arg\max_c \; p_t(c|f_t;\phi_c^*) \times \frac{e^{cosine(\mu_c^t, f_t)}}{\Sigma_{c'} e^{cosine(\mu_{c'}^t, f_t)}} \tag{6.9}$$

$$p_t(c|f_t;\phi_c^*) = \sum_{m'} p_s(c,m'|f_t;\phi_c^*) \times \frac{\delta_{target}^c}{\delta_{source}^c} \tag{6.10}$$

$$\begin{aligned} q^+ &= \{\mu_{c,m^+} |\; m^+ = \arg\max_m \; p_t(m|f_t,c;\phi_c^*), c = \hat{y}^{t,u}\} \\ q_c^- &= \{\mu_{c,m^-} |\; m^- = \arg\max_m \; p_t(m|f_t,c;\phi_c^*), c\} \forall c \neq \hat{y}^{t,u} \end{aligned} \tag{6.11}$$

The key concern is how to derive the target domain prototypes and determine the posterior probability for a given target embedding. Target domain prototypes per category are established through an exponential moving average, utilizing reliable pixel embeddings sourced from the target memory bank. This iterative process involves updating the target bank with reliable pixel embeddings from target mini-batches. It begins by computing class-specific average pixel embeddings based on pseudo labels and assessing their cosine similarity with class means. Subsequently, the M pixel embeddings with the highest cosine similarity scores, signifying their trustworthiness, are chosen and incorporated into the target bank. The posterior probability, which signifies the likelihood of a given target sample belonging to class c, is computed using Eq. 6.10. δ_{target}^c and δ_{source}^c are the prior distribution of the target and source domain and are updated using the exponential moving average during training. The rationale

behind adjusting the posterior using the ratio $\frac{\delta^c_{target}}{\delta^c_{source}}$ is to mitigate the label shift issue, as discussed in [52] (proof provided in supplement). Finally, the contrastive learning loss value is computed using Eq. 6.4 with $\alpha = 1$, given the positive and hardest negative prototypes selected via Eq. 6.11.

6.2.3.5 Weak labeled target pixel embeddings

In the context of limited or coarse labels for the target domain, we enhance model performance through refined contrastive learning using weak labels. This approach relies on the assumption that the unlabeled target data is close to both the labeled source data and the labeled target data with the same semantic label [126, 135]. Drawing on our knowledge of the source data distribution (i.e. Eq. 6.3), we employ Eqs. 6.9-6.10 to determine reliable pseudo-labels for unlabeled target pixels. These pseudo-labels are selected based on their proximity to the nearest GMM components in feature space, ensuring reliability. We subsequently determine positive and hard negative prototypes using Eq. 6.11. Additionally, we expect unlabeled target pixels to naturally cluster with labeled target samples sharing identical semantic labels. In cases of point or coarse annotations for target-domain images, we mitigate the influence of noisy pseudo-labels during training based on this expectation. The challenge lies in verifying this proximity. To address this challenge, we draw inspiration from [135] and develop a method that leverages observed similarities between labeled and unlabeled pixels within each image. This is accomplished by fitting a Gaussian Mixture Model with K Gaussian mixture components to each image, where K represents the number of annotated classes, as depicted in Fig. 6.1-c. Mean and covariance are computed for each GMM component, as outlined in Eq. 6.12.

$$\begin{aligned} \mu_k &= \frac{1}{\Sigma_{I_{y_n^{t,l}=k}}} \Sigma_n I_{y_n^{t,l}=k} f_{t,n} \qquad \forall k = y^{t,l} \\ \sigma_k &= \sqrt{\frac{1}{N} \Sigma_{n=1}^{N} \hat{y}_n^t (f_{t,n} - \mu_k)^2} \end{aligned} \tag{6.12}$$

$$\alpha = e^{-\frac{d^2}{2\sigma_k}} \qquad \forall k = m^+ \tag{6.13}$$

We then compute α values using 6.13 as weighting factors to adjust each pixel's contribution during contrastive learning through Eq. 6.4. Here, d is defined as the difference between f_t and μ_k. Lower α values indicate weaker proximity to pixels of the same class, allowing us to reduce their impact on

Table 6.1: Comparison with state-of-the-art methods for noisy labeled source data

Cityscapes→Dark Zurich																				
Model	Road	S.Wa	Bld.	Wall	Fence	Pole	T.Lig	T.Sig	Veget.	Ter.	Sky	Pers.	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
DAFormer[50]	84.8	47.2	66.5	35.0	13.3	**45.4**	14.4	**34.8**	48.1	**27.4**	62.2	48.9	44.5	63.3	52.6	**0.8**	83.7	**40**	35	44.6
SePiCo[51]	**88.1**	**54.7**	67.8	31.9	18.3	41.0	24.6	32.4	59.7	21.5	78.3	34.2	**45.1**	68.3	33.5	0	26.8	14.3	16.9	39.9
GenGMM	86.9	45.1	**73.5**	**41.0**	**19.70**	21.8	**42.0**	29.9	**66.3**	21.9	**79.3**	**54.7**	36.6	**76.4**	**75.8**	0.4	**85.2**	38.6	**38.6**	**49.0**

Table 6.2: Comparison for partially labeled source data

Data	**GTA5→City.**			**Synthia→City.**		
Model	50%	70%	100%	50%	70%	100%
DAFormer[50]	65.5	65.4	68.3	58.2	59.1	60.9
SePiCo[51]	63.8	65.0	69.7	59.7	60.5	62.2
GenGMM	**67.8**	**68.3**	**70.4**	**61.4**	**62.0**	**63.3**

the loss function. Also, in the presence of weak target data, the α is used to weight the self-training instead of w, as it is more informative (See Sec. **??**).

Table 6.3: The comparison with SOTA methods for Point labels

Data	**GTA5→City.**		**Synthia→City.**		
Model	mIoU	gap	mIoU	mIoU*	gap
WeakSegDA[56]	56.4	-	57.2	63.7	-
WDASS[57]	64.7	+8.3	62.8	68.7	+5.0
GenGMM	**71.4**	+6.7	**65.1**	**72.4**	+3.4

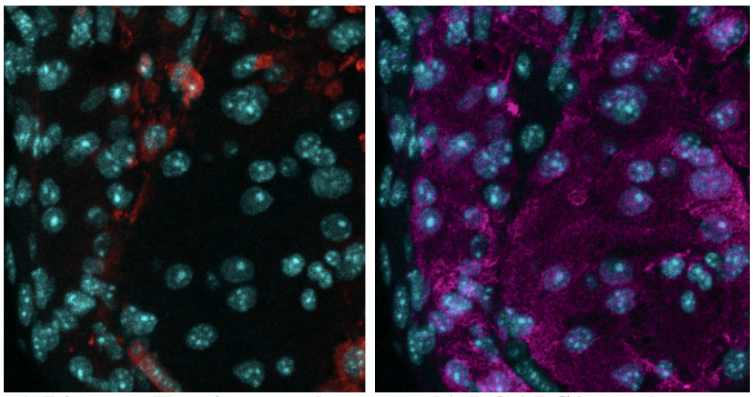


Figure 6.2: a) Blue-colored nuclei accompanied by the red Lineage tracing marker, b) Blue-colored nuclei accompanied by the purple LGALS3 marker

6.3 Experiments

6.3.1 Datasets and metric

We conducted our evaluation on four well-established domain adaptation benchmark datasets: GTA5[53], Cityscapes[129], SYNTHIA[54], and Dark Zurich[136]. For Cityscapes, we used 2975 training images and reported results on a validation dataset of 500 images. In the case of GTA5, Synthia, and Dark Zurich, we had 24,966, 9,400, and 2,416 training images, respectively. Our evaluation covered three scenarios: 1) noisy labeled source data, 2) partially labeled source data, and 3) weakly labeled target data. In the first scenario, we evaluated the model's performance on the Cityscapes $\rightarrow$ Dark Zurich DA benchmark by using Cityscapes coarse annotations as training labels to simulate a noisy source scenario. These annotations represent real-world label noise. In the second scenario, we divided the source data into labeled and unlabeled splits, with fine annotations from GTA5/Synthia in the labeled split. In the last scenario, we followed prior research [57] and conducted comparisons using both point and coarse labels. We generated point labels for each class within images at their original sizes by randomly selecting a small group of pixels. This selection was done within a randomly positioned circle with a radius of 4 per category. We opted for circles instead of points because different methods may use various image sizes, and point annotations can be lost when resizing. Additionally, we included

coarse weak labels from the Cityscapes dataset (gtCoarse). We primarily evaluated our approach based on the mean Intersection over Union (mIoU) score [50, 38, 51].

Furthermore, we evaluate the performance of the model on cell-type adaptation in immunofluorescent images, where each cell type serves as an individual domain. In this scenario, the source domain contains noise as the pixel-level segmentation masks are generated using point annotations. We highlight the effectiveness of the GenGMM model in improving segmentation/detection performance across different cell types. In this scenario, we utilized the Cardiovascular dataset which contains 26 3D multi-channel immunofluorescent images of advanced atherosclerotic lesions from two mouse models, with 13 used for training and 13 for testing. These images include multiple channels for nuclei and different markers such as Lineage Tracing, LGALS3, etc. as shown in Figure 6.2. Analyzing these images allows the identification of diverse cell types based on the overlapping presence of nuclei and various markers. The assignment of the "positive" label occurs exclusively when the detected nucleus precisely overlaps with the designated marker [137].

6.3.2 Network architecture and training

Our network architecture is based on the DAFormer framework [50], incorporating the Class-balanced Cropping (CBC) approach and regularization methods from [51] to enhance feature representations. These methods focus on prioritizing cropping from multi-class regions and ensuring smoother feature representations. We initialize the backbone model with pre-trained ImageNet weights [138] and add two 1×1 convolutional layers with ReLU activation [51] to obtain a 64-dimensional feature vector, which is further l_2-normalized (D=64). Our model was constructed utilizing PyTorch version 1.8.1, and trained on a single NVIDIA Tesla V100-32G GPU. The optimization strategy is AdamW [131] with betas (0.9, 0.999) and a weight decay of 0.01. Learning rates are $6 \times 10^{-}5$ for the encoder and $6 \times 10^{-}4$ for the decoder. All Exponential Moving Averages (EMAs) update weights are set to 0.9, except for the teacher network (β), which is set to 0.999. Our model undergoes training for 60,000 epochs with a batch size of 2. Notably, we impose a diagonal structure constraint on the covariance matrices $\Sigma \in \mathbb{R}^{D \times D}$ used in our GMM model. To optimize the performance of our GMM, we implement a generative optimization process, proposed in the GMMSeg framework [49]. Each iteration features

the momentum (Sinkhorn) Expectation-Maximization (EM) process, which is executed on both the current training batch and the external memory. The number of components is 5 for GTA5 and Dark Zurich and 3 for Synthia. The size of the external memory is 32k per category. This memory is governed on a first-in, first-out basis, with 100 pixels per class selected from each image for updates. Furthermore, for cell-type adaptation, the backbone of HoVer-Net+GenGMM is the same framework and settings as suggested in [137], with D set to 256. Also, we follow the same training regime as described in [137] for HoVer-Net+GenGMM. In HoVer-Net+GenGMM, we used the Adam optimizer [131] with the learning rate 1.0e-4 and set the betas set and weight decay at (0.9, 0.999).

6.3.3 Comparison with existing UDA methods

We perform three separate comparisons: 1) noisy labeled source data, 2) partially labeled source data, and 3) weakly labeled target data. In the first scenario, where the source data exhibits noise, we assess the performance of the GenGMM model using the Cityscapes → Dark Zurich Domain benchmark. To replicate the noisy source conditions, we employed coarse annotations from the Cityscapes dataset as training labels, representing real-world label noise. Our evaluation, detailed in Tab. 6.1, compares the GenGMM model against the DAFormer [50] and SePiCo [51] models on 151 test images, serving as the target test data for an online benchmark assessment available at online site[1]. Notably, our GenGMM method outperforms both the DAFormer and SePiCo models by a substantial margin, achieving a 4.4% and 9.1% mIoU improvement, respectively. The lower performance of SePiCo in the presence of source domain noise compared to the DAFormer model can be attributed to SePiCo's combination of contrastive training and self-training. As observed in [48], contrastive training is sensitive to noise and can significantly degrade model performance. In contrast, the GenGMM model effectively mitigates the noise's impact by utilizing the underlying distribution of the source domain.

Table. 6.2 presents the results for the second comparison, assuming scenarios where 50%, 70%, and 100% of the source data are labeled. In the 50% labeled scenario (i.e., 12,483 out of 24,966 for GTA5 and 4,700 out of 9,400 for Synthia), we compare the GenGMM model with the DAFormer [50] and SePiCo [51] models. Both the DAFormer and SePiCo models were trained by incorporating

[1]https://competitions.codalab.org/competitions/23553

Data	**GTA5→City.**		**Synthia→City.**		
Model	mIoU	gap	mIoU	mIoU*	gap
Coarse-to-fine[139]	66.7	-	61.6	67.2	-
WDASS[57]	69.1	+2.4	66.0	71.0	+3.8
GenGMM	**72.3**	+3.2	**71.6**	**76.2**	+5.2

Table 6.4: The comparison with SOTA methods for Coarse labels

Model	**GTA5→City.**	**Synthia→City.**
Point annotations		
Baseline (DAFormer)	68.9	61.9
GenGMM	**71.4**	**65.1**
Coarse annotations		
Baseline (DAFormer)	69.2	69.6
GenGMM	**72.3**	**71.6**

Table 6.5: The comparison with the Baseline model

Table 6.6: **Lineage Training → LGALS3**

Model	Precision	Recall	F1-score
HoVer-Net+Self-training	65.79	70.42	68.05
HoVer-Net+UniProto	77.4	68.9	72.0
HoVer-Net+GenGMM	73.1	76.4	73.9

Table 6.7: Comparison wrt to components

Lb	UL	GMM-Cl	**GTA5→City.**	**Synthia→City.**
✓			64.7	60.0
✓	✓		65.1	60.5
✓	✓	✓	**67.8**	**61.4**

Table 6.8: The effect of α during self-training (point annotations)

Data	**GTA5→City.**		**Synthia→City.**	
Model	w	α	w	α
GenGMM	70.5	**71.4**	63.6	**65.1**

self-training on the unlabeled source data. As observed, our GenGMM method outperforms both DAFormer [50] and SePiCo [51] models for both GTA5 → Cityscapes and Synthia → Cityscapes, as shown in Tab. 6.2. Similar trends are observed in the 70% and 100% labeled scenarios, highlighting the robust performance of the GenGMM model. The detailed results are shown in Tables A.1 and A.2.

In our latest comparison, we assessed the GenGMM model's performance with weakly labeled data, both point and coarse labels. We conducted comparisons with previous methods, namely, WeakSegDA [56] and WDASS [57] for point labels, and Coarse-to-fine [139] and WDASS [57] for coarse labels. The results, as shown in Tabs. 6.3 and 6.4, reveal that our GenGMM model consistently outperforms these prior approaches in both label types (point and coarse) across GTA5→ Cityscapes and Synthia→ Cityscapes datasets. Particularly, in the GTA5→ Cityscapes scenario, our method exhibits a significant performance boost, with a 6.7 mIoU increase for point labels and a 3.2 mIoU increase for coarse labels, surpassing state-of-the-art (SoTA) techniques. Similarly, in the Synthia→ Cityscapes setting, our method surpasses the SoTA for both weak label types, achieving a 2.3 and 5.6 mIoU increase for point and coarse labels across 16 classes. Furthermore, for 13 classes as indicated in Tabs. 6.3 and 6.4, our method secures a 3.4 and 5.2 mIoU increase for point and coarse labels, respectively. Tab. 6.5 compares the GenGMM and Baseline models, where the Baseline model utilizes DAFormer [50] trained with point and coarse labels. The results highlight the GenGMM model's superior performance across both DA benchmarks. We show the qualitative comparison of our framework with baselines for Dark zurich dataset in Fig. 6.4. We provide more qualitative results as well as classwise results in the Appendix.

6.3.4 Cell-type adaptation scenario

Furthermore, we evaluate the performance of our method for the cell-type adaptation with noisy source data. To adapt cell types in immunofluorescent images, we utilized the HoVer-Net+GenGMM domain adaptation model, built upon a modified HoVer-Net from [137]. In our approach, cell types are identified in each image based on their overlap with various markers. A cell is labeled positive for a specific marker if it exhibits overlap with that marker. Treating each cell type as a separate domain, our goal is to enhance the segmentation/detection performance of the trained model on one marker for other markers with different distributions, without the need for additional labeling efforts. In this scenario, the Lineage Tracing marker serves as a labeled source domain, while LGALS3 acts as an unlabeled target domain. We have access to positive/negative labels for nuclei only for the Lineage Tracing marker, while nuclei labels based on LGALS3 are unknown. This adaptation scenario involves both covariate and label shifts, as different markers have distinct distributions (covariate shift), and the distributions of positive and negative cells vary for different markers (label shift). It's worth noting that we follow the methodology outlined in [137] to convert these 3D images into 2D images within the source domain. In the target domain, we employ a linear combination approach to merge nuclei and marker channels, followed by slicing the images along the z-axis to transform them into 2D images. During model training, we extract patches of size 256×256 pixels with a 10% overlap from both the source and target domains. All segmentation pixel-level masks are generated using the point annotation and original images, following the approach introduced in [137]. Therefore the source domain labels are noisy. Table 6.6 highlights that GenGMM significantly improves the performance of the base domain adaptation model (HoVer-Net+Self-training), which relies solely on self-training, with increases of +7.31, +5.98, and 5.85 in precision, recall, and F1-score, respectively. Furthermore, HoVer-Net+GenGMM outperforms HoVer-Net+UniProto in terms of recall and F1-score while maintaining comparable precision. Additionally, Figure 6.3 presents a qualitative analysis of Lineage Tracing → LGALS3, illustrating that the HoVerNet+GenGMM model accurately predicts the labels of nuclei marked with dashed yellow color.

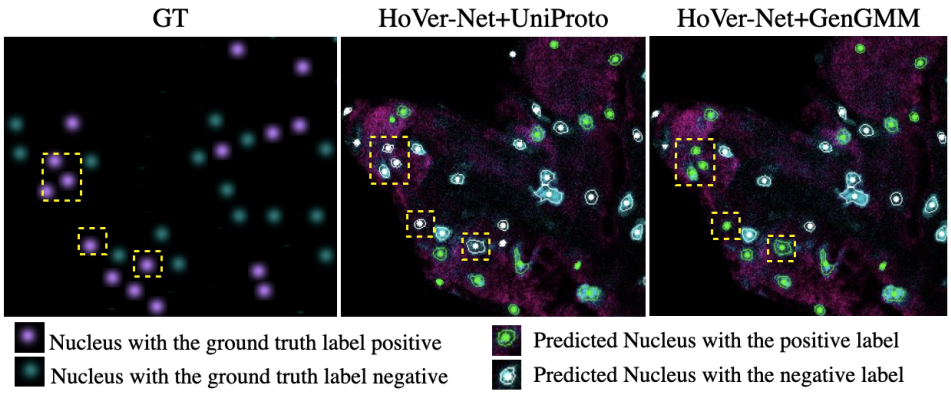


Figure 6.3: Qualitative analysis on Lineage Tracing → LGALS3.

Model	**GTA5→City.**				**Synthia→City.**			
M	1	3	5	7	1	3	5	7
mIoU	69.2	69.7	**70.4**	69.6	61.7	**63.3**	62.9	61.2

Table 6.9: Number of components

6.3.5 Ablation analysis

Tab. 6.7 investigates the advantages of training the model with unlabeled source data (UL) and applying GMM-based contrastive learning (GMM-Cl) in a scenario with 50% labeled source data. The results highlight the positive impact of training on unlabeled data and the effectiveness of GMM-Cl on model performance. As described in Sec. 6.2.3.5, we utilize the weight α derived from the GMM fitted to the target weak labels in place of the confidence weights w suggested in [51] for self-training. The impact of these α weights in self-training, in the context of target point annotations, is presented in Tab. 6.8. The data distribution for each source class is modeled using a mixture of M Gaussian components (Eq. 6.3). Tab. 6.9 shows optimal values for M in a 100% labeled data scenario: 5 for GTA5 and 3 for Synthia.

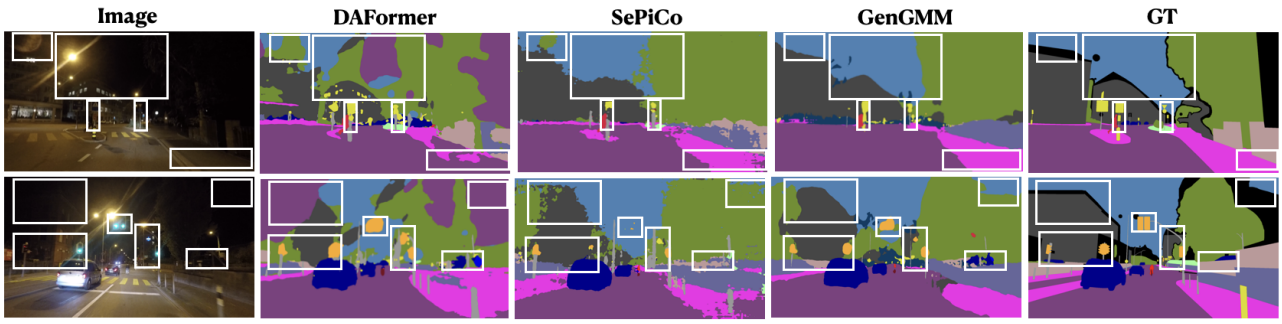


Figure 6.4: Qualitative results on Cityscapes→Dark Zurich

6.4 Conclusion

In this dissertation, we present GenGMM, a model designed for scenarios with partial or weak labeling in both source and target domains, collectively referred to as GDA. Many domain adaptation models assume perfectly labeled source data and unlabeled target data, which often doesn't hold in real-world scenarios. GenGMM addresses GDA by utilizing weak or unlabeled data from both domains to bridge the adaptation gap. It employs GMM models on source and target domains to capture similarities between labeled and unlabeled data, enhancing performance. Our experiments on various urban scene datasets as well as cell-type adaptation show significant performance improvements compared to existing approaches.

Chapter 7

Conclusion and future work

In this dissertation, we addressed the challenges of developing robust deep learning models for tasks with insufficiently labeled datasets. The widespread success of deep learning approaches often hinges on the availability of large labeled datasets, which may be impractical for diverse and complex tasks such as semantic segmentation and multivariate time series classification. To overcome these limitations, we designed four innovative approaches: Universal representation learning, Label-efficient Contrastive learning-based (LECL) model, ProtoGMM for self-training Domain Adaptation, and GenGMM for Generalized Domain Adaptation.

Our first two approaches, the Universal representation learning and LECL models, leverage supervised contrastive learning to handle limited labelings in tasks like time series classification and semantic segmentation. By introducing both instance and cluster-level contrastive learning, the model enhances its ability to discern meaningful patterns in scenarios where labeled data is scarce.

The third approach, ProtoGMM, focuses on self-training Domain Adaptation by incorporating a multi-prototype Gaussian-Mixture-based model. This model addresses the challenge of noisy pseudo-labels on the target domain by considering the underlying data distribution in both source and target domains, leading to improved intra-class semantic similarity and domain alignment.

Our fourth approach, GenGMM, extends the domain adaptation framework to the Generalized Domain Adaptation (GDA) setting, where both source and target domains may have partially or

noisily labeled data. By harnessing weak or unlabeled data from both domains, GenGMM refines noisy labels and effectively narrows the gap between them.

All three approaches were rigorously evaluated across diverse benchmarks, including multivariate time series classification, fluorescent image analysis, and urban scene adaptation. The results demonstrated significant improvements over state-of-the-art approaches, validating the effectiveness of our frameworks.

While our current research has made substantial contributions to addressing the challenges of limited labeling in deep learning tasks, there are several avenues for future exploration:

1. **Augmentation Strategies**: As mentioned in Chapter 1, future work will focus on exploring augmentation methods to further enhance the performance of the SupCon-TSC model. Investigating different augmentation techniques and evaluating their impact on the model's robustness and generalization could provide valuable insights.

2. **Scalability and Efficiency**: Scaling up the developed models to handle larger datasets and improving their computational efficiency is a critical aspect of future work. This involves optimizing the algorithms and architectures to accommodate more extensive and diverse datasets without compromising performance.

3. **Multi-Source Domain Adaptation**: Extending our research to accommodate scenarios with multiple source domains is a crucial avenue for future work. Many real-world applications involve data from diverse sources, and developing models capable of effectively leveraging information from these multiple domains remains a critical challenge. Investigating the generalizability of our developed approaches, ProtoGMM and GenGMM, to handle multiple source domains simultaneously will enhance their versatility and applicability in complex settings.

4. **Adaptation to Temporal Changes in Target Domain Distribution**: In dynamic real-world environments, the distribution of data in target domains can change over time. Future work should concentrate on the development and adaptation of the ProtoGMM and GenGMM models, enabling them to continuously learn and adjust to evolving distributions. This entails exploring techniques for detecting and responding to shifts in data characteristics, ensuring the

ongoing robustness and effectiveness of our models in dynamic scenarios.

In summary, the presented dissertation lays a foundation for future research in addressing labeling challenges in deep learning. Our approaches exhibit promising results, and ongoing efforts will continue to refine and extend these models for broader practical use.

Bibliography

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] X. Liu, C. Yoo, F. Xing, H. Oh, G. El Fakhri, J.-W. Kang, J. Woo, *et al.*, "Deep unsupervised domain adaptation: a review of recent advances and perspectives," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[3] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019.

[4] R. Tachet des Combes, H. Zhao, Y.-X. Wang, and G. J. Gordon, "Domain adaptation with conditional distribution matching and generalized label shift," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19276–19289, 2020.

[5] D. Minnen, T. Starner, I. Essa, and C. Isbell, "Discovering characteristic actions from on-body sensor data," in *2006 10th IEEE international symposium on wearable computers*, pp. 11–18, IEEE, 2006.

[6] X. Wang, Y. Gao, J. Lin, H. Rangwala, and R. Mittu, "A machine learning approach to false alarm detection for critical arrhythmia alarms," in *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pp. 202–207, IEEE, 2015.

[7] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 668–676, SIAM, 2013.

[8] P. Senin and S. Malinchik, "Sax-vsm: Interpretable time series classification using sax and vector space model," in *2013 IEEE 13th international conference on data mining*, pp. 1175–1180, IEEE, 2013.

[9] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 947–956, 2009.

[10] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, "Tapnet: Multivariate time series classification with attentional prototypical network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 6845–6852, 2020.

[11] J. Lines, S. Taylor, and A. Bagnall, "Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification," in *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 1041–1046, IEEE, 2016.

[12] B. Lucas, A. Shifaz, C. Pelletier, L. O'Neill, N. Zaidi, B. Goethals, F. Petitjean, and G. I. Webb, "Proximity forest: an effective and scalable distance-based classifier for time series," *Data Mining and Knowledge Discovery*, vol. 33, no. 3, pp. 607–635, 2019.

[13] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "Hive-cote 2.0: a new meta ensemble for time series classification," *Machine Learning*, vol. 110, no. 11, pp. 3211–3243, 2021.

[14] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 35, no. 2, pp. 401–449, 2021.

[15] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International joint conference on neural networks (IJCNN)*, pp. 1578–1585, IEEE, 2017.

[16] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[17] R. Virmani, F. D. Kolodgie, A. P. Burke, A. Farb, and S. M. Schwartz, "Lessons from sudden coronary death: a comprehensive morphological classification scheme for atherosclerotic lesions," *Arteriosclerosis, thrombosis, and vascular biology*, vol. 20, no. 5, pp. 1262–1275, 2000.

[18] B. M. Biccard, T. E. Madiba, H.-L. Kluyts, D. M. Munlemvo, F. D. Madzimbamuto, A. Basenero, C. S. Gordon, C. Youssouf, S. R. Rakotoarison, V. Gobin, *et al.*, "Perioperative patient outcomes in the african surgical outcomes study: a 7-day prospective observational cohort study," *The Lancet*, vol. 391, no. 10130, pp. 1589–1598, 2018.

[19] J. L. Rickard, G. Ntakiyiruta, and K. M. Chu, "Associations with perioperative mortality rate at a major referral hospital in rwanda," *World journal of surgery*, vol. 40, no. 4, pp. 784–790, 2016.

[20] P. Libby, "Inflammation in atherosclerosis," *Arteriosclerosis, thrombosis, and vascular biology*, vol. 32, no. 9, pp. 2045–2051, 2012.

[21] G. Pasterkamp, H. M. Den Ruijter, and P. Libby, "Temporal shifts in clinical presentation and underlying mechanisms of atherosclerotic disease," *Nature Reviews Cardiology*, vol. 14, no. 1, pp. 21–29, 2017.

[22] M. J. Davies, P. D. Richardson, N. Woolf, D. R. Katz, and J. Mann, "Risk of thrombosis in human atherosclerotic plaques: role of extracellular lipid, macrophage, and smooth muscle cell content.," *Heart*, vol. 69, no. 5, pp. 377–381, 1993.

[23] W. Adorno, L. S. Shankman, and D. E. Brown, "Combining multiple annotations to count cells in 3d cardiovascular immunofluorescent images," in *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–4, IEEE, 2021.

[24] F. Xing, H. Su, J. Neltner, and L. Yang, "Automatic ki-67 counting using robust cell detection and online dictionary learning," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 859–870, 2013.

[25] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot, "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Medical Image Analysis*, vol. 58, p. 101563, 2019.

[26] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi, "A dataset and a technique for generalized nuclear segmentation for computational pathology," *IEEE transactions on medical imaging*, vol. 36, no. 7, pp. 1550–1560, 2017.

[27] P. Naylor, M. Laé, F. Reyal, and T. Walter, "Nuclei segmentation in histopathology images using deep neural networks," in *2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017)*, pp. 933–936, IEEE, 2017.

[28] F. Mahmood, D. Borders, R. J. Chen, G. N. McKay, K. J. Salimian, A. Baras, and N. J. Durr, "Deep adversarial training for multi-organ nuclei segmentation in histopathology images," *IEEE transactions on medical imaging*, vol. 39, no. 11, pp. 3257–3267, 2019.

[29] H. Qu, G. Riedlinger, P. Wu, Q. Huang, J. Yi, S. De, and D. Metaxas, "Joint segmentation and fine-grained classification of nuclei in histopathology images," in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, pp. 900–904, IEEE, 2019.

[30] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Detecting overlapping instances in microscopy images using extremal region trees," *Medical image analysis*, vol. 27, pp. 3–16, 2016.

[31] H. Qu, P. Wu, Q. Huang, J. Yi, G. M. Riedlinger, S. De, and D. N. Metaxas, "Weakly supervised deep nuclei segmentation using points annotation in histopathology images," in *International Conference on Medical Imaging with Deep Learning*, pp. 390–400, PMLR, 2019.

[32] A. Chamanzar and Y. Nie, "Weakly supervised multi-task learning for cell detection and segmentation," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 513–516, IEEE, 2020.

[33] K. Nishimura, D. F. E. Ker, and R. Bise, "Weakly supervised cell instance segmentation by propagating from detection response," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 649–657, Springer, 2019.

[34] K. Tian, J. Zhang, H. Shen, K. Yan, P. Dong, J. Yao, S. Che, P. Luo, and X. Han, "Weakly-supervised nucleus segmentation based on point annotations: A coarse-to-fine self-stimulated learning strategy," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*, pp. 299–308, Springer, 2020.

[35] M. Treml, J. Arjona-Medina, T. Unterthiner, R. Durgesh, F. Friedmann, P. Schuberth, A. Mayr, M. Heusel, M. Hofmarcher, M. Widrich, *et al.*, "Speeding up semantic segmentation for autonomous driving," 2016.

[36] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[37] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation based on cascaded convolutional neural networks with uncertainty estimation," *Frontiers in computational neuroscience*, vol. 13, p. 56, 2019.

[38] S. Wang, D. Zhao, C. Zhang, Y. Guo, Q. Zang, Y. Gu, Y. Li, and L. Jiao, "Cluster alignment with target knowledge mining for unsupervised domain adaptation semantic segmentation," *IEEE Transactions on Image Processing*, vol. 31, pp. 7403–7418, 2022.

[39] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6936–6945, 2019.

[40] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[41] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2507–2516, 2019.

[42] C. Corbiere, N. Thome, A. Saporta, T.-H. Vu, M. Cord, and P. Perez, "Confidence estimation via auxiliary models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6043–6055, 2021.

[43] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5982–5991, 2019.

[44] H. Ma, X. Lin, Z. Wu, and Y. Yu, "Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4051–4060, 2021.

[45] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, "Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12414–12424, 2021.

[46] Y. Wang, J. Peng, and Z. Zhang, "Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9092–9101, 2021.

[47] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

[48] M. Vayyat, J. Kasi, A. Bhattacharya, S. Ahmed, and R. Tallamraju, "Cluda: Contrastive learning in unsupervised domain adaptation for semantic segmentation," *arXiv preprint arXiv:2208.14227*, 2022.

[49] C. Liang, W. Wang, J. Miao, and Y. Yang, "Gmmseg: Gaussian mixture based generative semantic segmentation models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31360–31375, 2022.

[50] L. Hoyer, D. Dai, and L. Van Gool, "Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9924–9935, 2022.

[51] B. Xie, S. Li, M. Li, C. H. Liu, G. Huang, and G. Wang, "Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[52] Y. Liu, J. Deng, J. Tao, T. Chu, L. Duan, and W. Li, "Undoing the damage of label shift for cross-domain semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7042–7052, 2022.

[53] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 102–118, Springer, 2016.

[54] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3234–3243, 2016.

[55] Z. Akata, B. Schiele, Y. He, Y. Xian, and A. Das, "Urban scene semantic segmentation with low-cost coarse annotation," 2022.

[56] S. Paul, Y.-H. Tsai, S. Schulter, A. K. Roy-Chowdhury, and M. Chandraker, "Domain adaptive semantic segmentation using weak labels," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pp. 571–587, Springer, 2020.

[57] A. Das, Y. Xian, D. Dai, and B. Schiele, "Weakly-supervised domain adaptive semantic segmentation with prototypical contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15434–15443, 2023.

[58] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," *Data Mining and knowledge discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[59] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 491–502, 2005.

[60] K.-P. Chan and A. W.-C. Fu, "Efficient time series matching by wavelets," in *Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*, pp. 126–133, IEEE, 1999.

[61] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, no. 1-23, p. 40, 2008.

[62] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, "Generalizing dtw to the multi-dimensional case requires an adaptive approach," *Data mining and knowledge discovery*, vol. 31, no. 1, pp. 1–31, 2017.

[63] I. Karlsson, P. Papapetrou, and H. Boström, "Generalized random shapelet forests," *Data mining and knowledge discovery*, vol. 30, no. 5, pp. 1053–1085, 2016.

[64] M. Wistuba, J. Grabocka, and L. Schmidt-Thieme, "Ultra-fast shapelets for time series classification," *arXiv preprint arXiv:1503.05018*, 2015.

[65] M. G. Baydogan and G. Runger, "Time series representation and similarity based on local autopatterns," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 476–509, 2016.

[66] K. S. Tuncel and M. G. Baydogan, "Autoregressive forests for multivariate time series modeling," *Pattern recognition*, vol. 73, pp. 202–215, 2018.

[67] M. G. Baydogan and G. Runger, "Learning a symbolic representation for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 29, no. 2, pp. 400–422, 2015.

[68] P. Schäfer and U. Leser, "Multivariate time series classification with weasel+ muse," *arXiv preprint arXiv:1711.11343*, 2017.

[69] F. He, T.-y. Fu, and W.-C. Lee, "Rel-cnn: Learning relationship features in time series for classification," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[70] K. Fauvel, T. Lin, V. Masson, É. Fromont, and A. Termier, "Xcm: An explainable convolutional neural network for multivariate time series classification," *Mathematics*, vol. 9, no. 23, p. 3137, 2021.

[71] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.

[72] R. Assaf, I. Giurgiu, F. Bagehorn, and A. Schumann, "Mtex-cnn: Multivariate time series explanations for predictions with convolutional neural networks," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 952–957, IEEE, 2019.

[73] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[74] X. Liu, K. Gao, B. Liu, C. Pan, K. Liang, L. Yan, J. Ma, F. He, S. Zhang, S. Pan, *et al.*, "Advances in deep learning-based medical image analysis," *Health Data Science*, vol. 2021, 2021.

[75] Q. Kang, Q. Lao, and T. Fevens, "Nuclei segmentation in histopathological images using two-stage learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 703–711, Springer, 2019.

[76] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers, "Cell detection with star-convex polygons," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 265–273, Springer, 2018.

[77] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P.-A. Heng, "Dcan: Deep contour-aware networks for object instance segmentation from histology images," *Medical image analysis*, vol. 36, pp. 135–146, 2017.

[78] H. Oda, H. R. Roth, K. Chiba, J. Sokolić, T. Kitasaka, M. Oda, A. Hinoki, H. Uchida, J. A. Schnabel, and K. Mori, "Besnet: boundary-enhanced segmentation of cells in histopathological images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 228–236, Springer, 2018.

[79] D. Liu, D. Zhang, Y. Song, C. Zhang, F. Zhang, L. O'Donnell, and W. Cai, "Nuclei segmentation via a deep panoptic model with semantic feature fusion.," in *IJCAI*, pp. 861–868, 2019.

[80] Y. Zhou, O. F. Onder, Q. Dou, E. Tsougenis, H. Chen, and P.-A. Heng, "Cia-net: Robust nuclei instance segmentation with contour-aware information aggregation," in *International Conference on Information Processing in Medical Imaging*, pp. 682–693, Springer, 2019.

[81] H. He, Z. Huang, Y. Ding, G. Song, L. Wang, Q. Ren, P. Wei, Z. Gao, and J. Chen, "Cdnet: Centripetal direction network for nuclear instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4026–4035, 2021.

[82] A. Bearman, O. Russakovsky, and V. Ferrari, "L., feifei. what's the point: Semantic segmentation with, point supervision," *arXiv preprint arXiv:1506.02106*, vol. 2, no. 6, 2015.

[83] Y. Zhou, Q. Dou, H. Chen, J. Qin, and P.-A. Heng, "Sfcn-opi: Detection and fine-grained classification of nuclei using sibling fcn with objectness prior interaction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[84] I. Yoo, D. Yoo, and K. Paeng, "Pseudoedgenet: Nuclei segmentation only with point annotations," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 731–739, Springer, 2019.

[85] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," *arXiv preprint arXiv:1812.11806*, 2018.

[86] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[87] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *International conference on machine learning*, pp. 819–827, PMLR, 2013.

[88] X. Liu, B. Hu, X. Liu, J. Lu, J. You, and L. Kong, "Energy-constrained self-training for unsupervised domain adaptation," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 7515–7520, IEEE, 2021.

[89] X. Liu, F. Xing, M. Stone, J. Zhuo, T. Reese, J. L. Prince, G. El Fakhri, and J. Woo, "Generative self-training for cross-domain unsupervised tagged-to-cine mri synthesis," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 138–148, Springer, 2021.

[90] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[91] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[92] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, "Crdoco: Pixel-level domain transfer with cross-domain consistency," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1791–1800, 2019.

[93] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[94] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7472–7481, 2018.

[95] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3752–3761, 2018.

[96] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, vol. 31, 2018.

[97] H. Wang, T. Shen, W. Zhang, L.-Y. Duan, and T. Mei, "Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation," in *European conference on computer vision*, pp. 642–659, Springer, 2020.

[98] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12635–12644, 2020.

[99] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1106–1120, 2021.

[100] Y. Zhang, P. David, H. Foroosh, and B. Gong, "A curriculum domain adaptation approach to the semantic segmentation of urban scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 8, pp. 1823–1841, 2019.

[101] X. Guo, C. Yang, B. Li, and Y. Yuan, "Metacorrection: Domain-aware meta loss correction for unsupervised domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3927–3936, 2021.

[102] M. Kim and H. Byun, "Learning texture invariant representation for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12975–12984, 2020.

[103] Z. Zheng and Y. Yang, "Unsupervised scene adaptation with memory regularization in vivo," *arXiv preprint arXiv:1912.11164*, 2019.

[104] M. Chen, H. Xue, and D. Cai, "Domain adaptation for semantic segmentation with maximum squares loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2090–2099, 2019.

[105] Y. Liu, J. Deng, X. Gao, W. Li, and L. Duan, "Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8801–8811, 2021.

[106] C. Sakaridis, D. Dai, S. Hecker, and L. 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)*, pp. 687–704, 2018.

[107] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.

[108] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *arXiv preprint arXiv:1406.2080*, 2014.

[109] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18661–18673, 2020.

[110] X. Wang, H. Fan, Y. Tian, D. Kihara, and X. Chen, "On the importance of asymmetry for siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16570–16579, 2022.

[111] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21271–21284, 2020.

[112] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.

[113] D. E. Brown, S. Sharma, J. A. Jablonski, and A. Weltman, "Neural network methods for diagnosing patient conditions from cardiopulmonary exercise testing data," *BioData Mining*, vol. 15, no. 1, pp. 1–15, 2022.

[114] N. Coronato, D. E. Brown, Y. Sharma, R. Bar-Yoseph, S. Radom-Aizik, and D. M. Cooper, "Functional data analysis for predicting pediatric failure to complete ten brief exercise bouts," *IEEE journal of biomedical and health informatics*, vol. 26, no. 12, pp. 5953–5963, 2022.

[115] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine learning research*, vol. 7, pp. 1–30, 2006.

[116] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[117] F. J. Baldán and J. M. Benítez, "Multivariate times series classification through an interpretable representation," *Information Sciences*, vol. 569, pp. 596–614, 2021.

[118] K. Fauvel, É. Fromont, V. Masson, P. Faverdin, and A. Termier, "Local cascade ensemble for multivariate data classification," *arXiv preprint arXiv:2005.03645*, 2020.

[119] L. M. Fan, A. Collins, L. Geng, and J.-M. Li, "Impact of unhealthy lifestyle on cardiorespiratory fitness and heart rate recovery of medical science students," *BMC public health*, vol. 20, no. 1, pp. 1–8, 2020.

[120] T. Matsuo, R. So, and M. Takahashi, "Estimating cardiorespiratory fitness from heart rates both during and after stepping exercise: A validated simple and safe procedure for step tests at worksites," *European Journal of Applied Physiology*, vol. 120, no. 11, pp. 2445–2454, 2020.

[121] R. Bar-Yoseph, S. Radom-Aizik, N. Coronato, N. Moradinasab, T. J. Barstow, A. Stehli, D. Brown, and D. M. Cooper, "Heart rate and gas exchange dynamic responses to multiple brief exercise bouts (mbeb) in early-and late-pubertal boys and girls," *Physiological reports*, vol. 10, no. 15, p. e15397, 2022.

[122] Y. Noguchi, M. Murakami, M. Murata, and F. Kano, "Microscopic image-based classification of adipocyte differentiation by machine learning," *Histochemistry and Cell Biology*, vol. 159, no. 4, pp. 313–327, 2023.

[123] Y. Nagao, M. Sakamoto, T. Chinen, Y. Okada, and D. Takao, "Robust classification of cell cycle phase and biological feature extraction by image-based deep learning," *Molecular biology of the cell*, vol. 31, no. 13, pp. 1346–1354, 2020.

[124] N. Moradinasab, Y. Sharma, L. S. Shankman, G. K. Owens, and D. E. Brown, "Weakly supervised deep instance nuclei detection using points annotation in 3d cardiovascular immunofluorescent images," 2022.

[125] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7303–7313, 2021.

[126] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," *arXiv preprint arXiv:1206.6438*, 2012.

[127] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Advances in neural information processing systems*, vol. 30, 2017.

[128] M. Kaya and H. Ş. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

[129] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[130] L. Hoyer, D. Dai, and L. Van Gool, "Hrda: Context-aware high-resolution domain-adaptive semantic segmentation," in *European Conference on Computer Vision*, pp. 372–391, Springer, 2022.

[131] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[132] Y. Zou, Z. Yu, B. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 289–305, 2018.

[133] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "Dacs: Domain adaptation via cross-domain mixed sampling," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1379–1389, 2021.

[134] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink, "Domain adaptive semantic segmentation with self-supervised depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8515–8525, 2021.

[135] L. Wu, Z. Zhong, L. Fang, X. He, Q. Liu, J. Ma, and H. Chen, "Sparsely annotated semantic segmentation with adaptive gaussian mixtures," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15454–15464, 2023.

[136] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7374–7383, 2019.

[137] N. Moradinasab, R. A. Deaton, L. S. Shankman, G. K. Owens, and D. E. Brown, "Label-efficient contrastive learning-based model for nuclei detection and classification in 3d cardiovascular immunofluorescent images," in *Workshop on Medical Image Learning with Limited and Noisy Data*, pp. 24–34, Springer, 2023.

[138] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[139] A. Das, Y. Xian, Y. He, Z. Akata, and B. Schiele, "Urban scene semantic segmentation with low-cost coarse annotation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5978–5987, 2023.

Appendix A

GTA5→Cityscapes																			
Model	Road	S.Wa	Bld.	Wall	Fence	Pole	T.Lig	T.Sig	Veget.	Ter.	Sky	Pers.	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
DAFormer	90.1	50.3	88.7	**54.2**	38.9	46.0	55.13	56.0	**89.6**	**50.9**	89.9	70.2	38.8	92.3	75.6	**79.9**	**63.5**	**53.2**	61.1	65.5
SePiCo	90.9	54.9	89.56	40.1	**50.2**	**50.9**	59.4	58.1	**89.6**	47.5	90.4	**73.8**	**44.3**	92.9	73.2	76.1	61.1	25.2	43.5	63.8
GenGMM	**97.0**	**77.3**	**89.9**	52.5	46.0	50.2	**60.0**	**60.4**	89.5	45.7	**91.4**	72.3	38.8	**93.4**	**76.8**	74.6	54.7	52.2	**64.3**	**67.8**
Synthia→Cityscapes																				
DAFormer	85.3	40.4	87.0	21.6	**3.6**	42.2	54.0	45.5	86.9	-	**93.6**	73.4	46.8	88.2	-	54.7	-	53.9	**54.4**	58.2
SePiCo	**90.2**	50.7	87.0	17.7	1.8	**50.8**	57.8	**47.7**	**87.1**	-	91.2	72.8	46.7	88.5	-	54.5	-	**57.2**	53.9	59.7
GenGMM	89.4	**55.5**	**88.0**	**38.2**	2.7	49.7	**59.1**	46.5	83.2	-	82.7	**75.46**	**47.7**	**89.65**	-	**64.4**	-	54.3	54.25	**61.4**

Table A.1: Comparison with state-of-the-art methods for partially labeled source data (50% labeled and 50% unlabeled)

GTA5→Cityscapes																				
Model	Road	S.Wa	Bld.	Wall	Fence	Pole	T.Lig	T.Sig	Veget.	Ter.	Sky	Pers.	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
DAFormer	88.8	48.4	88.6	47.4	37.3	45.0	56.2	52.5	**89.7**	**51.2**	90.3	70.6	42.7	92.2	75.5	**78.6**	**69.8**	**56.4**	61.9	65.4
SePiCo	96.1	71.22	89.5	**50.99**	47.1	49.4	58.5	59.2	89.2	46.2	90.5	73.5	41.0	92.4	63.0	70.4	31.6	51.3	62.8	65.0
GenGMM	**97.0**	**76.2**	**89.8**	48.6	**49.2**	**50.4**	**59.8**	**60.1**	89.5	48.2	**90.6**	**73.8**	**44.8**	**93.2**	**76.1**	76.3	57.7	52.4	**63.3**	**68.3**
Synthia→Cityscapes																				
DAFormer	84.6	40.0	87.1	**33.8**	5.8	40.0	53.7	31.7	**88.0**	-	**93.3**	74.0	46.4	86.9	-	**65.5**	-	54.6	**60.3**	59.1
SePiCo	**89.7**	53.8	86.7	26.1	2.9	46.4	**59.4**	41.7	85.7	-	89.7	**76.4**	**49.2**	88.9	-	56.8	-	**56.3**	58.3	60.5
GenGMM	89.6	**59.6**	**88.6**	26.7	**8.9**	**49.3**	57.5	**54.0**	86.2	-	90.5	76.1	**49.2**	**89.0**	-	62.5	-	54.9	59.0	**62.0**

Table A.2: Comparison with state-of-the-art methods for partially labeled source data (70% labeled and 30% unlabeled)

GTA5→Cityscapes																				
Model	Road	S.Wa	Bld.	Wall	Fence	Pole	T.Lig	T.Sig	Veget.	Ter.	Sky	Pers.	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
DAFormer	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	**92.5**	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
SePiCo	96.8	75.2	89.5	52.9	47.9	52.4	**59.7**	59.1	89.3	44.6	91.3	73.9	44.8	93.2	**78.8**	79.1	74.2	**56.7**	**65.8**	69.7
GenGMM	**97.3**	**79.5**	**90.1**	**55.6**	**52.2**	**53.7**	58.2	**63.4**	**90.5**	**49.5**	91.8	**74.6**	**46.4**	**93.3**	73.2	**80.0**	**68.7**	53.8	65.4	**70.4**
Synthia→Cityscapes																				
DAFormer	84.5	40.7	**88.4**	**41.5**	6.5	50.0	55.0	54.6	86.0	-	**89.8**	73.2	48.2	87.2	-	53.2	-	53.9	61.7	60.9
SePiCo	87.0	45.4	88.1	41.4	7.2	52.3	57.9	57.8	85.9	-	84.7	**75.2**	**50.6**	88.0	-	57.3	-	**55.7**	60.3	62.2
GenGMM	**93.4**	**64.3**	87.8	23.5	**14.1**	**53.6**	**60.1**	**59.4**	**86.3**	-	88.6	65.2	49.5	**89.3**	-	**62.3**	-	52.3	**63.6**	**63.3**

Table A.3: Comparison with state-of-the-art methods for partially labeled source data (100% labeled)

GTA5→Cityscapes																				
Model	Road	S.Wa	Bld.	Wall	Fence	Pole	T.Lig	T.Sig	Veget.	Ter.	Sky	Pers.	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
Coarse-to-fine	96.4	**75.1**	89.9	51.6	47.3	49.6	53.7	62.2	89.5	45.2	91.0	71.4	46.4	92.2	69.6	72.9	51.5	51.1	61.7	66.7
WDASS	95.5	71	89.2	49.3	**51.7**	52.0	**60.0**	64.2	89.8	**51.4**	91.5	**73.8**	46.5	91.5	69.4	75.3	68.3	55.0	68.4	69.1
GenGMM	**96.6**	74.6	**90.8**	**59.0**	50.3	**54.8**	59.0	**66.1**	**90.3**	50.3	**94.0**	73.0	**51.4**	**93.1**	**85.8**	**83.3**	**76.4**	**55.9**	**68.8**	**72.3**
Synthia→Cityscapes																				
Coarse-to-fine	95.5	69.9	87.3	38.4	29.7	44.9	40.1	53.7	87.0	-	90.3	70.9	39.9	-	87.8	53.6	-	35.4	61.6	61.6
WDASS	93.4	68.6	87.4	42.9	39.1	50.7	52.7	64.8	87.9	-	77.3	73.1	42.1	-	**89.3**	70.7	-	46.8	68.7	66.0
GenGMM	**96.8**	**77.0**	**90.6**	**50.1**	**51.2**	**54.5**	**58.2**	**69.6**	**90.0**	-	**93.6**	**74.7**	**50.5**	-	76.2	**83.8**	-	**58.7**	**70.5**	**71.6**

Table A.4: Comparison with state-of-the-art methods for weakly labeled target data (Coarse annotations)

GTA5→Cityscapes																				
Model	Road	S.Wa	Bld.	Wall	Fence	Pole	T.Lig	T.Sig	Veget.	Ter.	Sky	Pers.	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
WeakSegD	94.0	62.7	86.3	36.5	32.8	38.4	44.9	51.0	86.1	43.4	87.7	66.4	36.5	87.9	44.1	58.8	23.2	35.6	55.9	56.4
WDASS	95.5	71.3	87.6	43.3	43.3	47.7	51.3	58.7	87.0	45.5	86.4	**73.6**	49	91.4	56.7	65.2	63.2	46.8	67	64.7
GenGMM	**97.1**	**79.2**	**89.5**	**52.6**	**52.2**	**53.7**	**59.7**	**66.1**	**89.7**	**49.1**	**89.0**	72.3	**51.9**	**91.9**	**78.3**	**80.8**	**73.8**	**61.0**	**69.3**	**71.4**
Synthia→Cityscapes																				
WeakSegD	94.9	63.2	85.0	27.3	24.2	34.9	37.3	50.8	84.4	-	88.2	60.6	36.3	86.4	-	43.2	-	36.5	61.3	57.2
WDASS	**95.4**	**68.7**	85.4	37.5	**29.3**	44.0	48.9	56.4	86.8	-	86.8	70.6	47.1	**89.7**	-	50.8	-	41.1	**65.8**	62.8
GenGMM	90.6	50.5	**87.5**	**42.3**	4.3	**53.5**	**61.0**	**58.3**	**87.1**	-	**88.8**	**77.1**	**55.3**	89.6	-	**70.1**	-	**61.4**	63.9	**65.1**

Table A.5: Comparison with state-of-the-art methods for weakly labeled target data (point annotations)

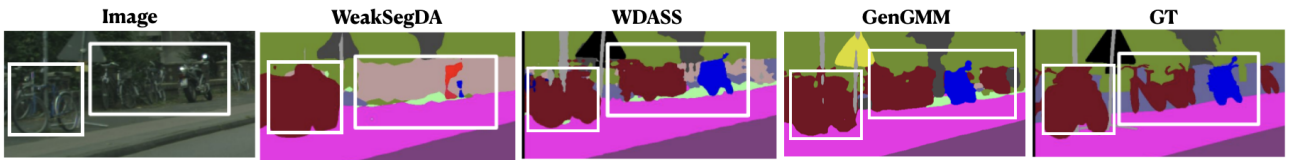


Figure A.1: Qualitative results on GTA5→Cityscapes setting for weakly labeled target data (point labels)

Figure A.2: Qualitative results on Synthia→Cityscapes setting for weakly labeled target data (point labels)

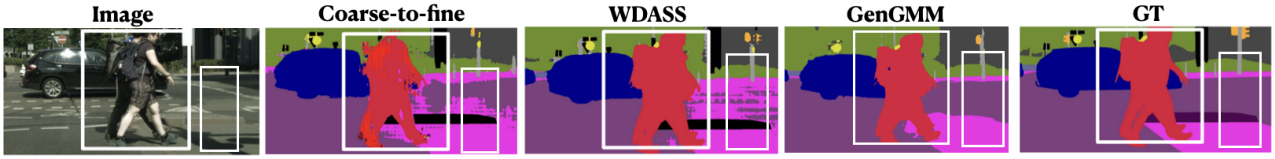


Figure A.3: Qualitative results on GTA5→Cityscapes setting for weakly labeled target data (Coarse labels)

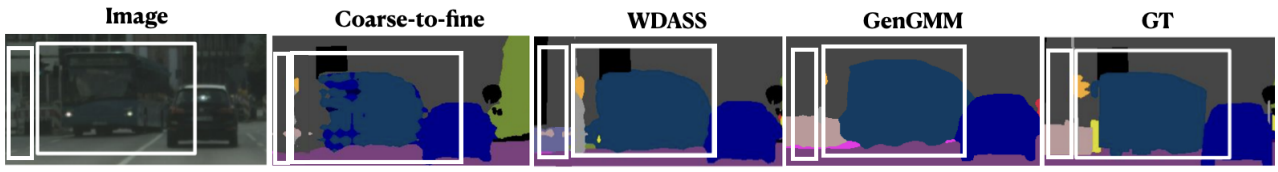


Figure A.4: Qualitative results on Synthia→Cityscapes setting for weakly labeled target data (Coarse labels)